



Tagungsband

Tim Berthold
Sebastian Brandt
Chris Schiermeyer
(Hrsg.)

Hannover
19. – 21. September 2016
Institut für Risiko und Zuverlässigkeit

© Institut für Risiko und Zuverlässigkeit 2016
1. Aufl. - Hannover: Institut für Risiko und Zuverlässigkeit, 2016

Herausgeber: Tim Berthold, Sebastian Brandt, Chris Schiermeyer
Institut für Risiko und Zuverlässigkeit
Leibniz Universität Hannover
Callinstraße 34, 30167 Hannover
www.irz.uni-hannover.de

Der Sammelband und dessen Beiträge wurden als Open-Access-Publikation unter den Bedingungen der Creative-Commons-Lizenz Namensnennung 3.0 Deutschland (CC BY 3.0 Deutschland) veröffentlicht (<https://creativecommons.org/licenses/by/3.0/de/>).

Die Verwendung, Verbreitung und Bearbeitung des Materials ist gestattet, sofern in angemessener Weise Urheber- und Rechteangaben gemacht werden.

Die Verantwortung für den Inhalt der einzelnen Beiträge liegt bei den jeweiligen Autoren.

Berichte aus der Bauinformatik

28. Forum Bauinformatik

Leibniz Universität Hannover

19. – 21. September 2016

Tim Berthold, Sebastian Brandt, Chris Schiermeyer (Hrsg.)

Hannover 2016

Vorwort

Das Forum Bauinformatik ist eine Tagung der Bauinformatiker/innen an deutschsprachigen Hochschulen und wird seit fast 30 Jahren jährlich an wechselnden Universitäten ausgetragen. In diesem Jahr dürfen wir als Mitarbeiter des Fachgebietes Bauinformatik am Institut für Risiko und Zuverlässigkeit der Leibniz Universität Hannover das Forum ausrichten. Die Veranstaltung richtet sich traditionsgemäß insbesondere an junge Wissenschaftler/innen. Dementsprechend sind in der Regel die meisten der Teilnehmer/innen wissenschaftliche Mitarbeiter/innen vor der Promotion aber auch Studierende höherer Semester, die ihre aktuellen Forschungsthemen und Abschlussarbeiten präsentieren.

Die Bauinformatik ist ein Querschnittsfach, das die Methoden der Informatik und Mathematik auf alle Bereiche des (Bau-)Ingenieurwesens anwendet. Dementsprechend vielseitig sind die Themengebiete, die auf dem Forum vertreten sind. Einen wesentlichen Schwerpunkt bildet seit einigen Jahren das Thema „Building Information Modeling (BIM)“. In diesem Jahr ist ungefähr ein Drittel der Beiträge mehr oder weniger eng diesem Bereich zuzuordnen. Wenn auch nicht unter dem Schlagwort BIM, so gehören die wesentlichen Aspekte von BIM traditionell zu den Forschungsfeldern der Bauinformatik: die objektorientierte, geometrische, semantische, Prozess- und Kooperationsmodellierung. Diese sind daher auch in der Lehre seit jeher ein fester Bestandteil.

Seit der ersten Veranstaltung war es immer die Intention des Forums, jungen Wissenschaftlern/innen eine Möglichkeit des Austausches über ihre Forschungsthemen in einer lockeren Atmosphäre zu geben, bei der man sich auf Augenhöhe begegnet. Dabei geht es weniger um die perfekte Präsentation, sondern vielmehr darum, sich offen und ehrlich auszutauschen, bei einem Glas Bier miteinander ins Gespräch zu kommen, über Forschungsfragen und auch -probleme zu diskutieren, Kontakte zu knüpfen und gemeinsam (auch unkonventionelle) Ideen zu entwickeln. Um sich nicht mit den „alten Hasen“ messen zu müssen, werden in der Regel bewusst nur Beiträge von Nachwuchswissenschaftlern/innen akzeptiert. Andererseits sind die Anregungen und Hinweise der „alten Hasen“ jederzeit willkommen. Sei es im Rahmen der Vorträge oder bei gemütlichen Diskussionsrunden danach. So können die jungen Wissenschaftler/innen von den erfahrenen lernen und profitieren. Andersherum ist dies für die „alten Hasen“ eine hervorragende Möglichkeit im Hinblick auf die eigene Personalplanung an der jeweiligen Universität oder in der Baupraxis den ein oder anderen Nachwuchskandidaten/in kennen zu lernen.

In diesem Sinne freuen wir uns auf das diesjährige Forum mit Euch jungen und jung gebliebenen Bauinformatikern/innen in lockerer Atmosphäre. Wir freuen uns auf interessante Forschungsthemen mit den dazugehörigen Fragen und Problemen, einen offenen und ehrlichen Austausch sowie auf konstruktive Kritik.

Zuletzt möchten wir uns bei allen bedanken, die sich für die Veranstaltung eingesetzt haben. Insbesondere danken wir den Mitgliedern des Arbeitskreises Bauinformatik für die Unterstützung, der Institutsleitung des Instituts für Risiko und Zuverlässigkeit für die Rückendeckung und allen, die in die Vorbereitungen involviert waren und bei der Ausrichtung der Tagung helfen.

Die Bauinformatiker/innen aus Hannover

Inhaltsverzeichnis

Session 1	9
<i>Robert Löhring, Nicole Hein</i> Erfassung, Modellierung und Persistierung großvolumiger spatio-temporaler Datenbestände für die Gewässerüberwachung	9
<i>Sören Loges</i> Vom Bildverband zum as-built BIM	19
Session 2	29
<i>Marcel Stepien, Dennis Pawlowski, Andre Vonthron</i> IFC-basierte Modellierung von Tunnelbauwerksmodellen	29
<i>Andreas Geiger, Iris Reichenbach, Karl-Heinz Häfele</i> IFC-Daten für die thermische Gebäudesimulation	41
<i>Maximilian Sternal, Kosmas Dragos</i> BIM-Based modeling of structural health monitoring systems using the IFC standard . . .	53
<i>Eric Robaszewski, Jakob Kirchner</i> Untersuchung von herstellereinspezifischen parametrischen Bauteiltypen in einem digitalen Gebäudemodell auf Vollständigkeit der Informationen und Einhaltung bautechnischer Randbedingungen	65
Session 3	75
<i>Ken Baumgärtel, Stephan Pirnbaum</i> Automatische Prüfung und Filterung in BIM mit Model View Definitions	75
<i>Alex Braun, Simon Vilgertshofer, Simon Daum</i> Automatisierte Prozessneugenerierung basierend auf Ergebnissen aus der Baufortschrittskontrolle	87
<i>Anna Wagner, Steffen Franz, Christian Leifgen</i> BIM-basierte Kollaborationsplattform mit objektorientierter Versionierungshistorie und Produktdatenkatalog zur Integration energetisch aktiver Fassadenelemente in Bauprozesse	97

<i>Christian Eller</i> Entwicklung eines Model Generators für private Haushalte auf Basis der IWU-Gebäude- typologie als Grundlage für EnergyPlus Gebäudesimulationen	109
Session 4	121
<i>Michael Kreger</i> Agenten-basierte Echtzeitdetektion von gefährlichen Bauprozessen	121
<i>Kosmas Dragos</i> An Embedded Physics-Based Modeling Approach for Wireless Structural Health Moni- toring Systems	129
Session 5	141
<i>Menno van Rahden, Jan Linxweiler</i> Mechanismen agiler Methoden	141
<i>Janis Zisenis, Jan Linxweiler</i> Entwicklung nachhaltiger Software mit Hilfe des objektorientierten Designs	153
<i>Sören Peters, Jan Linxweiler</i> Effiziente Implementierung eines Gittergenerators auf GPGPU's	165
<i>Stephan Lenz</i> Effiziente Simulation turbulenter Windströmungen im urbanen Raum	177
Session 6	191
<i>Ali Ismail, Yaseen Srewil, Saiful Abedin</i> Applying Visual Programming Methods for Parametric Bridge Modeling and Generation of Model Variations	191
<i>Katrin Jahr</i> Automatisierte Dimensionierung von Baustelleneinrichtungsgegenständen	205
<i>Marco Behrendt, Sebastian Brandt, Christoph Eckert</i> Optimierung von Gebietszerlegungen mit Hilfe der Partikelschwarmoptimierung	217
<i>Robert Irmeler</i> Anwendung genetischer Algorithmen zur Kalibrierung Typologie-basierter energetischer Gebäudemodelle	229
Session 7	241
<i>Quirin Aumann, Peter Kielar</i> A Modular Routing Graph Generation Method for Pedestrian Simulation	241

Giovanni Filomeno, Ingrid I. Romero, Ricardo L. Vásquez, Daniel H. Biedermann, Maximilian Bügler
Construction Site Pedestrian Simulation with Moving Obstacles 255

Daniel H. Biedermann, Sebastian Esser
Is macroscopic resolution enough? – A study about the use of different spatial scales in pedestrian dynamics 267

Marcus Dombois
Vergleich verschiedener Fingerprint-Berechnungsverfahren zur Standortbestimmung innerhalb von Gebäuden 279

Session 8 **293**

Dominik Ganghofer
Parametrisierte Architekturzeichnungen: Architektonische Entwurfsabsicht durch parametrisierte Koordinatenabhängigkeiten beschreiben und darstellen 293

Dorian Zank
Digitale Handskizzen Analyse: Stiftbasiertes Interface für Design Decision Support . . . 305

Jens Wiggenbrock
Integrierte Visualisierung von Installationsplänen und Installationskosten beim Glasfaser-Breitbandausbau 317

Erfassung, Modellierung und Persistierung großvolumiger spatio-temporaler Datenbestände für die Gewässerüberwachung

Robert Löhrling¹ und Nicole Hein²

Geodätisches Institut und Lehrstuhl für Bauinformatik & Geoinformationssysteme, RWTH University Aachen, Mies-van-der-Rohe-Str. 1, 52074 Aachen

¹E-Mail: loehrling@gia.rwth-aachen.de, ²E-Mail: hein@gia.rwth-aachen.de

Abstract Das Projekt *RiverView* ermöglicht eine ganzheitliche Sichtweise auf Fließgewässer, die den Detaillierungsgrad der EU-Wasserrahmenrichtlinie übertrifft und die Grundlage für ein zukunftsfähiges nachhaltiges Wassermanagement bildet. Der vorliegende Beitrag beschreibt das Konzept von der Datenaufnahme hin zur Datenmodellierung und Speicherung großer Datenmengen zum Gewässermanagement. Vorgestellt wird ein Ansatz, der insbesondere die Modellierung, Persistenz und das Abfragen der Daten effizient gestaltet und denen darauf aufbauende Analyse- und Visualisierungsverfahren eine effektive Grundlage bietet.

Keywords: Datenmodellierung, Datenmanagement, Geodatenbank, spatio-temporal

Einleitung

Die europäische Wasserrahmenrichtlinie (WRRL) beschreibt Maßnahmen und Ziele der Wasserpolitik, wie ein nachhaltiges Wassermanagement des Oberflächenwasserkörpers in Europa umgesetzt werden soll. Daraus resultiert die Notwendigkeit, den Zustand aller Gewässer Europas zu ermitteln und insbesondere den Aspekt der Renaturierung bzw. Überführung des Gewässers in seinen natürlichen Gewässerzustand in den Vordergrund zu stellen. Die Bestandsaufnahme fokussiert auf bestimmte Gewässerparameter, aus denen der Zustand abgeleitet werden kann, dazu zählt u. a. die Wasserqualität, die biologische Vielfalt oder die Gerinnebettmorphologie. Die Vorgehensweise der Zustandsbewertung eines Fließgewässers muss jedes Land der Europäischen Union eigenständig durchführen. Die Richtlinie beschreibt dazu normative Strukturparameter, die zu erheben sind und nach denen das Gewässer in Kategorien eingeordnet werden muss. Dieses Vorgehen dient der Ermittlung der ökologischen Bilanz des Fließgewässers und der ganzheitlichen Betrachtung des Gewässerzustandes, sodass eine Entwicklung der Fließgewässer hin zu einem guten ökologischen Zustand stattfindet, um folgenden Generationen eine gute Wasserqualität bereitzustellen (WRRL, 2000, Artikel 1).

Das Vorgehen im Land Nordrhein-Westfalen beschreibt das Amtsblatt 18 – Gewässerstruktur in Nordrhein-Westfalen – des Landesamts für Natur, Umwelt und Verbraucherschutz Nordrhein-Westfalen (LANUV). Es stellt die lokale Vorgehensweise als Kartieranleitung dar und umreißt,

welche Parameter zu erheben sind und wie Gewässer in Kategorien eingeordnet werden (LANUV, 2012). Das bisherige Vorgehen, um die Gewässer- und Strukturgüte bewerten zu können, besteht darin, Gewässer in Abschnitte einzuteilen und jeden Abschnitt einzeln vor Ort zu bewerten. Die Länge der Abschnitte variiert zwischen 100 m, 500 m oder 1000 m, je nachdem welche Gewässerbreite ansteht. Die Güte des Gewässers ermittelt sich schließlich aus der Summe der Einzelbewertungen der Abschnitte und mündet in eine normative Betrachtung in die Kategorien (sehr guter Zustand, guter Zustand, mäßiger Zustand und schlechter Zustand). Horizontale Bauwerke wie Staumauern oder Wehre stellen ein kritisches Element dar, das den kompletten Abschnitt oder sogar das gesamte Gewässer schlecht bewertet, obgleich vor und nach dem Bauwerk das Fließgewässer einen guten Zustand erhalten würde. Solche in situ Inspektionen sind äußerst aufwendig; daher existieren nur wenige Punktmessungen und seltene Kartierungen. Des Weiteren fehlt im bisherigen Vorgehen die gleichzeitige Erfassung von gewässermorphologischen Daten, Gewässergütedaten und hydrologischer Daten, weshalb es nur schwierig bis gar nicht möglich ist, quantitativ dynamische Prozesse und diverse Wechselwirkungen im Naturraum Fluss zu erkennen und abzubilden.

Das BMBF-geförderte Forschungsprojekt „*RiverView*“ soll einen Beitrag zum ganzheitlichen Gewässermonitoring von kleinen und mittleren Fließgewässern leisten und dabei insbesondere die Kriterien der europäischen Gewässerrahmenrichtlinie berücksichtigen. Das Vorhaben ist in der BMBF-Fördermaßnahme „Regionales Wasserressourcen-Management für den nachhaltigen Gewässerschutz in Deutschland“ (ReWaM) über den Förderschwerpunkt „Nachhaltiges Wassermanagement“ (NaWaM) eingebunden.

Ziel des Projekts ist der Entwurf einer Methode, mit der die integrierte Gewässerprozess- und Zustandsbeschreibung ermöglicht wird und die zusätzlich neue Analysemöglichkeiten bietet, um unter anderem die Morphodynamik des gesamten Flusses dauerhaft und hochauflösend darzustellen, Grundlagen für eine funktionale und leistungsorientierte Bewertung des fluvialen Ökosystems herauszustellen oder Prozessverständnisse aufzubauen und Stoffströme in einer raumzeitlichen Auflösung zu bilanzieren.

***RiverView* – Datenerfassung**

Grundlage von *RiverView* stellt die Entwicklung eines modularen unbemannten Mess-Katamarans (RiverBoat, siehe Abb. 1: RiverBoat - Mappingsystem für kleine und mittlere Fließgewässer.), ausgestattet mit multipler Sensorik zur räumlich sowie zeitlich referenzierten und hoch aufgelösten Erfassung von diversen Gewässerparametern dar. Mit *RiverView* soll es möglich werden, den Zustand und die Entwicklung des Gewässers direkt zu bewerten und automatisierte Analysen der Messwerte durchzuführen. Dazu werden bei jeder Befahrung eines Gewässers mit dem Katamaran hydrodynamische, -physikalische und -chemische Parameter u. a. durch den Einsatz von Multiparametersonden aufgezeichnet. Der Katamaran dient außerdem als Träger einer neuartigen Über- und Unterwassermappingplattform bestehend aus diversen Messeinheiten u.a. für die automatische Positions- und Orientierungsbestimmung des Bootes.

Die Sensorik des RiverBoats kann in drei zentrale Erfassungseinheiten untergliedert werden. Zum einen die bereits erwähnte Überwassermappingplattform (siehe Abb. 2: Überwassermappingplattform des RiverBoats.), die aus einem omnidirektionalen (360°-) Mehrkamerasystem sowie verschiedener Navigationssensoren besteht. In der Navigationseinheit sind eine GNSS (Global Navigation Satellite System)-Positionierungseinheit bestehend aus zwei räumlich divergent angebrachten Antennen und einer Inertialmesseinheit (Inertial Measurement Unit, IMU)



Abb. 1: RiverBoat - Mappingsystem für kleine und mittlere Fließgewässer.

zusammengefasst. Der plattformzugehörige Steuerrechner zeichnet alle Bild- und Navigationsdaten zeitlich und räumlich referenziert während einer Messfahrt auf. Als weitere Messeinheit ist die Unterwassermappingplattform zu nennen, die insbesondere die Beschaffenheit der Gewässersohle aufzeichnet. Hier werden ein Ultrabreitband-Echolot und ein Side-Scan-Sonar im Rumpf des Messbootes für die Abbildung des Gewässergrundes montiert. Das dritte Element stellt die Scan-Einheit dar, die das Messboot mit weiteren Messgeräten ergänzt. Dazu zählen ein mobiles Durchflussgerät, das, mithilfe der Acoustic Doppler Current Profiler (ADCP) Methode, Fließgeschwindigkeiten und Wassertiefen erfasst, eine Multiparametersonde (MPS-K16), die bis zu sechszehn Gewässergüteparameter (u. a. Temperatur, Leitfähigkeit, Sauerstoffgehalt oder Trübung) kontinuierlich aufzeichnet sowie eine Impuls-Neutron-Neutron Sonde (INN), welche die Stratigraphie der Gerinnebettsedimente mehrere Meter tief festhält.

Das angestrebte Ergebnis ist die lückenlose Bilddokumentation durch Rundumbilder des befahrenen Flusslaufs. Die Darstellung der Bildinformationen kann in Analogie zu Google StreetView verstanden werden, sie wird jedoch um 3D-Punktwolken und Sonardaten, für eine umfassende Kartierung der Über- und Unterwassertopografie, erweitert sowie mit den abgeleiteten Gewässergüteparametern aus den Daten der Multiparametersonde, verknüpft. Wesentliche Bedeutung erhält hierbei die Kalibrierung des Multisensorsystems und die Fusion der Navigationssensorik für nahtlose zeitliche und räumliche Referenzierung der Messdaten sowie zum Tracking des Bootes. Mithilfe von modernen Bildverarbeitungsverfahren (Stitching, Dense Image Matching) werden aus den Einzelaufnahmen der Panoramakamera Rundumbilder und 3D-Punktwolken erstellt. Structure-from-Motion-(SfM) Verfahren werden schließlich eingesetzt, um texturierte Oberflächenmodelle abzuleiten. So ist es möglich, für die geometrische Dokumentation von Uferbauwerken einen höheren Detailgrad zu erhalten. Des Weiteren wird

durch die Integration von Bildzuordnungsverfahren die Positions- und Orientierungsbestimmung des Messbootes verbessert.

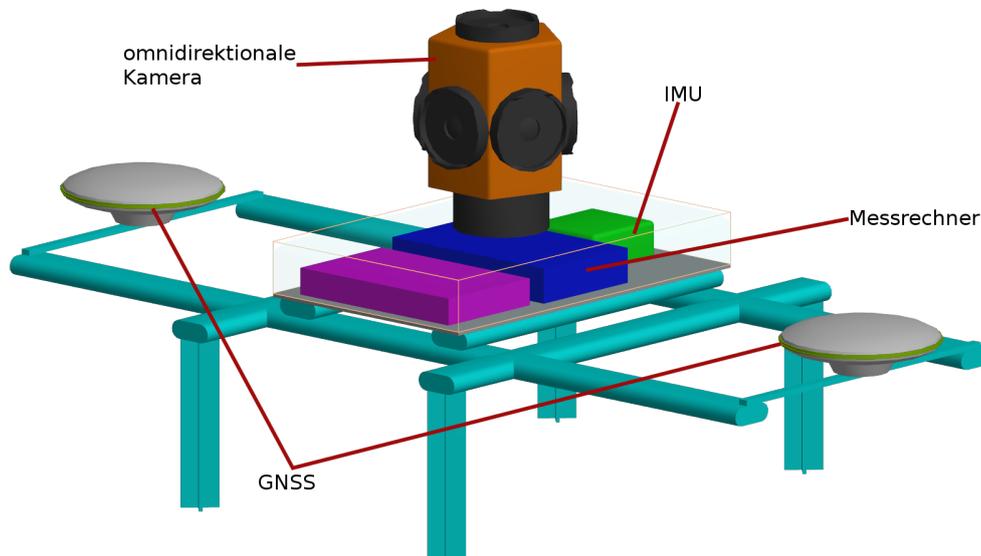


Abb. 2: Überwassermappingplattform des RiverBoats.

Die erfassten Daten sind deutlich detailreicher als es für eine grundlegende Gewässerstrukturgütekartierung notwendig ist, deshalb ermöglichen die Menge und der Detaillierungsgrad der Daten eine Vielzahl automatisierter Analyseverfahren für unterschiedliche Güteparameter. Da diverse Strukturparameter geometrischer Natur sind, bieten sich Analysemethoden direkt auf Datenbankebene an. Beispielsweise stellt die Laufkrümmung den Krümmungsgrad eines Gewässers dar, die aus deren Krümmungsradien abgeleitet werden. Diese Größe ist maßstababhängig, da sie die Gewässerbreite mit der Lauflänge ins Verhältnis setzt (Engels et al., 2016). Auch das Verhältnis zwischen Breiten und Tiefenerosion kann aus den gewonnenen Daten extrahiert werden. Dabei wird die Gewässerbreite zu der Gewässertiefe ins Verhältnis gesetzt und je nach Verhältnis dominiert die Breiten oder Tiefenerosion an dem Gewässerabschnitt. Als weiteres Beispiel ist die Breitenvarianz zu erwähnen. Es wird dabei die Variabilität der Gewässerbreite eines Gewässerabschnitts beurteilt, um das Querprofil eines Gewässers beschreiben zu können. Neben den erwähnten geometrischen Eigenschaften beschreibt (LANUV, 2012) viele Strukturparameter, die sich aus den ermittelten Daten direkt oder indirekt ableiten lassen, sodass eine zügigere und genauere Bewertung der einzelnen Streckenabschnitte ermöglicht wird.

Persistierung spatio-temporaler Datenbestände

Die aufgezeichneten Daten sind aufgrund der großen Anzahl unterschiedlicher Sensoren heterogen in ihrer Struktur und resultieren aufgrund der hohen Aufzeichnungsraten in großvolumigen Datenbeständen (Big Data), sodass die Herausforderung einer geeigneten Modellierung und Persistierung in einer raumzeitlichen Datenbank besteht. Der Aufbau dieser spatio-temporalen Datenbank umfasst die effiziente Speicherung der verschiedenartigen Messdaten, die nicht nur

zeitlich, sondern auch räumlich referenziert vom Über- und Unterwassermappingsystem des Katarans gesammelt werden. Neben der im Folgenden detailliert geschilderten Möglichkeit die erfassten Daten zu persistieren, werden derzeit verschiedene Datenbankmanagementsysteme untersucht, um vor allem die hohe Anzahl an Bilddaten effektiv verarbeiten zu können.

Im Folgenden wird das aktuell vorliegende Datenmodell beschrieben. Dabei wird insbesondere auf den Erstellungsprozess mit dem Entity-Relationship-Modell eingegangen.

Das Datenmodell und die Auswahl eines Datenbankmanagementsystems

Bei der Auswahl eines geeigneten Datenbankmanagementsystems (DBMS) ist auf verschiedene Aspekte zu achten. Zunächst wird herausgestellt, welche Daten später mithilfe des DBMS verwaltet werden sollen. Ebenfalls muss beachtet werden, wer diese Daten später verarbeiten möchte und in diesem Zusammenhang, von welchem Client aus diese Daten aus der Datenbank abgerufen werden.

Die während einer Fahrt gesammelten Daten müssen mit Raum- und Zeitreferenz persistiert und zur Verarbeitung in einem Geoinformationssystem (GIS) zudem kompatible Schnittstellen implementiert werden.

PostgreSQL (v9.4)¹ ist eine Open-Source-Software, die ein objektrelationales Datenbankmodell realisiert und somit das Bindeglied zwischen relationalen und objektorientierten Datenbanken bildet. Diese objektrelationalen Modelle werden häufig für GIS genutzt, da hier die Möglichkeit der Beschreibung von raumzeitlichen Beziehungen zwischen Objekten und Daten besteht.

PostgreSQL bietet dank langjähriger Entwicklung eine hohe Stabilität und ist vor allem für große Datenmengen konzipiert. Laut Schönig (o. J.) können Relationen einer PostgreSQL-Datenbank bis zu 128 Terabyte (bei einer Blockgröße von 32kB) an Daten umfassen. Datensätze, die sich in einer Relation befinden, die bis zu 1600 Spalten aufweisen und wovon zumindest in der Theorie unendlich viele angelegt werden können, können bis zu 1.6 Terabyte groß sein.

Durch ein effizientes Sperren von Tupeln bei mehreren gleichzeitig eingehenden Datenbankabfragen wird sichergestellt, dass das DBMS auch unter voller Last hoch skalierbar bleibt. PostgreSQL besitzt acht verschiedene Möglichkeiten, Tupel bei hohen Anfragezahlen zu sperren (Schönig (o. J.)).

PostgreSQL lässt sich leicht erweitern, entweder durch eigene, benutzerdefinierte Datentypen oder auch Operatoren, wodurch sich die Datenbank an eigene Bedürfnisse anpassen lässt. In diesem Zusammenhang stehen ebenso die Erweiterungen, die PostgreSQL auf verschiedenen Gebieten einsatzfähig macht. Insbesondere die PostGIS²-Erweiterung wird für dieses Projekt genutzt. Mit dieser Erweiterung ist es möglich, mit Geometrien, Topologien und räumlichen Funktionen zu arbeiten. PostGIS kann in PostgreSQL nachinstalliert werden und liegt zu diesem Zeitpunkt in der Version 2.2.0 vor.

¹<https://www.postgresql.org/>

²<http://www.postgis.net/>

Entity-Relationship-Modell – Phasen der Modellierung

Wird zur persistenten Speicherung von Daten eine Datenbank genutzt, so muss im Entwurf zunächst definiert werden, welche Objekte bzw. Datensätze in dieser Datenbank gespeichert werden sollen. In Burnus (2007) wird der sogenannte Life Cycle der Datenbankentwicklung beschrieben, der in vier Phasen unterteilt werden kann.

Die *Externe Phase* stellt die erste Phase dar, zu der vor allem die Analyse der Daten sowie die Datenmodellierung zählen. Diese Phase ist der eigentliche Datenbankentwurf, in dem mit dem zukünftigen Benutzer abgesprochen wird, welche Daten vorliegen und welche Daten für ihn abrufbar sein sollten. Das Ergebnis dieser Phase ist ein erstes Konzept der Datenbank.

In Bezug auf *RiverView* werden während einer Fahrt Daten von Sonargeräten, wasserphysikalischen Sensoren, der IMU, dem Echolot, den GNSS-Empfängern und der Panoramakamera erfasst, deren zu verarbeitende Datentypen verschiedene sind. Daher wurde in der ersten Phase ein Datenmodell entworfen und analysiert, welche Attribute mit diesen Daten korrespondieren und in welcher Beziehung die Daten zueinander und zum Parameter „Zeit“ stehen. Somit entstand ein erster Überblick über die Fülle und Vielfalt der Daten. Hier zeigte sich, dass viele Daten redundante Parameter beinhalten, wie etwa die Seriennummer und der Name der einzelnen Geräte.

Weiter beschreibt Burnus (2007) die zweite Phase und benennt sie als *Konzeptionelle Phase*. In dieser Phase entsteht ein Entity (Entitäten)-Relationship (Relationen)-Modell (ERM), das die gegebenen Objekte mit Beziehungen darstellt. Auch der Entwurf eines Unified-Language-Models (UML) wäre möglich, dessen Notation jedoch auf der ERM-Notation basiert und deswegen jede Datenbankentwicklung i.d.R. ein ERM zur Grundlage hat.

Mithilfe der modellierten und somit abstrahierten Daten wurde für *RiverView* eine erste Version eines ERMs erstellt. Anhand von Abbildung 3 (zeigt einen Ausschnitt eines ERM anhand des Echolots) ist erkennbar, dass die Attribute, die sich innerhalb eines Gerätes nicht ändern, unter „Metadaten“ gespeichert sind (hier unvollständig). Dies betrifft neben dem Namen und der Seriennummer ebenso die Beschreibung, die eventuell zusätzlichen Attribute und das Datum des ersten und letzten Einsatzes des Geräts sowie dessen Abmessung und Gerätedaten. Weiterhin ist sichtbar, dass jedes Gerät eine ID und einen Zeitstempel bezüglich der gemessenen Werte besitzt.

Beachtet werden muss, dass insgesamt sechs Einzelkameras (die Panoramakamera besteht aus fünf Seitenkameras und einer Zentralkamera) existieren und auch jede dieser Kameras als eigene Entität geführt wird. Ebenso verhält es sich mit den Sonargeräten und den übrigen Sensoren. Geplant, aber bisher noch nicht mangels Testdaten umgesetzt, ist das Konzept, dass nur neue Entitäten erstellt werden, sobald ein neuer Parameter eingelesen wird. Somit würden alle Sensoren, die den gleichen Parameter auf unterschiedlichen Trägersystemen messen, einer Entität angehören. Der Vorteil dieser Methode ist, dass keine neuen Entitäten mehr erzeugt werden müssen, sobald ein neuer Sensor (mit unterschiedlicher Seriennummer in Bezug auf die anderen Sensoren) registriert wird. Hierfür genügt es, wenn jeder Sensor Metainformationen darüber liefert, welcher Parameter gemessen wird, sodass jeder Sensor und dessen Daten richtig eingeordnet werden. Die ID setzt sich in diesem Fall aus Sensor-Seriennummer und Zeitstempel zusammen, womit es möglich ist, das Trägersystem des Sensors (durch die Seriennummer) zu analysieren und den Sensor für eine Fahrt eindeutig anhand der Einsatzzeit identifizierbar zu machen. Das in Abbildung 3 dargestellte ERM ändert sich derart, dass nicht mehr die verschiedenen Sensoren berücksichtigt werden, sondern die verschiedenen Parameter die Entitäten bilden. Werden neue Parameter erfasst, die noch nicht in der Datenbank verwaltet

werden, so wird eine neue Entität erstellt, in der die gemessenen Werte des Parameters anhand dieses Sensors gespeichert werden.

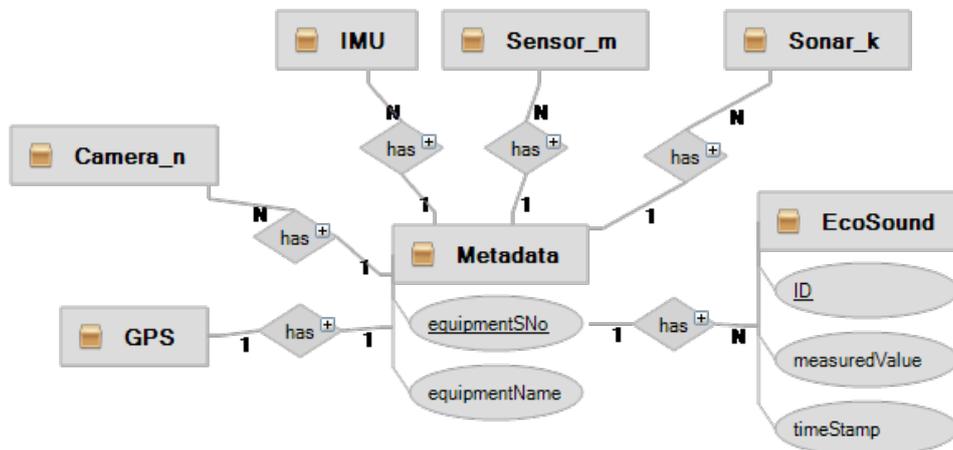


Abb. 3: ERM für RiverView (Ausschnitt).

Die Umsetzung der Entitäten „Camera_n“ der verschiedenen Kameras der Panoramakamera soll allerdings erhalten bleiben, da jede Kamera eine Information liefert, an welcher Position sie sich befindet. Dementsprechend werden die Bilddaten einer bestimmten Kamera stets über dieselbe Entität abgebildet.

Die *Logische Phase* setzt das ERM in ein Datenbankmodell um (Burnus (2007)). Spätestens in dieser Phase wird entschieden, welches Datenbankmanagementsystem zum Einsatz kommt.

Mithilfe des Open-Source-Datenbanksystems PostgreSQL wurde ein objektrelationales Datenbankmodell zur Umsetzung gewählt. Hierfür wurde für jede Entität eine Basisrelation angelegt, um die Attribute dieser Entitäten als Spalten festzulegen. Zusätzlich zu den bereits in Abbildung 3 dargestellten Entitäten kamen bisher noch die Entität „Track“, die Attribute für ID, Zeitstempel des Starts, Zeitstempel des Endes, den Name, die Beschreibung, Wetter, Attribute, gefahrene Route und Steuermann mitbringt und die Entität „Boat“ hinzu. Mithilfe der Attribute der Entität „Track“ kann unter anderem festgelegt werden, wann die Fahrt startete, wie sie heißt, welche Wetterbedingungen bzw. welche Jahreszeit vorherrschte, wer für die Fahrt zuständig war und wann sie endete. Dies ermöglicht aufgrund der zeitlichen und geometrischen Analyse eine genaue Zuordnung der Geräte zu einem „Track“, die ebenfalls georeferenziert ist. In der Entität „Boat“ werden alle Geräte des Trägersystems registriert, sodass deren Zugehörigkeit zu diesem Boot klar definiert ist. Mit jeder Änderung auf dem Boot (dynamisches Hinzufügen und Entfernen von Messinstrumenten) ändert sich ebenso die ID.

Die letzte Phase wird in Burnus (2007) als *Physische Phase* bezeichnet und folgendermaßen beschrieben: Das in der Logischen Phase entstandene Datenbankmodell wird in ein sogenanntes Datenbankmanagementsystem umgesetzt. In der *Physischen Phase* findet die eigentliche Implementierung statt.

Zusammenfassung

Im Rahmen des Projekts *RiverView* wird ein Datenbankmanagementsystem für gewässer- und bildbezogene Geodaten entwickelt, das in der Lage ist, großvolumige spatio-temporale Daten zu persistieren.

Wird der Life Cycle mit seinen vier Phasen betrachtet, so kann festgehalten werden, dass die Phasen 1, 2 und 3 abgeschlossen sind. Es ist bisher eine objektrelationale Datenbank entstanden, mit der georeferenzierte Daten persistiert und verwaltet werden können. Ziel ist es u. a., Daten insbesondere redundanzarm zu speichern, indem Tupel von identischen Werten vermieden werden. Dies wurde durch weitere Einführungen von Entitäten bewirkt, die allgemeine Werte (Seriennummer, Name, Beschreibung etc.) extern und nur einmalig für jedes Gerät speichern.

Funktionen, die bereits zur Verfügung stehen, sind beispielsweise administrative Tools für die Datenbank und deren Verbindung. Des Weiteren werden die bereits vorhandenen Testdaten analysiert und unter Zugehörigkeit zu den entsprechenden Entitäten hochgeladen. Sie können manuell durch einen entsprechenden SQL-Viewer für Datenbanken eingesehen werden.

Weitere Funktionen befinden sich zum jetzigen Zeitpunkt in der Implementierung. Von besonderer Bedeutung ist hierbei das Laden und Abfragen der Bilddaten der Panoramakamera und des Sonargeräts in die Datenbank. Da es sich bei diesen Daten um großvolumige Datenbestände handelt, ist die zeitliche Performance der Lade- und Abfragezeiten bisher noch sehr gering. Dieses Problem soll durch eine Erweiterung von PostGIS – PostGIS Raster – gelöst werden. Mithilfe dieser Erweiterung wird neben den Geometriedatentypen ein Rasterdatentyp eingeführt, der in der Lage ist, Rasterdaten (Bilder) in PostGIS zu unterstützen. Des Weiteren wird parallel an einer Lösung gearbeitet, die eine Array-Datenbank umsetzt, um Bilddaten spatio-temporal zu speichern. Beide Methoden finden operativ noch keine Anwendung in der aktuell vorliegenden Version der Implementierung, werden jedoch bereits ausgiebig getestet. Abbildung 4 zeigt den Geschwindigkeitsvergleich von PostGIS Raster (rot gestrichelt) mit einer Array-Datenbank (grün durchgezogen) beim Import einer Bilddatei der Größe 4.3MB in die Datenbank.

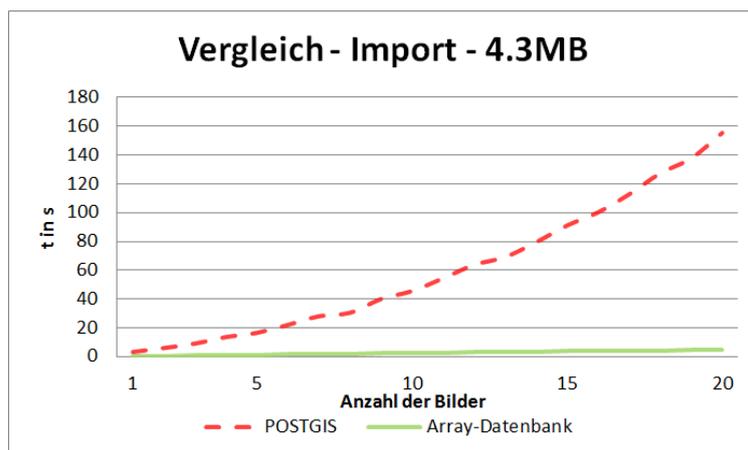


Abb. 4: Vergleich des Imports einer Array-Datenbank und PostGIS.

Auf der Abszisse ist die Anzahl der Bilder dargestellt, die zu diesem Zeitpunkt nacheinander in die Datenbank geladen wurden; auf der Ordinate ist die Zeit aufgetragen, die die jeweilige

Methode benötigt hat, um eine spezifische Anzahl der Bilder in die Datenbank zu schreiben. Deutlich zu erkennen ist, dass die Array-Datenbank um ein Vielfaches schneller Rasterdaten einlädt als die PostGIS-Erweiterung der PostgreSQL-Datenbank. Neben dem Laden der Daten spielt jedoch die räumlich indizierte Abfrage der Daten eine ebenso entscheidende Rolle. Die entsprechenden Untersuchungen mit Vergleich der beiden Implementierungen sind noch nicht abgeschlossen und Gegenstand der laufenden Arbeiten.

Ausblick

Das Forschungsprojekt *RiverView* umfasst die Entwicklung einer Messplattform, die in der Lage ist, zahlreiche heterogene Daten während einer Messfahrt zu sammeln. Dadurch ist es möglich, die Strukturgüte von Gewässern zu begutachten.

Neben der Strukturgütebewertung nach der europäischen Wasserrahmenrichtlinie sind diverse Einsatzszenarien für den Katamaran bzw. der mobilen Mappingplattform denkbar. Insbesondere bei Havarien und Störfällen, bei denen ein Stoffeintrag das Gewässer verunreinigt aber auch dynamische Veränderung des Gewässers durch diverse Umwelteinflussfaktoren (klimatischer Variabilität, Urbanisierung) können mit dem Fokus auf chemische und physikalische Gewässerdaten beschrieben und bewertet werden.

Literatur

- Brinkhoff, T. (2013). *Geodatenbanken in Theorie und Praxis: Eine Einführung in objektrelationale Geodatenbanken unter besonderer Berücksichtigung von Oracle Spatial*. Berlin Offenbach am Main: Wichmann.
- Burnus, H. (2007). *Datenbankentwicklung in IT-Berufen: eine praktisch orientierte Einführung mit MS Access und MySQL*. Wiesbaden: Vieweg+Teubner Verlag.
- Engels, R., Schüttrumpf, H., Blankenbach, J. und Schwermann, R. (2016). „RiverView – Die Digitalisierung der Wasserwirtschaft durch hochaufgelöste raumzeitliche Informationsmodellierung“. In: *Gewässerschutz, Wasser & Abwasser. 49. Essener Tagung für Wasser- und Abfallwirtschaft „Wasserwirtschaft 4.0“* (239). Hrsg. von J. Pinnekamp, S. 1–17.
- Hagwerdi-Poor, G. (2010). „GIS-Konzept und Konturen eines IT-Master-Plans: Planungs- und Systementwicklung für die Informationstechnologie“. In: Wiesbaden: Vieweg+Teubner Verlag.
- LANUV (2012). „LANUV-Arbeitsblatt 18 – Gewässerstruktur in Nordrhein-Westfalen“. In: *Kartieranleitung für die kleinen bis großen Fließgewässer*. Recklinghausen: Landesamt für Natur, Umwelt und Verbraucherschutz NRW.
- Redmond, E. und Wilson, J. R. (2012). *Sieben Wochen, sieben Datenbanken: Moderne Datenbanken und die NoSQL-Bewegung*. Köln: O'Reilly Verlag GmbH & Co. KG.
- Schönig, H.-J. (o. J.). *Gründe für PostgreSQL*. <http://www.cybertec.at/postgresql/grunde-fur-postgresql/> (besucht am 20.06.2016).
- Unterstein, M. und Matthiessen, G. (2013). *Anwendungsentwicklung mit Datenbanken*. Berlin Heidelberg: Springer-Vieweg Verlag.
- WRR (2000). *Wasserrahmenrichtlinie. Richtlinie 2000/60/EG des Europäischen Parlaments und des Rates zur Schaffung eines Ordnungsrahmens für Maßnahmen der Gemeinschaft im Bereich der Wasserpolitik vom 23. Oktober 2000*. Brüssel: Das europäische Parlament und der Rat der Europäischen Union.

Vom Bildverband zum as-built BIM

Sören Loges

*Geodätisches Institut, RWTH Aachen, Mies-van-der-Rohe Straße 1, Deutschland
E-Mail: loges@gia.rwth-aachen.de*

Abstract In diesem Paper wird grundlegend das Konzept von der Datenaufnahme hin zur automatisierten Erzeugung eines as-built BIM-konformen Modells skizziert. Hauptaugenmerk liegt auf der Segmentierung einer Punktwolke in einzelne Räume anhand der, während der Aufnahme zurückgelegten, Trajektorie. Der konzipierte Ansatz beruht auf der Analyse von Histogrammen. Für einzelne Positionen der Trajektorie werden die sichtbaren Punkte bestimmt und in Histogrammen zusammengefasst. Durch einen sequentiellen Vergleich werden die Histogramme der Positionen auf ausreichende Übereinstimmung getestet. Stimmen sie nicht überein wird ein Raumübergang detektiert. Das Ergebnis sind Histogramme einzelner Räume sowie die ungefähre Lage von Raumübergängen.

Abstract This paper outlines the general process from data acquisition to automated generation of as-built models. The main focus is on the segmentation of a point cloud in individual rooms based on the traveled trajectory while acquiring the data. The proposed approach is based on the analysis of histograms. From predefined positions on the traveled trajectory visible points are determined and summarized in histograms. These histograms are tested for sufficient agreement by sequential comparison. If the test failed, a transition (door) between two rooms is detected. The result contains histograms for the individual rooms and approximates the door placement.

Keywords: Photogrammetrie, SfM, as-built BIM

1 Einleitung

In Deutschland nimmt die Bedeutung der Methode Building Information Modelling (BIM) stark zu. So hat das Bundesministerium für Verkehr und digitale Infrastruktur im Dezember 2015 einen Stufenplan zur stufenweisen Einführung von BIM bei Planung und Umsetzung von großen Verkehrsprojekten bis 2020 verabschiedet. Die Grundlage von BIM bildet ein in der Regel semantikreiches bauteilorientiertes Gebäudemodell, das idealerweise während der Planungsphase erzeugt und über den Lebenszyklus des Gebäudes gepflegt wird. Handelt es sich jedoch nicht um einen Neubau sondern um eine Planung im Bestand, z.B. Sanierungs- oder Umnutzungsvorhaben, muss ein entsprechendes as-built Modell für die Planung erst aus dem Gebäudebestand erzeugt werden.

Für eine solche Bestandsaufnahme stehen verschiedene Verfahren bereit, welche jedoch meist eher mit konventionellen zeichnungsorientierten CAD-Modellen und weniger mit den geforderten bauteilorientierten Modellen konform gehen. Zur as-built Dokumentation bietet sich u.a. die Erzeugung von Punktwolken (z.B. durch terrestrische Laserscanner oder Photogrammetrie) an, da diese je nach Punktdichte ein detailreiches Abbild des Bestandes darstellen. Forschungsbedarf besteht hier in der Entwicklung und Erprobung von Aufnahmeverfahren, die eine kontinuierliche und zeiteffiziente sowie eine detailreiche Erfassung des Bestandes ermöglichen.

Die eigentliche Ableitung des Modells erfolgt nach der Datenaufnahme in der Nachbearbeitung. Im Falle einer aufgenommenen Punktwolke wird diese händisch strukturiert und Bauteile z.B. über die Extrusion von Schnittbildern erzeugt. Die Eingabe der weiteren Bauteileigenschaften erfolgt ebenfalls manuell. Diese größtenteils händische Erzeugung des bauteilorientierten Modells bedeutet einen hohen zeitlichen und personellen Aufwand. Diesen Prozess der Erzeugung des as-built Modells gilt es durch die Erforschung von automatisierten Verfahren effizienter zu gestalten.

2 Aufnahmeverfahren zur Bestandserfassung

Zur Bestandsdatenerfassung für das Gebäudeaufmaß können verschiedene originäre Messverfahren verwendet werden. Abb. 1 liefert hierzu eine Übersicht. Nach Blankenbach (2015) sind die vier für ein Gebäudeaufmaß grundsätzlichen Basisverfahren die Photogrammetrie, das terrestrische Laserscanning, die Tachymetrie und das elektronische Handaufmaß. Eine Diskussion der verschiedenen Messverfahren wird in Blankenbach (2015) gegeben und aus diesem Grund an dieser Stelle nicht weiter thematisiert.

Im BIM Kontext wird besonders das terrestrische Laserscanning für die as-built Dokumentation angepriesen, welches den Begriff „Scan-to-BIM“ geprägt hat. (Braunes, 2014). Ein Grund dafür ist die Erfassung von hochaufgelösten Punktwolken, die eine flexible Erzeugung von Modellen unterschiedlicher Detaillierungsgrade ermöglicht. Nachteile des terrestrischen Laserscannings sind die hohen Kosten, die lange statische Scandauer und die notwendige Verknüpfung, Registrierung, von Punktwolken unterschiedlicher Scanpositionen sowie der hohe manuelle Modellierungsaufwand im Post-Processing. Eine deutlich günstigere Alternative ist die Erzeugung einer Punktwolke aus Bildern mittels der photogrammetrischen Methode Structure from Motion (SfM), die eine zügige mobile Datenaufnahme ermöglicht, bereits eine zusammenhängende Punktwolke liefert und orientierte Bilder zur weiteren Analyse bereitstellt. Im Folgenden wird daher näher auf die Photogrammetrie und SfM eingegangen.

2.1 Photogrammetrie

Die Photogrammetrie umfasst allgemein Methoden, um aus einem oder mehreren Bildern die Form und Lage eines beliebigen Objektes zu ermitteln. Man unterscheidet in der Photogrammetrie die Aufnahme Konfigurationen: Einzelaufnahme, Stereoaufnahme und die Mehrbildaufnahme. Die Einzelaufnahme beschreibt die Erfassung eines Objektes mit einem Bild und findet ihre häufigste Anwendung in der Erstellung von Entzerrungen und Orthobildern sowie der ebenen Objektauswertung. In der praktischen Anwendung für die Gebäudemodellierung wird sie in der Auswertung ebener Hausfassaden genutzt. Die Stereoaufnahme beschreibt

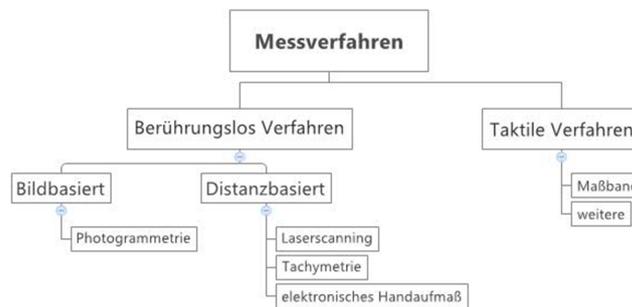


Abb. 1: Übersicht der originären Messverfahren zur Bestandsdatenerfassung

die Aufnahme eines Objektes anhand von zwei Bildern. Ist die äußere Orientierung der Aufnahmen zueinander bekannt, können 3D-Koordinaten homologer Punkte über einen räumlichen Rückwärtsschnitt bestimmt werden. Die äußere Orientierung beschreibt allgemein die Lage und Ausrichtung des räumlichen Bildkoordinatensystems im übergeordneten Objektkoordinatensystem und wird durch sechs Parameter, drei Translationen und drei Rotationen, festgelegt. Wie Abbildung 2 zu entnehmen ist, setzt sich die äußere Orientierung aus dem räumlichen Vektor X_0 , der vom Ursprung des Objektkoordinatensystems zum Projektionszentrum O der Kamera zeigt, und der Rotationsmatrix R, welche die Orientierung im Raum beschreibt, zusammen. Die Rotationsmatrix kann durch die Drehungen ω , φ , κ um die Koordinatenachsen X, Y, Z gebildet werden. Ebenfalls aus Abbildung 2 zu entnehmen sind die Parameter der inneren Orientierung (Kamerakonstante c, Bildhauptpunkt H, Verzeichnungsparameter) auf die jedoch an dieser Stelle nicht weiter eingegangen werden soll; hierzu sei auf die Fachliteratur (z.B. (Luhmann, 2003)) verwiesen. Die Mehrbildphotogrammetrie ist die gebräuchlichste Aufnahmeconfiguration. Ein Objekt wird mit beliebig vielen Bildern erfasst und ermöglicht so z.B. eine rundum Aufnahme oder die gleichzeitige Innen- und Außenaufnahme. Die Bestimmung der Objektkoordinaten erfolgt durch Verfahren der Bündeltriangulation. In diesem überbestimmten Gleichungssystem werden die 3D-Koordinaten, Orientierungsparameter der Bilder und weitere Modellparameter zusammen mit Angaben ihrer Genauigkeit und Zuverlässigkeit durch Ausgleichsrechnung bestimmt (Luhmann, 2003).

2.2 Structure from Motion

Der Prozess Structure from Motion (SfM) beschreibt die Erzeugung von 3D-Oberflächen aus Bildern unterschiedlicher Perspektiven. Zur Erzeugung der 3D-Oberfläche ist lediglich die Eingabe von sich überlappenden Bildern notwendig. Die Berechnung von 3D-Koordinaten, der Kamerapositionen und -orientierungen wird automatisch gelöst. Hierzu wird eine hoch redundante und iterative Prozedur der Bündelgleichung verwendet, die auf automatisch aus den Bildern extrahierten Features beruht. Features sind Merkmale in Bildern, bestehend aus einer Position und einer Beschreibung, welche invariant gegenüber Skalierung, Rotation und teils gegenüber Belichtungsänderungen sind (Westoby et al., 2012). Features können unter anderem mittels des SIFT-Algorithmus automatisch gefunden und beschrieben werden. Die Beschreibung wird durch Histogramme der Gradienten einer definierten Nachbarschaft des Merkmals realisiert (Lowe, 2004).

Aus den extrahierten Features werden u.a. die äußere Orientierung der Kameras und die 3D-Koordinaten geschätzt. Hierzu werden die Features über alle Bilder abgeglichen und Verlin-

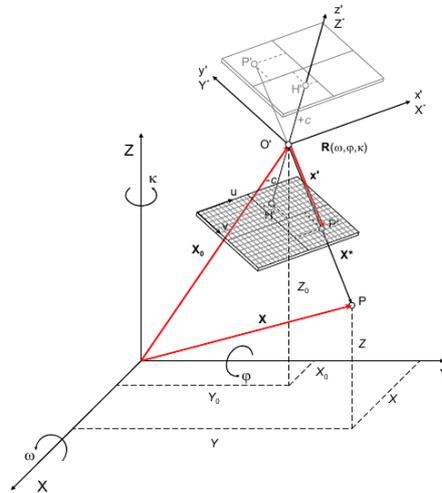


Abb. 2: Verdeutlichung der äußeren Orientierung anhand von Luhmann (2003) „Koordinatensystem“

kungen, sogenannte Tracks, zwischen gleichen Features und den zugehörigen Bildern erzeugt. Diese Tracks dienen als Zwang für die anschließende Bündelausgleichung. Das Ergebnis der Bündelausgleichung ist die Orientierung der Bilder und eine dünne 3D-Punktwolke bestehend aus den 3D-Koordinaten der Features (Westoby et al., 2012).

Ausgehend vom Ergebnis der Bündelausgleichung erzeugt das Dense Image Matching eine Verdichtung der bisherigen dünnen Punktwolke. Verschiedene Algorithmen werden in Remondino et al. (2013) vorgestellt. Im Allgemeinen erfolgt eine Zerlegung des Bildverbandes in Cluster oder Stereopaare, aus denen zunächst einzelne Punktwolken erzeugt und später zu einer gemeinsamen dichten Punktwolke zusammengefügt werden.

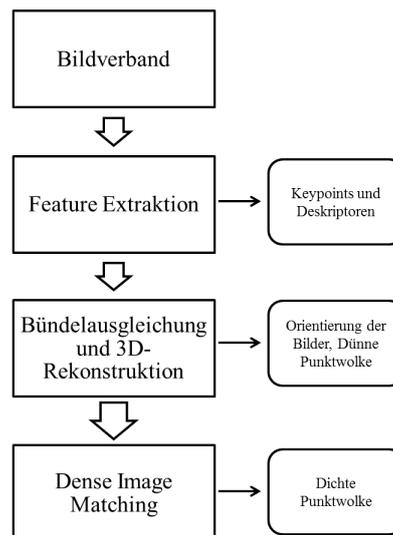


Abb. 3: Allgemeiner Ablauf von Structure from Motion

3 Der Weg von der Punktwolke zum as-built BIM

Die Erzeugung eines as-built BIM-konformen Modells kann in vier Teilaufgaben gegliedert werden: Datenaufnahme, Segmentierung, Klassifizierung, Zuordnung. Erläuterungen, aktuelle Forschungen und die eigenen Ansätze zur Umsetzung der Aufgaben sollen im Folgenden dargestellt werden.

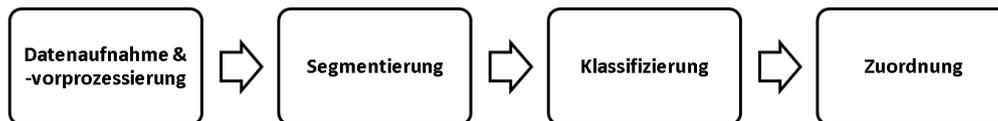


Abb. 4: Die vier Teilaufgaben zur Erzeugung eines as-built Modells aus einer 3D-Punktwolke

Datenaufnahme und Datenvorprozessierung

Der erste Schritt ist die Datenaufnahme (siehe Abschnitt 2), welche auch die erste Weiterverarbeitung der Daten beinhaltet. Diese kann die Registrierung einzelner Punktwolken sowie die Ausdünnung und Ausrichtung der Punktwolken umfassen. Die Ausrichtung der Punktwolke hat das Ziel, die Punktwolke so im Raum zu drehen, dass die Hauptachsen des Gebäudes parallel zu den Koordinatenachsen des Koordinatensystems der Punktwolke liegen. Dies kann über die iterative Analyse der Normalen der Punkte einer Punktwolke geschehen (Khosravani, 2016). Eine weitere hier verwendete Methode ist die Analyse von Schnittbildern mittels einer Hough-Transformation (Hough, 1962). Es werden Schnittbilder der Punktwolke durch ihre XY-Ebene, XZ-Ebene und durch ihre YZ-Ebene erzeugt und mittels einer Hough-Transformation Linien in den Bildern detektiert. Anhand der Winkel α der dominierenden Linien erfolgt eine iterative Drehung der Punktwolke. Anschließend wird eine höhenmäßige Ausrichtung der Punktwolke vorgenommen. Dies geschieht über die Auswertung eines Histogramms der Höhenwerte der Punkte. Die Bodenebene erzeugt das erste lokale Maximum des Histogramms. Die Punktwolke wird nun so verschoben, dass die Höhenwerte der Bodenpunkte null annehmen. Ein bereits korrigiertes Histogramm der Höhenwerte eines Raumes ist in Abb. 5 dargestellt.

Segmentierung

Im nächsten Arbeitsschritt erfolgt die Segmentierung der Punktwolke. Diese kann in die grobe und feine Segmentierung unterteilt werden. In der groben Segmentierung wird zunächst die Punktwolke in die einzelnen Räume unterteilt. In Macher et al. (2015) wird hierzu ein Binärbild aus einem Schnittbild der Punktwolke auf Deckenhöhe erzeugt. Das erzeugte Binärbild zeigt eindeutig die Umrisse der einzelnen Räume ohne Raumübergänge an, sodass die Räume über ein Region Growing bestimmt werden können. In Ochmann et al. (2014) werden Ebenen in der Punktwolke erkannt, in die Horizontalebene projiziert und miteinander verschnitten. Die Lösung von Mura et al. (2014) beruht ebenfalls auf der Extraktion von vertikalen Ebenen und der Projektion dieser in der Horizontalebene. Zur Lösung der Segmentierung werden die Schnittpunkte der projizierten Ebenen über ein angelegtes Raster gewichtet. In Abschnitt 4 Ansatz zur Segmentierung wird der konzipierte auf Histogrammen basierte Ansatz zur Segmentierung der Punktwolke durch Detektion von Raumübergängen diskutiert.

In der feinen Segmentierung wird der Raum weiter in seine Bestandteile wie Wände, Decke, Boden, Fenster und Türen segmentiert. Die Segmentierung von Boden- und Deckenpunkten

kann über die Analyse eines Histogramms der Punkthöhen (siehe Abb. 5) erreicht werden. Wandsegmente können über die Analyse eines Schnittbildes bestimmt werden (Okorn et al., 2010). Fenster und Türen können als leere Flächen innerhalb der Wandebenen detektiert werden (Adan und Huber, 2011).

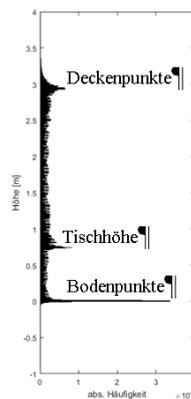


Abb. 5: Verteilung der Höhenpunkte eines Raumes

Klassifizierung

In der Klassifizierung werden die segmentierten Bestandteile Klassen zugeordnet und somit mit einer für das as-built Modell notwendigen Semantik versehen. Adan und Huber (2011) klassifizieren rechteckige Lücken in den Wandebenen mittels Support Vector Machines (SVM) zu Fenstern und Türen. (Xiong et al., 2013) erweitert diesen Ansatz und nutzt auch die Informationen von Nachbarschaften zur Klassifizierung. Eine regelbasierte Klassifizierung wird in Wang et al. (2015) vorgestellt.

Zuordnung

Im Anschluss an die Klassifikation erfolgt die Objektbildung. Hierzu werden die klassifizierten Elemente wieder zu einem Gebäudemodell, im vorliegenden Fall einem bauteilorientierten Modell, zusammengesetzt. Die Ebenen der Wandflächen einzelner Räume werden zu parametrischen Volumenmodellen zugeordnet und verschnitten.

4 Ansatz zur Segmentierung

Der konzipierte Ansatz zur Segmentierung beruht auf der Tatsache, dass sich einzelne Räume durch die Verteilung ihrer Punkte signifikant voneinander abgrenzen. Zusätzlich zur Punktwolke wird die bekannte Trajektorie der Bewegung bei der Datenaufnahme als weitere Information genutzt. Die Grundlage des Ansatzes bildet eine Grundrissdarstellung der Punktwolke durch die Erzeugung eines Schnittbildes in der Horizontalebene.

Im ersten Schritt werden von jeder Aufnahmeposition n kreisförmig angeordnete Zielstrahlen mit dem erzeugten Schnittbild verschnitten. Die X- und Y-Komponenten der Endpunkte der

Zielstrahlen werden jeweils in Histogrammen getrennt zusammengefasst. In Abb. 6 ist dies beispielhaft für 20 Zielstrahlen und dem dazugehörigen Histogramm der X-Werte dargestellt.

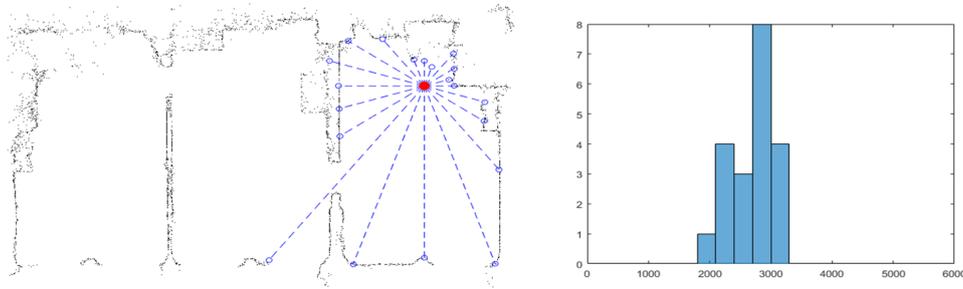


Abb. 6: Verschneidung der Zielstrahlen (*links*); Histogramm der X-Werte (*rechts*)

Für die Robustheit gegenüber Ausreißern, welche aufgrund der Streuung der Punktwolke sowie Tür- oder Fensteröffnungen entstehen können, wird nur ein Bereich betrachtet in dem 50% der Werte liegen. Hierzu werden die beiden Verteilungen der X- und Y-Werte durch statistische Lagemaße beschrieben. Diese werden durch die Quartile Q1 und Q3 realisiert.

Sequentiell werden die Quartile der einzelnen Aufnahmepositionen analysiert und übereinstimmende Histogramme zu einem Raumhistogramm addiert. Stimmt das Histogramm der aktuellen Position mit dem Raumhistogramm der bisherigen Positionen ausreichend überein, wird das Histogramm der aktuellen Position zu dem Raumdiagramm addiert und mit der nächsten Position fortgefahren. Übersteigt die Differenz zwischen aktuellem Histogramm und Raumdiagramm einen definierten Schwellenwert, wird das Abbruchkriterium erreicht. An dieser Stelle wird ein Raumübergang detektiert und ein neuer Raum begonnen. Die Wahl des Abbruchkriteriums ist somit entscheidend für die erfolgreiche Detektion von Raumübergängen. Für die Quartile Q1 und Q3 werden Konfidenzintervalle festgelegt. Der Bereich wird definiert über den Interquartilsabstand und einem Maßstab m , welcher den Bereich skaliert und maßgeblich die Schärfe der Trennung steuert. Erfahrungen aus verschiedenen Testdaten zeigen, dass für m Werte zwischen 0,5 und 0,8 gute Ergebnisse liefern.

$$[Q1_{Raum} - Interquartilsabstand_{Raum} \cdot m, Q1_{Raum} + Interquartilsabstand_{Raum} \cdot m] \quad (1)$$

Das Ergebnis der Segmentierung ist eine nach Räumen aufgeteilte Punktwolke. Zusätzlich werden auch die Raumübergänge näherungsweise bestimmt und die Aufnahmepositionen den Räumen zugeordnet

5 Ergebnisse

Getestet wurde der Ansatz in einem Szenario von drei benachbarten Büroräumen, welche über zwei Türöffnungen miteinander verbunden sind. Die durch SfM erzeugte Punktwolke wurde aus 138 Aufnahmen gerechnet. In Abbildung 7 ist die erzeugte Punktwolke und die während der Aufnahme zurückgelegte Trajektorie im Schnittbild dargestellt.

Für die Segmentierung wurden 200 Zielstrahlen pro Kameraposition genutzt und der Maßstab m mit 0,6 festgesetzt. In Abb. 8 ist das Ergebnis der Segmentierung, die bestimmten

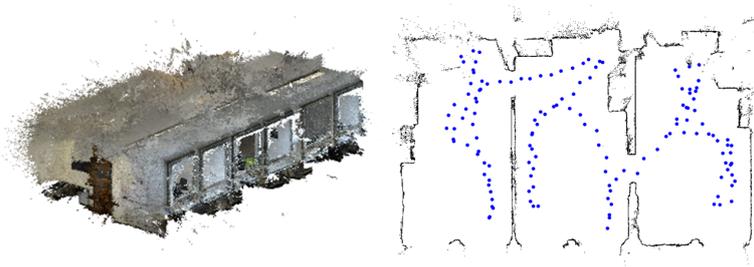


Abb. 7: erzeugte Punktwolke (*links*), Trajektorie der Aufnahme (*rechts*)

Raumhistogramme sowie das Schnittbild mit den eingefärbten Schnittpunkten, dargestellt. Die Histogramme liefern eine eindeutige Beschreibung der Räume, obwohl die Punktwolke teils verrauscht ist. Anhand der Histogramme kann so schon ein stark vereinfachtes Modell der Räume erzeugt werden.

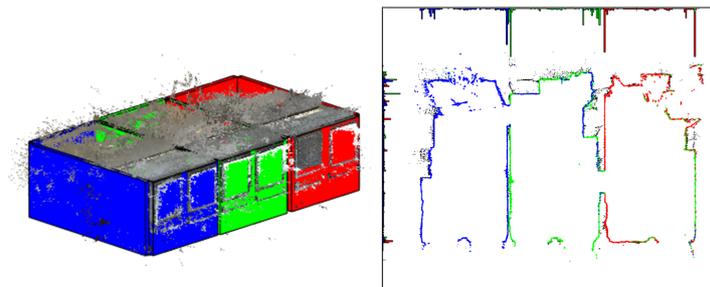


Abb. 8: Vereinfachtes Modell (*links*), Raumhistogramme und eingefärbte Schnittpunkte (*rechts*)

In Abb. 9 sind die detektierten Raumübergänge aufgetragen, mit rot für die letzte Kameraposition und grün für die erste Kameraposition eines Raumes. Anhand dieser Positionen ist eine erste Approximation der Raumübergänge möglich.

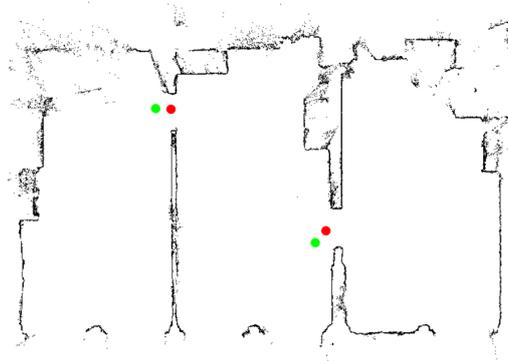


Abb. 9: Detektierte Räumübergänge (rot = letzte Position, grün = erste Position)

6 Zusammenfassung und Ausblick

Die Ergebnisse zeigen, dass sich über den konzipierten Ansatz eine Punktwolke in Räume aufteilen lässt und Raumübergängen ungefähr lokalisiert werden können. Zukünftig wird der konzipierte Ansatz hinsichtlich der Robustheit und der Flexibilität für komplexere Aufnahmesituationen weiterentwickelt. Auch ist die Kombination mit anderen bereits bestehenden Methoden denkbar. Zudem soll der Ansatz in Bezug auf die Kompatibilität mit anderen mobilen Aufnahmeverfahren, z.B. das mobile Laserscanning oder eine Plattform wie das Google Tango, getestet werden.

Literatur

- Adan, A. und Huber, D. (2011). „3D reconstruction of interior wall surfaces under occlusion and clutter“. In: *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*. IEEE, S. 275–281.
- Blankenbach, J. (2015). „Bauwerksvermessung für BIM“. In: *Building Information Modeling - Technologische Grundlagen und industrielle Praxis*. Springer, S. 247–254.
- Braunes, J. (2014). „Von der Punktwolke zum Building Information Model (BIM) Konsequenzen und Wege für die Bestandserfassung“. In: *VDVmagazin*, S. 132–136.
- Hough, P. V. (1962). *Method and means for recognizing complex patterns*. Techn. Ber.
- Khosravani, A. M. (2016). „Automatic modeling of building interiors using low-cost sensor systems“. Diss. University of Stuttgart.
- Lowe, D. G. (2004). „Distinctive image features from scale-invariant keypoints“. In: *International journal of computer vision* 60.2, S. 91–110.
- Luhmann, T. (2003). *Nahbereichsphotogrammetrie: Grundlagen, Methoden und Anwendungen*. Wichmann.
- Macher, H., Landes, T. und Grussenmeyer, P. (2015). „Point clouds segmentation as base for as-built BIM creation“. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2.5, S. 191–197.
- Mura, C., Mattausch, O., Villanueva, A. J., Gobbetti, E. und Pajarola, R. (2014). „Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts“. In: *Computers & Graphics* 44, S. 20–32.
- Ochmann, S., Vock, R., Wessel, R., Tamke, M. und Klein, R. (2014). „Automatic generation of structural building descriptions from 3d point cloud scans“. In: *Computer Graphics Theory and Applications (GRAPP), 2014 International Conference on*. IEEE, S. 1–8.
- Okorn, B., Xiong, X., Akinci, B. und Huber, D. (2010). „Toward automated modeling of floor plans“. In: *Proceedings of the symposium on 3D data processing, visualization and transmission*. Bd. 2.
- Remondino, F., Spera, M. G., Nocerino, E., Menna, F., Nex, F. und Gonizzi-Barsanti, S. (2013). „Dense image matching: comparisons and analyses“. In: *IEEE*. Bd. 1, S. 47–54.
- Volk, R., Stengel, J. und Schultmann, F. (2014). „Building Information Modeling (BIM) for existing buildings—Literature review and future needs“. In: *Automation in construction* 38, S. 109–127.
- Wang, C., Cho, Y. K. und Kim, C. (2015). „Automatic BIM component extraction from point clouds of existing buildings for sustainability applications“. In: *Automation in Construction* 56, S. 1–13.

Vom Bildverband zum as-built BIM

- Westoby, M., Brasington, J., Glasser, N., Hambrey, M. und Reynolds, J. (2012). „Structure-from-Motion photogrammetry: A low-cost, effective tool for geoscience applications: A low-cost, effective tool for geoscience applications“. In: *Geomorphology* 179, S. 300–314.
- Xiong, X., Adan, A., Akinci, B. und Huber, D. (2013). „Automatic creation of semantically rich 3D building models from laser scanner data“. In: *Automation in Construction* 31, S. 325–337.

IFC-basierte Modellierung von Tunnelbauwerksmodellen

Marcel Stepien¹, Dennis Pawlowski² und Andre Vonthron³

Informatik im Bauwesen, Ruhr-Universität Bochum, Universitätsstr. 150, 44780 Bochum

¹E-Mail: marcel.stepien@rub.de, ²E-Mail: dennis.pawlowski@rub.de,

³E-Mail: andre.vonthron@rub.de

Abstract Die Methode des Building Information Modeling (BIM) ermöglicht die digitale Begleitung von Bauwerken. Im Rahmen der OpenBIM-Initiative werden die zugehörigen digitalen Gebäudemodelle mithilfe der Industry Foundation Classes (IFC) zwischen Projektbeteiligten ausgetauscht. Enthält derzeit die IFC-Spezifikation Klassendefinitionen für Bauwerkselemente vorwiegend aus dem Hochbau, ist es mehr und mehr erforderlich BIM-Modelle für Sektoren wie den Infrastrukturbau bereitzustellen. Elemente, für die keine korrespondierenden Klassen existieren, werden gezwungenermaßen als Proxy-Elemente realisiert. Für den adäquaten Austausch von Tunnelbauwerksmodellen, erstellt durch Ringbauweise, wird ein Konzept zur Modellierung auf IFC-Basis vorgestellt. Zunächst werden fachspezifische Klassen aus dem bestehenden IFC-Schema abgeleitet. Dazu gehören die Definition von Produkt-Elementen (z. B. Tübbings) sowie die Definition räumlicher Strukturen (Ringe, Sektionen). Ebenso wird auf die Verknüpfung von Objekteigenschaften als auch des zugrundeliegenden Trassenverlaufs (IFC-Alignment) eingegangen. Neben der semantischen Modellierung wird ebenfalls die parametrisch-geometrische Konstruktion des digitalen Tunnelmodells erläutert. Zur Demonstration wird eine prototypische Implementation anhand Daten aus dem U-Bahn Projekt der Wehrhahnlinie in Düsseldorf durchgeführt.

Keywords: Tunnel Information Modeling, BIM, IFC Infrastructure, IFC Alignment

1 Einleitung

Die Anwendung der Methode des Building Information Model (BIM) erfordert die Erstellung eines digitalen Gebäudemodells (auch BIM-Modell), welches als zentrale Entität im gesamten Projektverlauf fungiert. Solche BIM-Modelle werden zwischen den Stakeholder eines Bauprojektes (Designer, Planer, Ingenieure) ausgetauscht und zur Abwicklung spezifischer Teilaufgaben benutzt. Ein offener Datenaustausch zwischen diversen BIM-Softwareprodukten findet mittels der Industry Foundation Classes (IFC) statt, welche unter Entwicklung und Vertrieb von buildingSmart (buildingSMART, 2015) stehen.

Wohingegen klassische BIM-Projekte vorwiegend im Hochbau angesiedelt sind, findet mehr und mehr eine Erweiterung auf Bereiche wie z. B. den Infrastrukturbau statt. Hier ist der Datenaustausch durch die Benutzung von IFC insofern eingeschränkt, dass keine Domänen mit Klassen zur Modellierung solcher BIM-Modelle zur Verfügung stehen. Umgangen wird diese Problematik durch die Nutzung von Proxy-Klassen, welche als generische Entitäten fungieren.

Diese können zwar Geometrie enthalten und in Relation zu Eigenschafts-Sets verknüpft werden, jedoch wird das Prinzip der semantischen Auszeichnung somit stark vernachlässigt. Einerseits sind Entitäten nicht als spezifischer Klassen-Typ ausgezeichnet (wie z. B. *IfcWall* zur Modellierung einer Wand) und besitzen demnach auch keine spezifischen Attribute und Randbedingungen. Eigenschaften können jedoch als Property-Sets verknüpft werden. Andererseits findet keine domänenspezifische Hierarchisierung des Modells in logische bzw. räumliche Strukturen statt, sondern es kann nur eine generische Hierarchisierung mit Instanzen von *IfcBuilding* und unter Verwendung verschiedener Dekompositionstypen (Complex, Partial, Element) durchgeführt werden. Um die Semantik solcher Modelle nicht zu reduzieren ist eine Erweiterung des IFC-Schemas somit erforderlich.

Im Kontext dieses Papers wird die IFC-basierte Modellierung von Tunnelbauwerken durch Ringbauweise beschrieben. Einerseits liegt hierbei der Fokus auf der Erweiterung des aktuellen IFC4-Schemas, um adäquate Klassen zur Modellierung solcher Bauwerke bereitzustellen. Dies beinhaltet die Definition von räumlichen Strukturen, physischen Elementen und zugehörigen Property-Sets. Ebenso muss ein Ansatz gefunden werden, um die zugrundeliegende Trassierung zu referenzieren. Andererseits ist neben der semantischen Modellierung auch die geometrische Implementierung eine essentielle Aufgabe zur Realisierung eines solchen Tunnelmodells. Dies beinhaltet sowohl die geometrisch-parametrische Umsetzung als auch die effiziente Organisation von repetitiven Strukturen.

Der Aufbau des Papers ist wie folgt gegliedert. Zunächst werden nötige Hintergrundinformationen vermittelt, wodurch ein Überblick über den maschinellen Tunnelbau, die IFC und bisherige Umsetzungen gegeben wird. Folgend wird das eigentliche Konzept präsentiert, bestehend aus der Definition des zugrundeliegenden hierarchischen Modells, der daraus abgeleiteten IFC Klassen und schließlich der dazu gehörenden geometrischen Modellierung. Das Konzept wird anhand einer Fallstudie umgesetzt und evaluiert. Abschließend werden die Erkenntnisse in einem Fazit zusammengefasst sowie ein Ausblick geliefert.

2 Hintergrund & Stand der Technik

Maschineller Tunnelbau

Im maschinellen Tunnelbau findet der Abtragung des Erdreiches mittels Tunnelvortriebsmaschinen (TBM) statt. An der Front der TBM ist ein Schneidrad montiert, welches für die Abtragung des Bodenmaterials zuständig ist. Direkt hinter dem Schneidrad befindet sich das Schild, welches den noch nicht fixierten Tunnel stützt. Das abgetragene Bodenmaterial wird mittels Fördertechnik im hinteren Teil der Maschine und durch den bereits erschlossenen Tunnel abtransportiert. Über diesen werden auch bereits gefertigte Betonsegmente, die sogenannten Tübbings, angeliefert. Mittels hydraulischer Pressen werden diese ringweise zur Tunnelwand zusammengepresst, wodurch sich gleichzeitig die TBM weiter in Vortriebsrichtung bewegt. Abbildung 1 illustriert dieses Prinzip.

Industry Foundation Classes

Für das Verständnis zur Ableitung von IFC-Klassen im Kontext dieses Papers wird ein kurzer Überblick über die wichtigsten Elemente der Klassenhierarchie, der relationalen Verknüpfung

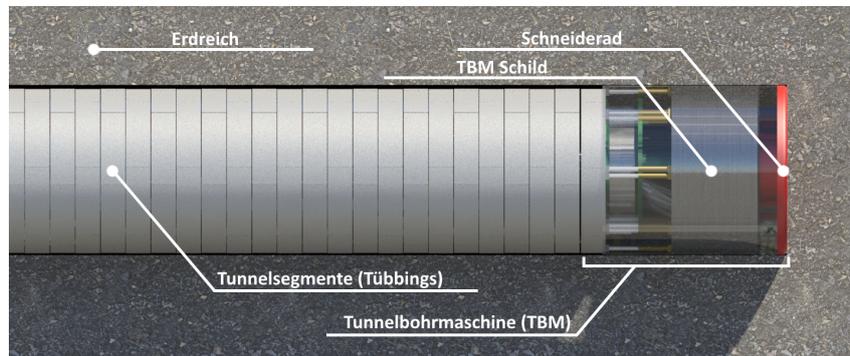


Abb. 1: Aushub des Erdreiches und Konstruktion eines Tunnels mittels einer Tunnelvortriebsmaschine (TBM)

sowie grundlegender Eigenschaften vermittelt. Die wichtigsten Basisklassen in der IFC Hierarchie sind *IfcProduct*, *IfcRelationship* und *IfcPropertyDefinition* (vgl. Abb. 2a). Entitäten vom Typ *IfcProduct* speichern räumlich-strukturelle oder geometrische Inhalte. Die Subklasse *IfcElement* stellt eine Basis für physisch existierende Objekte dar und beinhaltet üblicherweise deren Geometrie. Die Art der Geometrie kann einerseits parametrisch bedingt sein und bei Darstellung erst von einem Interpreter konstruiert werden (wie z. B. die Definition einer *IfcWall* über die Parameter Länge, Breite, Höhe usw.). Andererseits können aber auch explizite Geometrien enthalten sein, üblicherweise bestehend aus Polygon-basierten Boundary-Repräsentationen. Subentitäten vom Typ *IfcSpatialStructureElement* hingegen sind zur Definition räumlich-struktureller Bereiche angedacht. Diese können ebenfalls einen geometrischen Kontext besitzen, welche unter Umständen zur Auszeichnung des abzudeckenden Raumvolumens verwendet wird.

In Relation gesetzt werden die Produktklassen durch spezielle Entitäten vom Typ *IfcRelationship* (vgl. Abb. 2b). Zur Dekomposition diverser räumlicher Strukturen (z. B. Etagen in einem Gebäude sowie Räume pro Etage) werden die entsprechenden Instanzen vom Basis-Typ *IfcSpatialStructureElement* durch Entitäten von *IfcRelAggregates* verknüpft. Konkrete physische Elemente (z. B. eine Wand) werden schlussendlich durch eine Relation vom Typ *IfcRelContainedInSpatialStructure* in einer räumlich-strukturellen Entität verankert. Die Verknüpfung von typ- bzw. objektspezifischen Eigenschaften wird auf zwei unterschiedliche Arten umgesetzt. Einerseits sind für bestimmte Typen entsprechende Attribute gemäß IFC-Spezifikation vorhanden, da sie entweder relevant zur deren Konstruktion oder Verständnis sind. Andererseits lassen sich zusätzliche, nicht vordefinierte Eigenschaften über *IfcPropertySet*-Objekte verknüpfen, in Relation gesetzt durch *IfcRelDefines*.

IFC im Infrastrukturbau

Die Nutzung von BIM in Infrastrukturprojekten erfordert die Anpassung und Erweiterung relevanter Austauschformate auf die domänenspezifischen Anforderungen. Für die IFC führt buildingSMART dazu aktuell drei offizielle Projekte (buildingSMART, 2016b) zur Erweiterung des bestehenden Klassenmodells. Dazu gehören IFC-Alignment zur Definition von Trassierungsverläufen, IFC-Bridge für den Brückenbau, sowie IFC-Road für den Straßenbau. Weiterhin existieren noch Ansätze zur Modellierung von Tunnelbauwerken durch Schildvortrieb (Amann et al., 2013; Yabuki, 2009; Yabuki et al., 2013) sowie die Definition von IFC-Klassen für Tunnelvortriebsmaschinen (Hegemann et al., 2012).

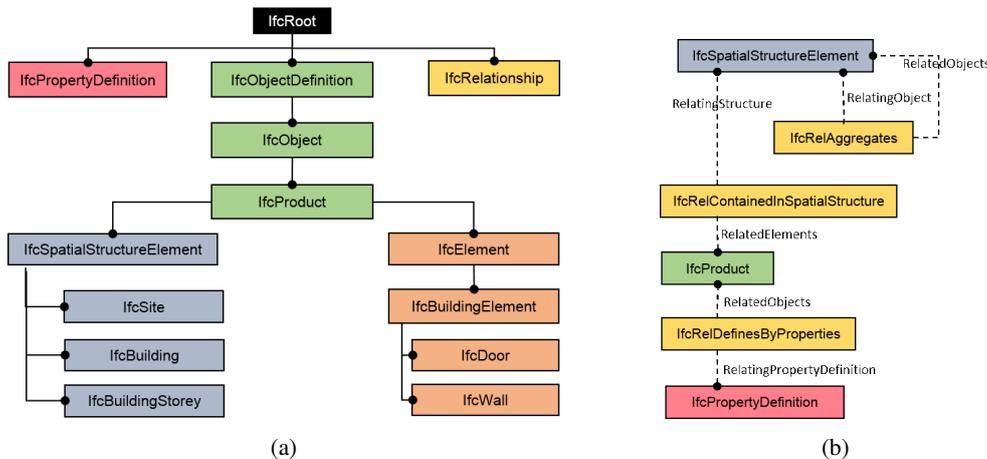


Abb. 2: (a) IFC Klassenhierarchie und (b) Relationale Modellierung, Quelle: Vonthron und König (2016)

Für den Bau von Straßen, Brücken oder Tunneln ist die Definition eines zugrundeliegenden Trassierungsverlaufs zwingend erforderlich. Während Formate wie LandXML die Modellierung von Trassierungsverläufen von Grund auf unterstützen, ist eine Integration in die IFC zunächst nicht vorgesehen. Mit der Einführung eines Produktmodells zu IFC-Bridge (Yabuki et al., 2006) wurde bereits ein in IFC vordefinierter Kurventyp benutzt. Dieser stellt eine geometrische Repräsentationsform dar, entlang welcher ein Infrastrukturbauwerk ausgerichtet werden kann. Jedoch existiert dabei keine semantische Auszeichnung der Trassierung. Amann et al. (2013) stellen bezüglich dieser Problemstellung ein IFC4-basierte Klassenerweiterung vor und demonstrieren die Umsetzung an einem Tunnelbauwerk. Eine aus diesem Konzept resultierende Fassung wurde bereits im Juli 2015 als Standard (IFC Alignment 1.0) von buildingSMART akzeptiert (vgl. Abb. 3). Demnach ist ein Alignment als Referenzierungssystem für ein Linienbauwerk zu verstehen. Dieses kann entweder als Kombination aus horizontalem und vertikalem Anteil bestehen oder direkt durch eine resultierende dreidimensionale Repräsentation angegeben werden. Einzelne Dekompositionen werden in Segmente unterteilt, welche typische Kurvenrepräsentation zur Definition von Trassierungen enthalten, wie z. B. Linie, Klothoide und Kreisbogen. Die adäquate Referenzierung des IFC-Alignment zu zugehörigen Produktmodellen gestaltet sich jedoch noch als schwierig. Als Entität im Ableitungsgraphen aus *IfcProduct* kann beispielsweise eine Referenz durch *IfcRelContainedInSpatialStructure* im obersten Strukturelement eines Produktmodells geschaffen werden (Li et al., 2016). Basierend auf der ISO/TC 211 (2012) gibt es aktuell auch Ansätze, ein IFC-Alignment in lineares Referenzierungssystem zu integrieren und als Vererbung aus *IfcObjectPlacment* direkt als Plazierungskomponente zu nutzen.

Ein Ansatz zur IFC-basierten Modellierung von Tunnelbauwerken durch maschinellen Schildvortrieb wurde erstmals in Yabuki (2009) vorgestellt. Zentrale Komponenten sind die Erweiterung aus *IfcElement* zur Repräsentation physischer Elemente sowie die Spezialisierung von Klassen aus *IfcSpatialStructureElement* zur Auszeichnung räumlicher Strukturen. Eine neuere Version der Klassendefinition wurde in Yabuki et al. (2013) veröffentlicht, in welcher einerseits zusätzliche physische Elemente als spezifische Attribute ergänzt werden. Ebenfalls Amann et al. (2013) stellen eine Ableitungshierarchie zur Definition von IFC-Klassen für die genannte Domäne vor. Diese zeichnet sich insbesondere dadurch aus, dass sie allgemeingültiger gehalten ist, d.h. nicht so tiefe Detail Klassen definiert, sondern beispielsweise in der Vererbungstiefe, in

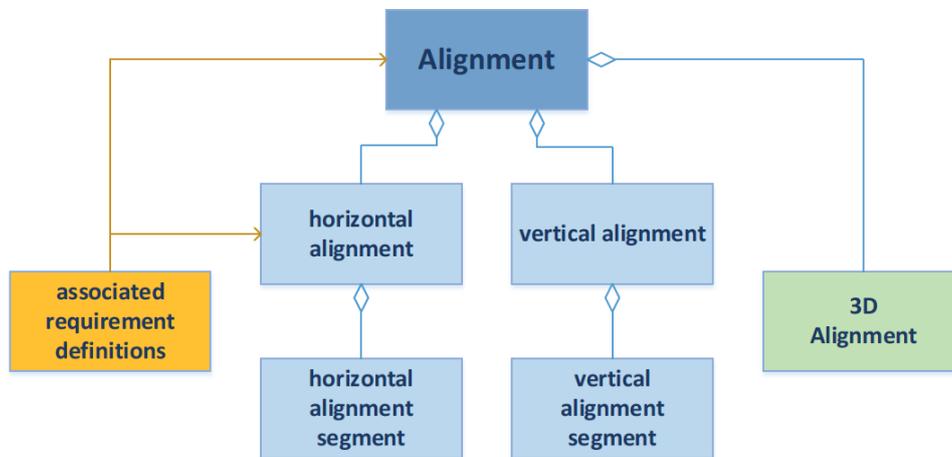


Abb. 3: IFC-Alignment Konzeptmodell (buildingSMART, 2016a)

welcher keine Attribut-basierte Unterscheidung mehr notwendig ist, die Modellierung in Form von Enumerationen anstatt Klassen vorgenommen wird.

3 Konzept

Zur Modellierung eines Tunnelmodelles auf IFC-Basis wird ein Ansatz in drei Schritten verfolgt (vgl. Abb. 4). Zunächst gilt es zu definieren, wie das zu modellierende Endprodukt hierarchisch organisiert ist. Auf Basis des hierarchischen Modells kann eine Ableitung aus IFC-Klassen für die räumliche Aufteilung als auch für konkret physisch existierende Elemente vorgenommen werden. Ebenfalls findet in diesem Schritt auch die Definition von zugehörigen Attributen und Relationen, als auch die Konzeption einer Verknüpfung zu dem zugrundeliegenden Alignment eines Tunnelbauwerkes statt. Während bis zu diesem Punkt vorwiegend semantische Details festgelegt worden sind, werden im letzten Schritt Details zur Implementierung des geometrischen Modells erläutert.



Abb. 4: Modellierungsansatz

Das Produktmodell im Kontext diese Papers unterliegt folgender hierarchischen Einteilung. An oberster Stelle steht das komplette Tunnelmodell (vgl. Abb. 5a). Der Tunnel ist unterteilt in eine zählbare Anzahl von Sektionen, welche innerhalb des kompletten Modells liegen (vgl. Abb. 5b). Innerhalb solcher Sektionen findet eine weitere räumliche Unterteilung statt. Für die Modellierung von Tunneln durch Ringbautechnik werden entsprechend Ringbereiche (vgl. Abb. 5c) definiert, welche einen räumlichen Container für die physischen Ringsegmente darstellen.

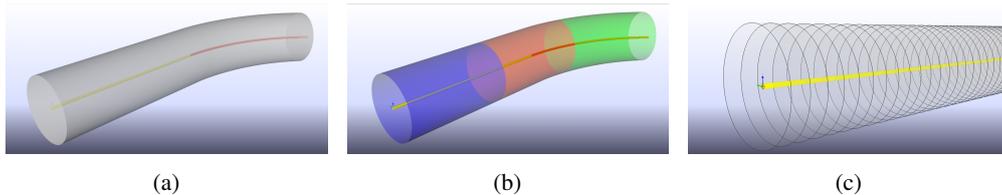


Abb. 5: Räumliche Definition

Die Ableitung zu spezifischen IFC-Klassen unterteilt sich daraus in die Definition von Klassen für räumlich-strukturelle Aufteilung (abgeleitet aus *IfcSpatialStructureElement*) sowie für die Repräsentation von physischen Elementen (abgeleitet aus *IfcElement*). Die Ableitung für die räumlich-strukturelle Definition beginnt nicht direkt aus *IfcSpatialStructureElement*, sondern es wird die bereits von Yabuki (2009) und Yabuki et al. (2006) spezialisierte Unterklasse *IfcCivilStructureElement* verwendet. Diese Klasse soll eine abstrakte Basisklasse für die Strukturierung von Elementen aus typischen Linienbauwerken wie Brücken, Straßen oder Tunnel bereitstellen.

Für die Domäne des Tunnelbaus wird eine weitere Spezialisierung auf *IfcTunnelStructureElement* vorgenommen, welche als Basis jeglicher räumlich-struktureller Dekompositionen von Tunnelstrukturen dient. Aufgrund der vielen möglichen Spezialisierung wird diese Klasse zunächst nicht abstrakt spezifiziert, sondern es ist direkt möglich eine Instanziierung vorzunehmen. Als konkrete Strukturklassen werden *IfcTunnel*, *IfcTunnelSection* und *IfcTunnelRing* definiert. *IfcTunnel* stellt die Klasse zur strukturellen Repräsentation des gesamten Tunnelbauwerkes dar (vgl. Abb. 5a). Die zu einem Tunnel zugehörige Trasse wird hier durch die Vergabe eines Attributs von der Klasse *IfcAlignment* direkt referenziert. Eine Inklusion durch *IfcRelContainedInSpatialStructure* würde erfordern, eine aufwendige Suche auf Anfrage danach unter diesen Relationen vorzunehmen. Für den ebenfalls genannten ISO-Standard existieren anwendbaren Veröffentlichungen. Die Modellierung von Sektionen innerhalb des Tunnels findet mithilfe von *IfcTunnelSection* (vgl. Abb. 5b) statt, deren Instanzen nur als Dekomposition von *IfcTunnel* existieren sollten. Für Tunnelbauwerke in Ringbauweise wird eine weitere Dekomposition mithilfe von *IfcTunnelRing* (vgl. Abb. 5c) ermöglicht. Hier werden Eigenschaften zur sequenziellen Einordnung (Ringnummer) sowie zur Beschreibung des Querschnitts (Anzahl Segmente, Durchmesser, Breite, Konizität) vergeben und über Property-Sets verknüpft. Die Übersicht der strukturellen IFC Ableitung ist in Abbildung 6 dargestellt.

Neben der Definition für die räumlich-strukturelle Aufteilung eines Tunnelmodells ist die Entwicklung eines Schemas für die Repräsentation der physikalisch existierenden Elemente erforderlich. Hierfür existiert bereits eine Basisklasse *IfcCivilElement* in der aktuellen IFC4 Spezifikation, welches zur Repräsentation von Elementen in Linienbauwerken angedacht ist. Weitergehend wird eine Spezialisierung auf *IfcTunnelElement* vorgenommen. Für die Repräsentation von Segmenten, verbaut in Ringbauweise, wird hiervon eine Klasse *IfcTunnelSegment* abgeleitet. Als Eigenschaften werden Segmenttyp, Material, Gewicht und Volumen definiert und als Property-Set verknüpft. Neben der Klassendefinition zur Modellierung von Segmenten



Abb. 6: Definition von strukturellen Klassen

findet ebenfalls die Definition der Klasse *IfcTunnelLining* statt, welche eine generelle Modellierung einer Tunnelschale ermöglicht. Dies kann entweder ringweise, für eine Sektion oder den ganzen Tunnel erfolgen. Die räumlich-strukturelle Einordnung findet darauf basierend entweder innerhalb *IfcTunnelRing*, *IfcTunnelSection* oder *IfcTunnel* statt. Eine Übersicht der Klassenspezifikation ist in Abbildung 7 dargestellt.

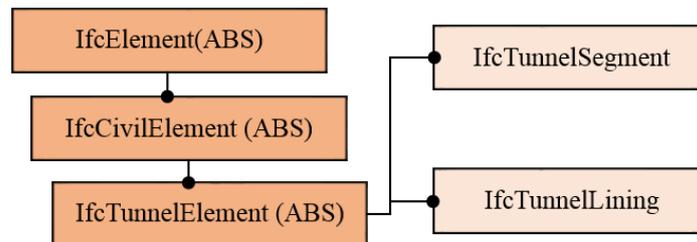


Abb. 7: Ableitung der physikalischen IFC Tunnel-Klassen

Neben der Ableitung spezifischer Klassen wird ebenfalls das zugehörige Konstruktionschema für die konkrete Instanziierung vorgestellt (vgl. Abb. 8). Üblicherweise existiert in jedem IFC-Modell ein Projekt (*IfcProject*), welches mindestens eine Baustelle (*IfcSite*) enthält. Der strukturelle Aufbau eines Tunnels (*IfcTunnel*) wird beginnend auf einer solchen Baustelle begonnen. Die zum Tunnel zugehörige Trassierungsinformation (*IfcAlignment*) wird hier direkt als Attribut referenziert. Die weiteren Unterteilungen finden im Folgenden mittels Tunnelsektionen (*IfcTunnelSection*), sowie darin befindliche Tunnelringebereiche (*IfcTunnelRing*) statt. Die zuvor genannte strukturelle Unterteilung wird durch *IfcRelAggregates*-Relation umgesetzt. Diese modellieren jeweils eine 1:N-Beziehung (z. B. 1 Tunnelsektion aggregiert N Tunnelringe). Durch den daraus entstehenden Graphen lässt sich das entstehende IFC Modell traversieren und ein hierarchischer Projektbaum ableiten.

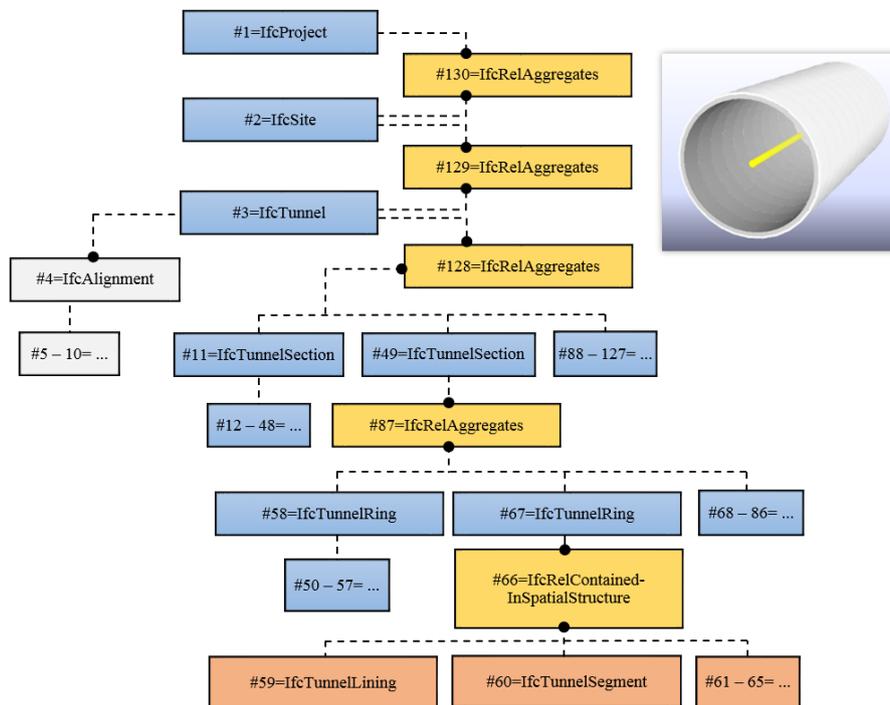


Abb. 8: Konstruktionsschema

Physische Elemente zur Repräsentation eines Ringsegments (*IfcRingSegment*) werden mittels Relationen vom Typ *IfcRelContainedInSpatialStructure* verlinkt. Die Modellierung weiterer Auskleidungskomponenten der Tunnelschale (*IfcTunnelLining*) kann per Definition entweder pro Sektion oder pro Ring erfolgen. Eine Einordnung in die Modellstruktur findet, unabhängig von der hierarchischen Definition, ebenfalls mit einer Relation vom Typ *IfcRelContainedInSpatialStructure* statt.

Die Definition der konkreten Geometrie kann in den IFC auf zwei Weisen erfolgen. Einerseits kann Geometrie implizit angegeben werden, beispielsweise als Constructive Solid Geometry (CSG) oder Non-Uniform Rational Basis Splines (NURBS). Diese muss dann zur Darstellung zwangsweise von einem Interpreter in explizite Geometrie umgewandelt werden. Vorteile sind z. B. die Wahl eines konkreten Detaillevels sowie der geringe Speicherplatzbedarf von impliziten Definitionen. Jedoch kann der Interpretationsprozess impliziter Geometrien u. U. einen hohen Bedarf an Rechenzeit erfordern. Explizite Geometrien hingegen, meistens polygonale Oberflächenrepräsentationen, können ohne weiteren Interpretationsprozess dargestellt werden, besitzen jedoch ein festes Detaillevel und können zu hohen Speicherplatzbedarf führen.

Ausgehend von vordefinierten Geometrien für die einzelnen Ringsegmente müssen diese im Modell positioniert werden. Die Positionierung erfolgt auf Basis lokaler Koordinatensysteme. Hervorgehend aus der Klassenhierarchie von *IfcProduct* besitzen sowohl Instanzen von *IfcElement* als auch *IfcSpatialStructureElement* die Möglichkeit, ein lokales Koordinatensystem über ein *IfcObjectPlacement* zu definieren. Nach diesem Prinzip findet die Platzierung der Instanzen von *IfcTunnelSegment* innerhalb eines Ringes statt. Ringe werden wiederum innerhalb einer Tunnelsektion lokal positioniert, sowie Tunnelsektionen im Koordinatensystem des Tunnels selbst positioniert werden. Für die beiden letzten Fälle muss zunächst das zugrun-

deliegende *IfcAlignment* zu einer Polyline approximiert werden. Dies wird typischerweise in Breite eines Tunnelrings vorgenommen. Eine Sektion sollte genau ein Vielfaches der Tunnelringbreite abdecken, sodass die Punkte der Polyline ebenfalls zur dessen Positionierung benutzt werden können. Nach Ausführen der Approximation werden jeweils die Vektoren zwischen zwei aufeinanderfolgenden Punkten dazu benutzt, um die Ringe im lokalen Koordinatensystem des Tunnels auszurichten (vgl. Abb. 9).

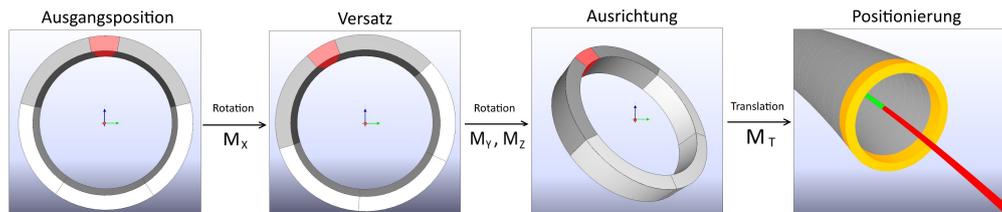


Abb. 9: Ausrichtung der Ringsegmente und Platzierung des Rings im Koordinatensystem einer Tunnelsektion

Neben der Positionierung der Segment-Geometrien ist ebenfalls deren effiziente Instanziierung und Verlinkung von essentieller Bedeutung für die Anwendbarkeit des Modells. Die hier verwendeten Tunnel sind prädestiniert zur Erzeugung redundanter Geometrien resultierend aus der mehrfachen Instanziierung von sich wiederholenden geometrischen Definitionen. Zur Vermeidung redundanter Inhalte sollte nach dem Flyweight-Design-Pattern konstruiert werden, wie in Vonthron und König (2016) für die Verwendung mit IFC Modellen aufgezeigt worden ist. Durch die Trennung von Semantik und Geometrie kann dies im Kontext des Papers leicht umgesetzt werden. Der geometrische Inhalt in aus *IfcProduct* abgeleiteten Instanzen wird mittels Referenzen auf eine geteilte *IfcProductRepresentation* verlinkt. Für Segmente, welche den gleichen geometrischen Aufbau besitzen, wird somit nur eine einzige geometrische Repräsentation instanziiert und durchgehenden über alle Tunnelsegmente referenziert. Semantischer Inhalt und spezifische Platzierung für jedes *IfcProduct* hingegen werden getrennt betrachtet.

4 Fallstudie

Für die Demonstration des vorgestellten Konzepts findet eine Implementierung anhand einer konkreten Fallstudie statt. Dazu werden Trassierungsdaten sowie Querschnittsparameter aus dem U-Bahn-Projekt „Wehrhahnlinie“ in Düsseldorf, Deutschland, herangezogen. Zur Implementierung werden die IFC Tools Project (Tauscher, 2016) verwendet. Diese bieten eine Java-basierte IFC-Schnittstelle, welche nach dem Early-Binding Verfahren funktioniert. Hierbei werden für ein definiertes IFC Schema äquivalente Java-Klassen produziert und können somit als Java-Objekte instanziiert und direkt angesprochen werden. Neben diesem Kernel bieten die Tools außerdem integrierte Anzeigen zur Darstellung der Geometrie (*3D View*) sowie von räumlichen Strukturen (*Spatial Structure View*) und Objekteigenschaften (*Property View*).

Die erwähnten Projektdaten kommen aus einem Projektstatus nach der Fertigstellung des Tunnels. In diesem existiert die Trassierung nur als Folge von Punkten, welche genau in Ringabstand angeordnet sind. Die Trassierung wird somit direkt als 3D-Alignment modelliert und stellt bereits eine Approximation zur Ausrichtung von Ringsegmenten dar. Die geometrische Modellierung findet hier auf Basis expliziter Geometrie statt. Pro Segment wird eine Boundary-Repräsentation erstellt, deren Oberflächen aus eine Menge von Dreiecken konstruiert sind. Das resultierende BIM-Modell ist in Abbildung 10 dargestellt. Neben der Anzeige der Geometrie

können ebenso die Struktur des Modells als auch die verlinkten Eigenschaften eingesehen werden. Der *Spatial Structure View* zeigt die strukturelle Dekomposition des Tunnels in Sektionen und jeder Sektion in Ringe. In den Ringen selbst sind Segmente und weitere Ausschaltungsgeometrie enthalten. Werden konkrete Elemente im *3D View* selektiert, so lassen sich verlinkte Eigenschaften im *Property View* anzeigen. Für ein Tunnelsegment werden hier z. B. Eigenschaften wie Breite und Gewicht vermittelt. Ebenso lassen sich aber auch Eigenschaften für räumlich-strukturelle Komponenten über eine individuelle Selektion im *Spatial Structure View* einsehen.

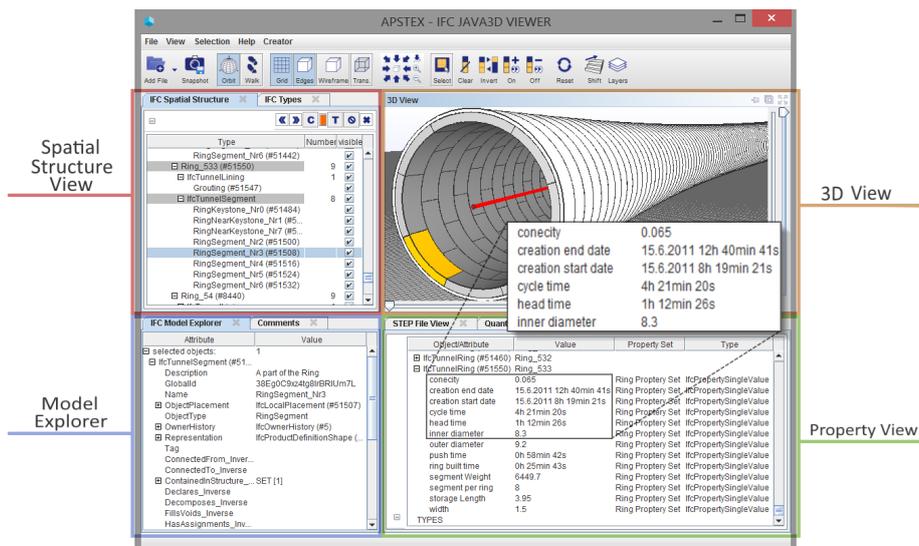


Abb. 10: Darstellung des erstellten BIM-Modells des Tunnels im APSTEX IFC JAVA3D VIEWER (Tauscher, 2016)

Da das Tunnelbauwerk kontinuierlich aus identischen aufgebauten Tunnelringen besteht, konnten die Geometrien der unterschiedlichen Segmenttypen nach dem Flyweight-Design-Pattern durchgehend für jeden Ring referenziert werden. Dies führt zu einer sehr geringen Größe für die resultierende IFC Datei von 4.1 Megabytes(MB) (vgl. die ohne Flyweight-Pattern erzeugte IFC Datei resultiert in 149 MB).

5 Fazit

Für den offenen Datenaustausch auf IFC-Basis im Rahmen des Building Information Modeling ist die Betrachtung nicht klassischer Anwendungsdomänen zunehmend von Bedeutung. Die Anwendung im Infrastrukturbau verzeichnet hierbei gerade in Deutschland ein enormes Wachstum. Angewandt auf dem maschinellen Tunnelbau ist ein Konzept zur IFC-basierten Modellierung von Tunnelbauwerken in Ringbauweise vorgestellt worden. Auf der einen Seite wurden fachspezifische Klassen abgeleitet, aufgeteilt in Definitionen für strukturelle und für physische Repräsentation. Auf der anderen Seite ist beschrieben worden, wie auf Basis einer zugrundeliegenden Trassierung die geometrisch-parametrische Umsetzung stattfindet. Demonstriert worden ist das Konzept anhand einer konkreten Fallstudie. So konnte gezeigt werden, dass neben der geometrischen Komponente auch die semantischen Modellierungs-Anforderungen der Fachdomäne erfüllt worden sind.

Das bestehende Konzept enthält keine Methodik, um Tunnelsektion über bestimmte Trassierungsbereiche parametrisch zu referenzieren, um z. B. logische Zwischenbereiche festzulegen. Hier wird eine Erweiterung des IFC-Alignment und der Aufbau einer entsprechenden Relation zu *IfcTunnelSection* angestrebt. Weiterhin zeigen aktuelle Tunnelprojekte unter der Anwendung von BIM, dass neben der Modellierung im Rohbau auch eine IFC-basierte Modellierung im Ausbau an Bedeutung gewinnt. Hier gilt es ebenfalls die Ableitung spezifischer Klassen in einem geeigneten Detaillevel in Betracht zu ziehen.

Literatur

- Amann, J., Borrmann, A., Hegemann, F., Jubierre, J. R., Flurl, M., Koch, C. und König, M. (2013). „A refined product model for shield tunnels based on a generalized approach for alignment representation“. In: *Proc. of the ICCBEI*.
- buildingSMART (2015). *IFC Overview Summary*. <http://www.buildingsmart-tech.org/specifications/ifc-overview> (besucht am 15.06.2016).
- buildingSMART (2016a). *Alignment Overview*. <http://www.buildingsmart-tech.org/infrastructure/projects/alignment> (besucht am 01.07.2016).
- buildingSMART (2016b). *Ongoing Projects*. <http://www.buildingsmart-tech.org/infrastructure/projects> (besucht am 18.08.2016).
- Hegemann, F., Lehner, K. und König, M. (2012). „IFC-based product modeling for tunnel boring machines“. In: *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2012*, S. 289–296.
- ISO/TC 211 (2012). *Geographic information – Linear referencing vom 15.02.2012*. Veröffentlichungsnr.: ISO 19148:2012.
- Li, X., Lin, X., Zhu, H., Wang, X. und Liu, Z. (2016). „A BIM/GIS-based management and analysis system for shield tunnel in operation“. In: 16th International Conference on Computing in Civil and Building Engineering. Osaka.
- Tauscher, E. (2016). *IFC Tools Project*. <http://www.ifctoolsproject.com/> (besucht am 01.07.2016).
- Vonthron, A. und König, M. (2016). „Modeling of repetitive IFC Building Elements using the Flyweight Design Pattern Approach“. In: 16th International Conference on Computing in Civil and Building Engineering. Osaka.
- Yabuki, N. (2009). „Representation of caves in a shield tunnel product model“. In: *Proc. of the 7th European Conference on Product and Process Modelling, Sophia Antipolis, France*, S. 545–550.
- Yabuki, N., Aruga, T. und Furuya, H. (2013). „Development and application of a product model for shield tunnels“. In: *Proc. of the 30th International Symposium on Automation and Robotics in Construction, Montréal, Canada*.
- Yabuki, N., Lebegue, E., Gual, J., Shitani, T. und Zhantao, L. (2006). „International collaboration for developing the bridge product model IFC-Bridge“. In: *Proc. of the 11th Int. Conf on Computing in Civil and Building Engineering*.

IFC-Daten für die thermische Gebäudesimulation

Andreas Geiger^{1a}, Iris Reichenbach^{2b} und Karl-Heinz Häfele^{3a}

^a*Institut für Angewandte Informatik, Karlsruher Institut für Technologie, Postfach 3640, 76021
Karlsruhe, Deutschland*

¹*E-Mail: andreas.geiger@kit.edu, ³E-Mail: karl-heinz.haefele@kit.edu*

^b*Hottgenroth Software, Von-Hünefeld-Straße 3, 50829 Köln, Deutschland
²E-Mail: i.reichenbach@hottgenroth.de*

Abstract Die gesetzlichen Anforderungen an Energieeffizienz von Wohn- und Nichtwohngebäuden haben in den vergangenen Jahrzehnten kontinuierlich zugenommen. Dies betrifft nicht nur den Neubau, sondern insbesondere die Modernisierung von Bestandsimmobilien. Der Einsatz von Simulationsprogrammen zur Analyse und Bewertung von Gebäuden spielt mittlerweile eine entscheidende Rolle und wird in allen Bereichen zur Unterstützung des Bauplanungsprozesses eingesetzt.

buildingSMART propagiert mit einem modellbasierten Ansatz, dem Building Information Modeling (BIM), eine ganzheitliche Beschreibung von Gebäuden (inklusive der Beschreibung von Prozessen und Ressourcen) und bietet mit dem internationalen Standard IFC (Industry Foundation Classes – ISO 16739) die Möglichkeit Daten modellbasiert auszutauschen. Insbesondere Simulationswerkzeuge können von diesem modellbasierten Ansatz profitieren.

Üblicherweise arbeiten Simulationsprogramme mit einem Ersatzmodell, in dem die Bauteile eines Gebäudes durch eine vereinfachte Flächenbeschreibung und physikalische Parameter abgebildet werden. Dieses Ersatzmodell kann entweder aus der IFC-Geometrie der Gebäudeelemente (i.d.R. ein Volumenmodell) abgeleitet werden, oder es wird, falls vorhanden, aus der Relation zwischen Raum und Gebäudeelement (Space Boundaries - Raumbegrenzungsflächen) erzeugt. Die für eine Simulation zusätzlich notwendigen Parameter lassen sich aus den Bauteilparametern und den Bauteilbeziehungen (Entity-Relationship-Model) (Eastman, 1999) ableiten.

Der hier vorgestellte Beitrag beschreibt das in IFC definierte Konzept der Raumbegrenzungen und deren Einsatz als Ausgangsbasis für die thermische Simulation nach der Richtlinie VDI 6007. Dabei wird diskutiert, inwieweit die Definitionen von IFC2x3 ausreichend sind und welche erweiterten Möglichkeiten die neue Version IFC4 bietet. Die praktische Umsetzung wird anhand von Softwaretools der Firma Hottgenroth Software aufgezeigt und durch Beispiele veranschaulicht.

Keywords: IFC, BIM, Gebäudemodell, Thermische Gebäudesimulation

1 Einleitung

Mit der steigenden Leistungsfähigkeit von Computern wurde die numerische Simulation des Energiebedarfs komplexer Gebäude möglich und für den Planungsalltag immer bedeutsamer. So setzt zum Beispiel die Neufassung der DIN 2078 auf eine Simulation des Kühlbedarfs, eine tabellarische, vereinfachte Rechnung ist nur noch unter bestimmten Bedingungen zulässig. Dies ist auch sinnvoll, da die Gebäude-Simulation präziser ist, leichter Spezialfälle abbilden kann und, im Gegensatz zu der Vielzahl nationaler Normen zur Berechnung des Energiebedarfs, allgemeingültig ist. Daher können Simulationsprogramme auch ganz besonders von einem internationalen Standard wie IFC profitieren.

Allerdings stellen Simulationsprogramme eigene Ansprüche an Qualität und Inhalte der zu verarbeitenden Daten. In diesem Beitrag sollen diese Ansprüche anhand einer konkreten Simulationssoftware der Firma Hottgenroth Software verdeutlicht, sowie die Möglichkeiten des IFC-Standards zur Unterstützung thermischer Simulationen diskutiert werden. Dazu werden im ersten Abschnitt die Anforderungen einer energetischen Simulation an das zugrunde liegende Gebäudemodell erläutert, sowie die Unterschiede zwischen Architektur- und Simulationsmodellen verdeutlicht. In weiteren Abschnitten werden die verschiedenen Arten der Raumbegrenzungen in IFC vorgestellt, sowie allgemein die Unterstützung thermischer Gebäudesimulationen durch die aktuell in Softwareprodukten umgesetzte Version IFC2x3 im Vergleich zur zukünftigen Version IFC4 aufgezeigt.

Prototypisch realisiert wird dies in der Software IFCEXplorer des Karlsruher Instituts für Technologie (KIT), die Eingabedatensätze für die Simulationssoftware der Firma Hottgenroth Software aus IFC-Modellen generiert. Hierbei wird ein besonderes Augenmerk auf den Umgang mit den vom IFCEXplorer gelieferten Rohdaten im Hinblick auf verschiedene Normen für Energiesimulationen auf Basis von Gebäudemodellen gelegt. Eine Zusammenfassung beendet den Beitrag.

2 Geometrische Anforderungen an Gebäudemodelle für energetische Simulationen

Die Anforderungen an ein Gebäudemodell, das für eine energetische Simulation verwendet werden soll sind sehr spezifisch und unterscheiden sich grundlegend von den Anforderungen an reine Architekturmodelle. Diese Unterschiede beziehen sich sowohl auf die geometrische Repräsentation, als auch auf den Detaillierungsgrad der verschiedenen Parameter die als Eingangswerte für eine Simulation benötigt werden. Hierzu zählen unter anderem detaillierte Materialeigenschaften, Nutzungsprofile, sowie die Definition von thermischen Zonen.

Während bei einem Architekturmodell der Schwerpunkt auf der geometrisch korrekten Darstellung über eine volumetrische Beschreibung der Bauteile liegt, nutzen die meisten Simulationswerkzeuge eine vereinfachte Flächenbeschreibung. Diese beschreibt entweder die Kontaktflächen zwischen den physikalischen Bauteilen wie Wände, Decken usw. und den Räumen (siehe Abbildung 1a) oder ein Bauteil wird – wie am Beispiel einer Innenwand – nur durch eine Fläche abgebildet (siehe Abbildung 1b). Eine weitere Besonderheit stellen Öffnungen wie Fenster und Türen dar. Diese Elemente werden nicht wie in einem Architekturmodell herausgeschnitten, sondern zusätzlich zu den Bauteil-Bruttoflächen repräsentiert.

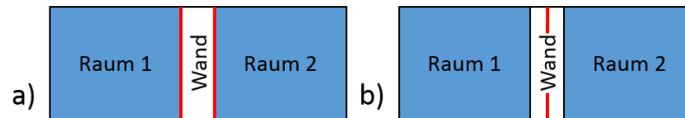


Abb. 1: Unterschiedliche flächenhafte Repräsentationen einer Wand: (a) Repräsentation durch die Kontaktflächen Raum – Wand; (b) Repräsentation durch eine Fläche in Mitte der Wand

Für die energetische Betrachtung eines Gebäudes sind die Räume das zentrale Element. Ziel ist, eine möglichst genaue Abbildung des Wärmeaustausches zwischen einzelnen Räumen bzw. zwischen Außenräumen und der Umgebung zu realisieren. Dieser Wärmeaustausch erfolgt in Form von Wärmestrahlung, Konvektion und Diffusion durch die raumbegrenzenden Bauteile. Aus diesem Grund werden für Räume die topologischen Beziehungen zu angrenzenden Bauteilen benötigt, um so den Energieaustausch korrekt abbilden zu können.

3 Raumbegrenzungselemente in IFC

buildingSMART stellt mit dem Datenschema IFC ein allgemeines Austauschformat für Softwareprodukte im Baubereich bereit. Es erlaubt die Betrachtung eines Gebäudes über alle Phasen des Lebenszyklus. Neben der üblicherweise verwendeten volumetrischen Abbildung der Bauteile unterstützt IFC auch das Konzept von Raumbegrenzungselementen zur Beschreibung der Nachbarschaftsbeziehungen zwischen Räumen und Bauteilen.

Eingeführt wurde dieses Konzept mit der Version IFC2.0 durch einen Vorschlag von Jiri Hietanen im Rahmen des BLIS-Projektes (Building Lifecycle Interoperable Project) (BLIS-Project, 2002), durch das eine Reihe von Erweiterungen in die IFC Entwicklung eingeflossen sind. In dieser initialen Fassung wurde hierzu das Entity *IfcSpaceBoundary* eingeführt (Hietanen, 1999). Mit der Version IFC2x wurde daraus dann das heutige Relationskonzept zur Abbildung der Beziehung zwischen Räumen und angrenzenden Bauteilen mit dem Entity *IfcRelSpaceBoundary* (siehe Abbildung 2). Durch dieses Konzept lassen sich alle Entitäten, die in der Vererbungshierarchie von *IfcElement* abgeleitet sind, mit einem Raum (*IfcSpace*) verknüpfen. *IfcElement* ist eine abstrakte Basisklasse, die unter anderem für alle Bauteile, die

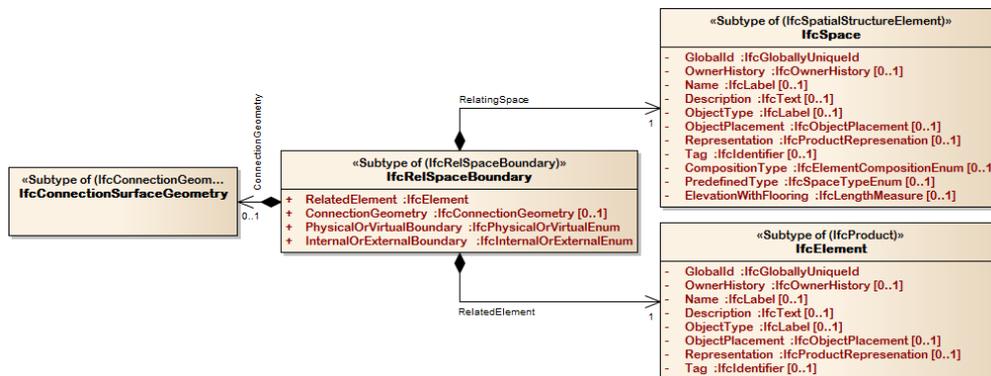


Abb. 2: UML Modell des Entity *IfcRelSpaceBoundary*

maßgeblich ein Gebäude beschreiben (*IfcBuildingElement*), für alle haustechnischen Komponenten (*IfcDistributionElement*) und auch für Einrichtungsgegenstände (*IfcFurnishingElements*) verwendet wird. Neben den beiden Attributen für die Nachbarschaftsbeziehung lässt sich optional die Berührungsfläche zwischen Bauteil und Raum geometrisch beschreiben. In Anhängigkeit davon, wie diese Berührungsfläche gebildet wird, werden Berührungsflächen 1. und 2. Art (1st und 2nd Level Space Boundaries) unterschieden. Weiterhin verfügt das Entity *IfcRelSpaceBoundary* über zwei verpflichtende Attribute, die beschreiben, ob es sich um eine Beziehung zu einem Innen- oder Außenbauteil und um eine physikalische oder eine virtuelle Beziehung handelt. Von virtuellen Begrenzungselementen spricht man z. B. im Fall von aneinander grenzenden Räumen ohne trennendes physikalisches Bauteil, oder wenn es keine direkte Berührung zwischen einem Raum und einem Bauteil gibt (z. B. der Raum kleiner als das durch die umgrenzenden Bauteile umschlossene Volumen oder nach außen nicht begrenzt ist). Bei der Verwendung von Räumen für die thermische Simulation muss berücksichtigt werden, dass diese im Normalfall aus Sicht der Architektur definiert sind. Bereiche mit gleichen thermischen Eigenschaften, wie sie durch virtuelle Begrenzungen als Raumteiler entstehen (z. B. Trennung Wohn-Essbereich), werden nicht berücksichtigt.

Die Definition der Raumbegrenzungen von IFC unterstützt aktuell zwei Konzepte für unterschiedliche Anwendungsfälle. Dabei spricht man von 1st und 2nd Level Space Boundaries:

- **1st Level Space Boundary (Abb. 3a):**

Bei diesem Typ von Raumbegrenzungen wird die Kontaktfläche Raum – Bauteil nur aus Sicht dieses Raumes, d.h. ohne Berücksichtigung eventuell auf der anderen Bauteilseite angrenzender Räume, gebildet. So würde beispielsweise die Längsseite eines Flures, von dem verschiedene Räume abgehen, nur durch eine einzige Space Boundary repräsentiert werden. Diese Begrenzungselemente kommen insbesondere bei der Mengenermittlung und dem Facilitymanagement zum Einsatz. Für die Energiesimulation sind diese Beziehungen nicht geeignet.

- **2nd Level Space Boundary (Abb. 3b):**

Bei diesem Typ von Raumbegrenzungen werden Räume auf der anderen Bauteilseite mit berücksichtigt. Die Kontaktflächen zwischen Raum und Bauteil werden so aufgeteilt, dass es eine eindeutige geometrische Zuordnung der Space Boundaries auf beiden Seiten eines Bauelements gibt. Diese Begrenzungselemente können für alle gängigen Arten von Simulationen als Ausgangsbasis verwendet werden. Sie beschreiben die Bauteile aus Sicht des Raumes und berücksichtigen gleichzeitig Beziehungen zu benachbarten Räumen.

Neben diesen beiden Definitionen taucht auch immer wieder der Begriff „3rd Level Space Boundaries“ auf. Diese Bezeichnung beschreibt die Situation, bei der „hinter“ der Begrenzung kein Raum sondern ein Bauteil liegt. Damit wird der Wärmeübergang zu diesen Bauteilen in einigen Systemen gesondert betrachtet. Diese Definition gibt es offiziell noch nicht in IFC und wird hauptsächlich im Zusammenhang mit Energy+ (Bazjanac, 2010) verwendet. Mit einer geometrischen Analyse der Raumbegrenzungselemente des 2nd Level lässt sich diese Information bei Bedarf auch ermitteln.

Abbildung 3 zeigt im linken Bild ein Beispiel für 1st Level Space Boundaries und im rechten Bild die 2nd Level Space Boundaries. Die rot markierte Fläche ist ein Beispiel für den Typ 3rd Level Space Boundary.

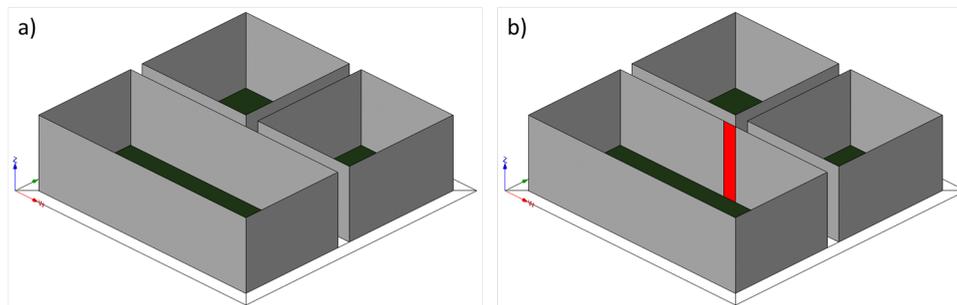


Abb. 3: Unterschiede zwischen 1st/2nd Level Space Boundaries

3.1 IFC 2x3 - Status quo

Für die Verwendung von Raumbegrenzungselementen gibt es aktuell zwei Implementierungs-Richtlinien die sich auf IFC2x3 beziehen. Der erste Leitfaden „IFC2x3 Space Boundary Implementation Summary“ (Häfele und Liebich, 2009/2010) beschreibt allgemein den Aufbau und die zu verwendenden Geometrien und diente als Ausgangsbasis für das durch die GSA (US Government Services Administration) in Auftrag gegebene Handbuch „IFC Implementation Guide – Space Boundaries for Thermal Analysis“ (Weise et al., 2009). Darin richtet sich der Schwerpunkt gezielt auf die Anwendung für die thermische Analyse.

Die aktuelle Version von IFC bietet mit den Raumbegrenzungselementen (Borrmann et al., 2015) bereits heute schon eine gute Ausgangsbasis für Simulationssysteme. Mit ca. 35 zertifizierten Produkten ist ein breites Spektrum von erzeugenden Systemen vorhanden. Dabei muss allerdings berücksichtigt werden, dass das Entity *IfcRelSpaceBoundary* nicht Teil des zertifizierten *CoordinationView 2.0* ist und auch nicht alle Systeme in der Lage sind, Raumbegrenzungselemente zu erzeugen. Bei der sogenannten *Space Boundary Add-On View* handelt es sich um eine Erweiterung zu IFC2x3 *CoordinationView 2.0*, die auch im Header einer IFC-Datei SPF (STEP Physical File) entsprechend angegeben werden muss. Hierzu muss, neben der Angabe des *CoordinationView 2.0*, zusätzlich eines der Schlüsselwörter *SpaceBoundary1stLevelAddOnView* oder *SpaceBoundary2ndLevelAddOnView* verwendet werden, um zu signalisieren um welche Art der Raumbegrenzungselemente es sich handelt. Dies ermöglicht es dem empfangenden System, zu entscheiden, ob der Datensatz auch die erwarteten Informationen enthält. Neben der Information im Header wird zusätzlich in den Instanzen von *IfcRelSpaceBoundary* im Attribut *Name*, durch die Angabe von „1stLevel“ bzw. „2ndLevel“, der entsprechende Level festgelegt.

Geometrisch wird eine Berührungsfläche in IFC durch das Entity *IfcConnectionSurfaceGeometry* abgebildet. Dabei kann es sich entweder um eine einzelne Fläche (*IfcCurveBoundedPlane*) oder um die resultierende Fläche einer linearen Extrusion (*IfcSurfaceOfLinearExtrusion*) handeln. Diese beschreibt einen Linienzug, bestehend aus Geometrieobjekten vom Typ Linie, Kreis bzw. Ellipse, der extrudiert wird. Auf diese Art lassen sich beispielsweise gebogene Wände sehr kompakt beschreiben, und die Zerlegung solcher Berührungsflächen in eine Vielzahl einzelner Flächenstücke wird vermieden.

4 IFC4 – Erweiterte Funktionalität

Mit der neuen Version IFC4 (MSG, 2014) wurde eine Vielzahl an Erweiterungen und Verbesserungen, auch in Bezug auf eine verbesserte Datenbasis für Simulationssysteme, eingeführt.

In Simulationen spielt die geographische Lage eines Gebäudes eine zentrale Rolle. Daher wurde in IFC4 die Funktionalität in diesem Bereich grundlegend erweitert. Es ist nach wie vor noch möglich, die Attribute des *IfcSite* Objekts für die geographische Länge und Breite zu verwenden. Da aber in den meisten Ländern unterschiedliche Lagesysteme eingesetzt werden, können diese auch entsprechend angegeben werden. Besonders wenn es um Nachbarschaftsbeziehungen zwischen Gebäuden geht, ist dies auch für die Simulation relevant.

Da es sich um eine entscheidende Information handelt, welcher Typ von Raumbegrenzung (1st Level / 2nd Level) verwendet wird, hat man sich entschlossen, hierfür eigene EXPRESS Entities zu definieren. Aus diesem Grund wurden die Entities *IfcRelSpaceBoundary1stLevel* und *IfcRelSpaceBoundary2ndLevel* eingeführt (siehe Abbildung 4), die zusätzlich die Topologie der Begrenzungselemente abbilden. Hierzu gibt es zum einen das Attribut *ParentBoundary*, das die Beziehung von übereinander liegenden Raumbegrenzungen, wie sie für Fenster und Türen verwendet werden, beschreibt. Zum anderen wird bei 2nd Level Space Boundaries die Beziehung der gegenüber liegenden Raumbegrenzungen mit dem Attribut *CorrespondingBoundary* beschrieben. Dies vereinfacht die Interpretation erheblich, da nicht mehr geometrisch die Verbindungen der Raumbegrenzungselemente ermittelt werden müssen. Zusätzlich wurde die EXPRESS *Where* Rule (Regel) des Entity *IfcRelSpaceBoundary* erweitert, um für virtuell definierte Begrenzungselemente auch Elemente vom Typ *IfcOpeningElement* zu erlauben.

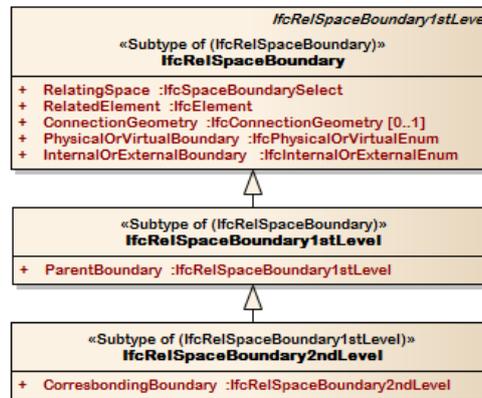


Abb. 4: Vererbungshierarchie des Entity *IfcRelSpaceBoundary* in IFC4

Eine der größten Neuerungen im Zusammenhang mit Raumbegrenzungen ist die Einführung des Entity *IfcExternalSpatialElement*. Damit ist es möglich, die äußere Hülle eines Bauwerkes zu beschreiben, wie sie in der energetischen Simulation benötigt wird. Mit dem Entity *IfcExternalSpatialElement* wird hierzu entweder virtuell die Luft um das Gebäude oder physikalisch das umliegende Gelände bzw. das Nachbargebäude beschrieben. Über die Raumbegrenzungen ist es somit möglich, die Außenhülle eines Bauwerkes abzubilden.

Abbildung 5 zeigt den direkten Vergleich zwischen a) den Raumbegrenzungen aus IFC2x3 und b) den detaillierten Begrenzungsflächen zur Abbildung der Außenhülle eines Bauwerkes. Das Attribut *PredefinedType* des Entity *IfcExternalSpatialElement* spezifiziert zusätzlich noch

die Angrenzbedingung, die beschreibt mit welchem Stoff ein Außenbauteil eines Gebäudes in Berührung steht. Dieses Attribut unterscheidet hierbei zwischen Luft (EXTERNAL), Erdrreich (EXTERNAL_EARTH), Wasser / Grundwasser (EXTERNAL_WATER) und Nachbargebäuden (EXTERNAL_FIRE).

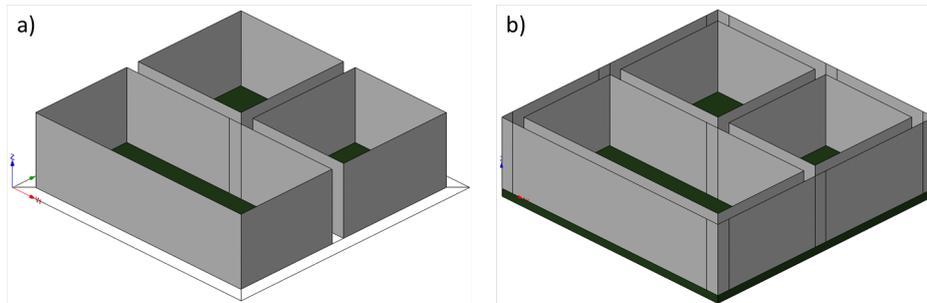


Abb. 5: Gegenüberstellung IFC2x3 und IFC4

Die letzte entscheidende Änderung für den Anwendungsfall Simulation betrifft die Einführung des neuen Zonelements *IfcSpatialZone*. Bislang war es nicht möglich, für das Entity *IfcZone* eine eigene Geometrie zu spezifizieren, da es sich hierbei um ein Entity vom Typ *IfcGroup* handelt und nur zum logischen Gruppieren von Elementen vorgesehen ist. *IfcSpatialZone* hingegen ist von *IfcSpatialElement* abgeleitet, das auch die Basis für Elemente wie *IfcSite*, *IfcBuilding* oder *IfcSpace* bildet und damit auch eine geometrische Repräsentation erlaubt.

Diese Vielzahl an Änderungen bietet neue Möglichkeiten und erlaubt es insbesondere den sogenannten „Downstream Applications“, die nicht unbedingt über einen eigenen geometrischen Kern verfügen, IFC4 konforme Daten leichter zu verarbeiten.

5 Konvertierung der IFC Raumbegrenzungselemente für die energetische Simulation

Im Rahmen einer langjährigen Kooperation mit der Firma Hottgenroth Software wurde für die am Institut für angewandte Informatik des KIT entwickelte Anwendung IFCEXplorer ein Modul zur Verarbeitung von IFC-Gebäudemodellen in deren Softwareprodukten implementiert. Für alle Softwareprodukte wurde bei Hottgenroth Software ein zentrales Datenmodell als konkrete Implementierung von BIM entwickelt, das auch für die Anbindung des IFCEXplorer verwendet wird. Die Datenüberführung von IFC basiert auf dem Konzept der Raumbegrenzungen von IFC2x3 unter Berücksichtigung aller notwendigen von IFC bereitgestellten Relationen.

Das Ausgangsmodell des Transformationsprozesses ist ein in Abbildung 6a dargestelltes Architekturmodell. Alle Bauteile sind in den realen Dimensionen als Volumengeometrie beschrieben. Aus solch einem Architekturmodell sind viele CAAD (Computer-Aided Architectural Design) in der Lage ein Modell der 2nd Level Raumbegrenzungen abzuleiten (siehe Abbildung 6b). Dieses Modell ist die Ausgangsbasis für die Simulation. Die dargestellten Flächen befinden sich an der Innenseite der Räume. Somit sind bei Innenbauteilen doppelte Flächen zu jeder angrenzenden Raumseite vorhanden. Die Darstellung in Abbildung 6c zeigt das endgültige Berechnungsmodell. Dabei sollen die Vorgaben länderspezifischer Berechnungsvorschriften berücksichtigt werden. Aktuell wird diese Anpassung in der Software der Firma Hottgenroth

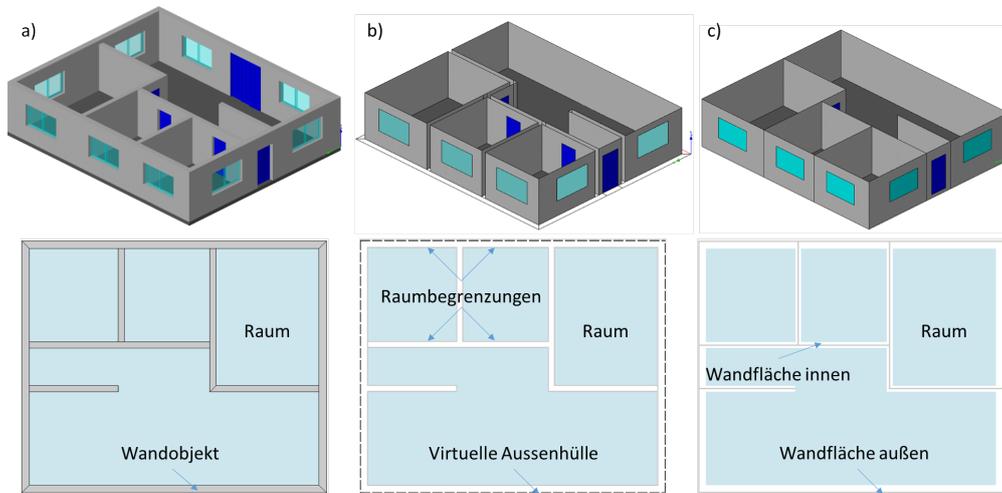


Abb. 6: Modellüberführung vom Architekturmodell zum Berechnungsmodell

Software durchgeführt und anschließend in tabellarischer Form raumweise präsentiert. Eine grafische Darstellung des endgültigen Berechnungsmodells existiert noch nicht.

Das Simulationsmodul von Hottgenroth Software basiert auf den Berechnungsgrundlagen der VDI 6007 "Berechnung des instationären thermischen Verhaltens von Räumen und Gebäuden – Raummodell“, in der die Räume inklusive ihrer Nachbarschaftsbeziehungen einzeln simuliert werden. Dies bezieht sich auch auf Räume mit gleichen thermischen Eigenschaften, die zu thermischen Zonen zusammengefasst sind. Nach diesem Schema wird auch die Datenaufbereitung im IFCEXplorer durchgeführt. Für jeden Raum wird ein topologisches Kantenmodell aus den Raumbegrenzungsflächen abgeleitet. Das bedeutet, die Nachbarschaftsbeziehungen der einzelnen an einen Raum angrenzenden Bauteilflächen werden im positiven geometrischen

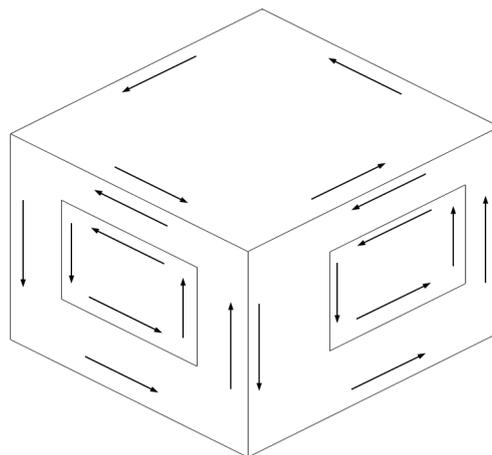


Abb. 7: Topologie eines Raumes

Umlaufsinn topologisch aufgebaut. Bei korrekten und in sich geschlossenen Raumbegrenzungselementen ist jede Kante doppelt, mit gegenläufigem Richtungssinn, enthalten. In Abbildung

7 sind für die sichtbaren Wand- und Dachelemente die Kanten inklusive ihrer Orientierung dargestellt. Eine Besonderheit stellen Fenster- und Türbeziehungen dar. Dort gibt es keine vollständige Kantentopologie, da diese Elemente nicht geometrisch abgezogen werden, sondern nur eine Zugehörigkeitsbeziehung zu einem Bauteil existiert. Fenster- und Türelemente dürfen somit auch nicht für eine topologische Prüfung auf Geschlossenheit eines Raumes berücksichtigt werden.

Das Kantenmodell besitzt neben den geometrischen Informationen wie Kantenlänge und dem Verbindungswinkel auch die Elementinformationen der Nachbarbauteile. Für Softwareprodukte von Hottgenroth Software ist diese Information notwendig, da hiermit die normativen Anpassungen der Bauteil-Geometrien durchgeführt werden können. Darunter versteht man die unterschiedliche Positionierung und Dimensionierung der resultierenden Bauteilflächen. Berücksichtigt werden unter anderem die Normen DIN 12831, DIN 18599 und VDI 6007. In Tabelle 1 sind beispielhaft für diese drei Normen die Lage der Berechnungsflächen aufgezeigt.

Tabelle 1: Lage der Berechnungsflächen nach unterschiedlichen Normen

DIN 12831	Es wird immer die Fläche der Rauminnenseite verwendet.
DIN 18599	Beidseitig beheizte und beidseitig unbeheizte Räume verwenden die Mittel­fläche. Bei Außenwänden und bei Innenwänden zwischen einem beheizten und unbeheizten Raum wird immer die vom beheizten Raum aus betrachtete äußere Wandfläche verwendet.
VDI 6007	Innenwände werden immer mit der Mittel­fläche berechnet und Außenwände mit der äußeren Wandfläche.

Nach dem Aufbau der Kantentopologie erfolgt die Übernahme der Daten in das Datenmodell. Dabei werden zusätzliche semantische Informationen aus dem IFC-Gebäudemodell herangezogen. Hierzu zählen unter anderem die Relationen zum Abbilden der Projekt- und Gebäudestruktur und die Beziehungsinformation zwischen Bauteilen, sowie der darin enthaltenen Öffnungselemente.

Etwas schwieriger wird es bei der Übernahme von Materialinformationen aus IFC und der korrekten Zuordnung zum Materialkatalog des Zielsystems. Dieser Schnitt ist notwendig, da bei den meisten Architekturmodellen nicht davon ausgegangen werden kann, dass auch Materialspezifische Parameter, wie z. B. der U-Wert bereits angegeben sind. Die verwendeten Bezeichner für das Material, aber auch für die Schichten des Materialaufbaus, sind nutzer- und landesspezifisch und nicht standardisiert. Die Übergabe von Materialinformationen lässt sich im Moment nur durch eine manuelle Zuordnung der Material- und Schichtaufbauten im Zielsystem realisieren. Ist das Material nur bei Bauteilen angegeben, so ist dies mit einem erheblichen Aufwand verbunden. Bei der Verwendung von Materialdefinitionen an den Bauteiltypen reduziert sich der Aufwand auf die Anzahl der verwendeten Typen.

Da die aktuelle Version IFC2x3 keine Angrenzbedingungen kennt, müssen diese ebenfalls händisch an den entsprechenden Bauteilen ergänzt werden.

Auf Basis dieser Modellüberführung ist in verschiedenen Softwaretools der Firma Hottgenroth Software eine Berechnung mit Daten aus einem IFC-Gebäudemodell möglich und steht Kunden zur Verfügung. Abbildung 8 zeigt die Ergebnisse eines über IFC importierten Gebäudes. Links dargestellt sieht man die Gebäudestruktur (Geschosse und Räume) des Gebäudes in einer Baumdarstellung, rechts eine Tabelle der einzelnen Bauteile mit den spezifischen Bauteilparametern. Das Gebäude wird mit den Klimadaten für den gewählten Standort für ein Jahr

IFC-Daten für die thermische Gebäudesimulation

in Stundenschritten simuliert und das Ergebnis wird grafisch dargestellt. Es werden unter anderem die Außen- und Innenlufttemperatur, die thermischen Lasten durch die Sonne, sowie innere Lasten durch Personen und Geräte aus frei einstellbaren Nutzungsprofilen, sowie die resultierende Heiz- bzw. Kühllast für jede Stunde des Jahres dargestellt

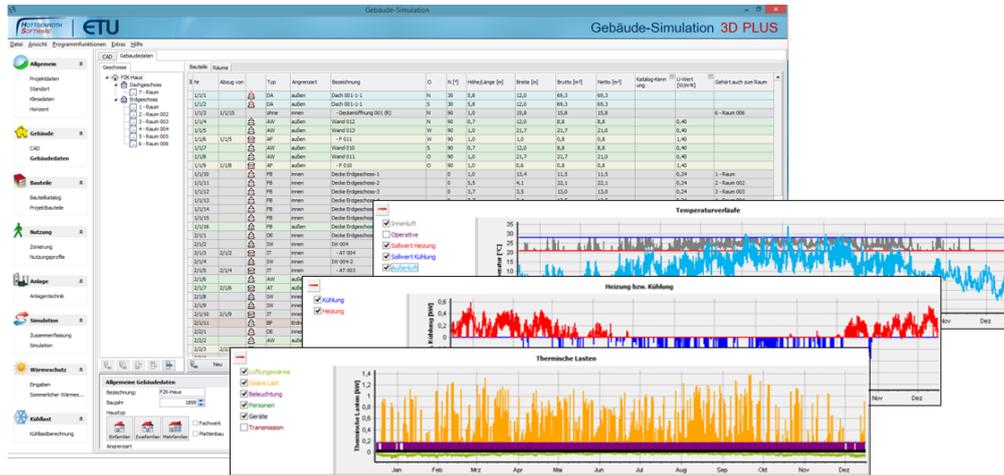


Abb. 8: Simulationsergebnisse in Gebäude-Simulation 3D PLUS

6 Zusammenfassung

buildingSMART stellt mit dem internationalen ISO-Standard IFC ein Datenmodell bereit, das einen herstellerunabhängigen Datenaustausch unter anderem von CAAD Systemen hin zu Simulationswerkzeugen ermöglicht. In diesem Beitrag werden hierzu die Funktionalität von Raumbegrenzungselementen der aktuellen Version IFC2x3, sowie die Erweiterungen von IFC4 erläutert. Am Beispiel von Softwaretools der Firma Hottgenroth Software wird aufgezeigt, wie in der Praxis eine Transformation der IFC-Daten für die Simulation des thermischen Verhaltens realisiert wurde.

Aktuell basiert diese Transformation auf der Version IFC2x3. Der entscheidende Vorteil, in der Verwendung von Modelldaten, wie sie IFC bereitstellt, liegt in der vollständigen Übernahme von Räumen und Bauteilen in die tabellarischen Ansichten der Produkte der Firma Hottgenroth Software. Eine manuelle Überarbeitung dieser Daten ist allerdings weiterhin notwendig, da es zum einen keine standardisierten Bezeichnungen für Material- und Schichtaufbauten gibt und zum anderen verfügt die Version IFC2x3 über keine Möglichkeiten die Außenhülle eines Gebäudes, sowie Angrenzbedingungen zu definieren. Ebenfalls ist es bislang noch nicht möglich Zonen, insbesondere thermische Zonen geometrisch zu beschreiben. Für diese Beschränkungen bietet die Version IFC4 nun Lösungen an.

In der aktuellen Implementierung erfolgt im IFCEXplorer ausschließlich eine Visualisierung der Raumbegrenzungselemente wie sie aus den IFC-Daten eingelesen werden. Die normative Anpassung erfolgt auf Seite der Softwareprodukte der Firma Hottgenroth Software. Dabei steht allerdings keine grafische Kontrollmöglichkeit mehr zur Verfügung.

Zukünftig wird der Konvertierungsprozess für die normativen geometrischen Anpassungen bereits im IFCEXplorer durchgeführt. Durch die Verlagerung dieses Schritts in den IFCEXplorer,

kann das Berechnungsmodell vor der Simulation grafisch und geometrisch validiert werden. Mit den Erweiterungen von IFC4 können die Berechnungsflächen der Außenhülle mit deren Angrenzbedingungen berücksichtigt werden.

Die Implementierung hat gezeigt, dass die Datenübernahme aus einem BIM Modell zuverlässiger und weniger fehleranfällig ist, als über die tabellarische Eingabe. Trotz manueller Aufbereitung der Eingangsdaten wird damit eine erhebliche Zeitersparnis erzielt. Dieses verdeutlicht, dass viele Arbeitsabläufe am Bauprozess durch eine konsequente Nutzung von BIM vereinfacht werden können. Gerade in der Planungsphase mit vielen Entwurfsänderungen lässt sich dadurch Zeit und Kosten sparen.

Literatur

- Bazjanac, V. (2010). „Space boundary requirements for modeling of building geometry for energy and other performance simulation“. In: *CIB W78: 27th International Conference*.
- BLIS-Project (2002). *Building lifecycle interoperable software (BLIS)*. <http://www.blis-project.org/index2.html> (besucht am 08/2016).
- Borrmann, A., König, M., Koch, C. und Beetz, J. (2015). *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*. Springer-Verlag.
- Eastman, C. M. (1999). *Building product models: computer environments, supporting design and construction*. CRC press.
- Häfele, K.-H. und Liebich, T. (2009/2010). „IFC Implementation Agreement, Space Boundary“. In: *buildingSMART*.
- Hietanen, J. (1999). *Space boundaries in IFC R2.0*. http://www.blis-project.org/private/proposals/IFCR2_SpaceBoundaries_991124_jh.pdf.
- Model Support Group of buildingSMART (2007). *IFC Specification, IFC2x Edition 3 Technical Corrigendum 1, buildingSMART*.
- Model Support Group of buildingSMART (2014). *IFC Specification, IFC Version 4 - Addendum 1, buildingSMART*.
- Weise, M., Liebich, T., See, R., Bazjanac, V. und Laine, T. (2009). *IFC Implementation Guide Space Boundaries for Thermal Analysis*.

BIM-Based modeling of structural health monitoring systems using the IFC standard

Maximilian Sternal¹ and Kosmas Dragos²

Chair of Computing in Civil Engineering, Bauhaus University Weimar, Germany

¹*E-Mail: maximilian.sternal@uni-weimar.de,* ²*E-Mail: kosmas.dragos@uni-weimar.de*

Abstract Wireless structural health monitoring (SHM) systems are installed at a growing number of civil engineering structures to detect potential damage or unexpected structural response. In the past years, building information modeling (BIM) has substantially been changing the workflow of planning and operating engineering structures. Typically, building information is stored and exchanged via model files that are based, for example, on the Industry Foundation Classes (IFC) standard. Even though the IFC provide tools to model certain sensor types, modeling a complete SHM system including information about sensors and monitoring strategies, which in the case of wireless SHM systems encompasses, for example, information on sensor nodes and embedded algorithms (“monitoring-related information”), is not possible with current IFC standard. This paper reviews and assesses different semantic approaches towards modeling monitoring-related information for a BIM-based representation of wireless SHM systems. Using a laboratory test structure, a case study is presented to demonstrate the modeling approach on a wireless SHM system. Autoregressive analysis algorithms are illustratively embedded into the wireless sensor nodes for damage detection. The case study is used for assessing potential limitations of current standards. Finally, the outcome of the case study is discussed and approaches towards BIM-based modeling of wireless SHM systems using the IFC standard are proposed.

Keywords: Structural health monitoring (SHM), building information modeling (BIM), wireless sensor networks, monitoring information modeling

Introduction

Structural health monitoring is a key technology for assessing and maintaining the structural safety of civil infrastructure. In recent years, maintaining existing civil infrastructure has received increasing attention. The need for infrastructure maintenance has been emphasized in reports from national institutions, such as the American Society of Civil Engineers (ASCE, 2013) or the German Committee on Transport and Digital Infrastructure (Deutscher Bundestag, 2014). In Germany, financial projections for maintaining existing civil infrastructure reach the amount of € 7.2 billion (Deutscher Bundestag, 2014). Implementing structural health monitoring systems facilitates infrastructure maintenance, thus reducing the probability of damage as well as the need for extensive repair works.

Wireless structural health monitoring systems are composed of wireless sensor nodes with embedded computing capabilities that allow processing and analyzing sensor measurements on board. Wireless SHM systems, as compared to cable-based SHM systems, are easier to install and relatively low-cost. With the growing use of wireless SHM systems, there is an increasing need for modeling wireless SHM systems to facilitate the documentation and the communication of monitoring-related information. Monitoring-related information in wireless SHM systems includes, for example, the description of the algorithms embedded into wireless sensor nodes, information about the monitoring strategies, or the interactions among wireless sensor nodes. Some approaches towards a semantic description of wireless SHM systems have been proposed in the fields of electrical and software engineering as, for example, reported in Compton et al. (2012). However, describing monitoring-related information of wireless SHM systems in civil engineering has not been adequately addressed.

Building information modeling (BIM) is a widely employed technology in the building and construction industry. BIM is gaining increasing importance in civil engineering for planning and design of civil infrastructure. The objective of BIM is to provide an integrated digital methodology that supports planning, maintenance and operation of structures throughout their whole life cycle. Supporting interoperability and information exchange, BIM is recommended in publicly-funded building projects in all European countries (European Parliament, 2014). In Germany, planning and building with BIM becomes mandatory for infrastructure projects in 2020 (BMVI, 2015). In the UK, authorities release the “Digital Built Britain” strategy, in which “BIM Level 3” proposes the integration of data, such as sensor information, into the BIM process (HM Government, 2015).

As a technological basis, the Industry Foundation Classes (IFC) are a widely-used, open standard for digital modeling of building information. Using the current IFC standard, information relevant to SHM, such as sensor measurements, can be mapped. However, with the current IFC standard, it is not possible to map all monitoring-related information required to adequately represent wireless SHM systems. Integrating monitoring-related information into the IFC standard would facilitate documenting wireless SHM systems, while enabling a consistent digital representation of all information relevant to wireless SHM systems throughout the whole life cycle of civil infrastructure, thus substantially enhancing the monitoring quality.

This paper focuses on the review and assessment of existing guidelines related to modeling wireless structural health monitoring systems. Furthermore, existing ontologies to be used for modeling sensor systems are reviewed and compared with the IFC standard to evaluate the possibilities of integrating monitoring-related information into IFC. A prototype wireless SHM system is modeled using two different modeling approaches, the Semantic Sensor Network (SSN) ontology and the IFC standard. The prototype system is implemented and validated in laboratory tests on a frame structure. Specifically, embedded autoregressive time series analysis algorithms are implemented into the wireless sensor nodes for structural assessment. The design and implementation, of the wireless SHM system, based on SSN and IFC, is discussed. Finally, conclusions with respect to extending the current IFC standard are drawn.

Modeling of structural health monitoring systems

To develop a semantic model for wireless SHM systems, existing regulations and standards are considered that provide the minimum requirements for the information the model must include. Apart from SHM regulations, an increasing number of standards for semantic modeling of sensor information have been developed and published in the past years (Daum, 2013). In this

section, existing regulations and standards are reviewed, while approaches towards semantic modeling of monitoring-related information, based on existing regulations and standards, are proposed.

Existing regulations and standards

Most regulations related to monitoring focus on traditional monitoring activities, such as visual inspections, rather than on automated SHM procedures. In Germany, the inspection and test of engineering structures in connection roads must be conducted in accordance with the DIN 1076 standard (DIN, 1999). On the other hand, several research groups have reported approaches to describe the implementation and documentation of monitoring techniques (Rücker et al., 2006). For example, the German Federal Institute of Material Research and Testing (BAM) together with the Structural Assessment Monitoring and Control network (SAMCO), have published the “Guideline for Structural Health Monitoring”, which describes different monitoring technologies along with instructions about their application. The “Guideline for Structural Health Monitoring” includes, for example, selection criteria for sensors, such as sensor resolution, measurement accuracy, and measurement frequency.

In addition to the regulations and guidelines, the semantic modeling of sensor information has been approached in multiple modeling languages and standards. Most standards related to semantic sensor information modeling are assigned to specific fields of engineering, such as the Open Building Information Xchange (OBIX) for building automation, or the Sensor Model Language (SensorML). SensorML is part of the Sensor Web Enablement (SWE) initiative, which provides standardized communication protocols and web services in order to describe sensors and the measurement process. Using SensorML as a basis, the World Wide Web Consortium (W3C) has started the SSN Incubator Group in 2009, which has defined the SSN ontology to describe sensors and related information (Compton et al., 2012). Therefore, the SSN ontology models a sensor, concentrating on what is sensed and how. The main focus is on the sensor measurements, the system as well as the deployment of sensors and the properties that describe the sensor and the observations.

Integrating monitoring-related information into building information models

Building information modeling integrates multi-level data covering all phases of the life cycle of structures, including design, construction, maintenance, and, possibly, reuse or demolition, thus improving cost efficiency and enabling a better planning (Eastman et al., 2011). With respect to modeling SHM systems, a few research endeavors combining structural health monitoring and building information models can be found in the literature (e.g. integrating real-time sensor measurements into BIM). For example, Chen et al. (2014) and Rio et al. (2013) demonstrated how integrated sensor measurements can be visualized, evaluated and further used in BIM-based applications. However, the integration of monitoring-related information, as defined previously, into BIM has not received adequate attention. A main idea behind BIM is to have one centralized working model for the whole life cycle of a building, with the building information being accessible and editable any time. In this context, models created in the design phase are unable to “react” to changes during the life cycle of the building.

As mentioned earlier, the IFC are an open standard for building information modeling (ISO, 2013). Developed and maintained by BuildingSMART International, the IFC standard provides

a common data format supporting interoperability between different software packages and stakeholders. It is defined in the ISO 16739 standard (ISO, 2013). An IFC object model is composed of entities with specific attributes and relationships among the entities. The modeling language EXPRESS is used to formalize the data, whereas the “Standard for the Exchange of Product Model Data and the Extended Markup Language” (STEP-XML) provides a basis for exchanging the IFC models.

The IFC standard mainly focuses on buildings; in the past years, several research projects have made efforts to extend the IFC (version 4) format in order to integrate infrastructure elements such as roadways (BuildingSMART, 2015), tunnels (Borrmann et al., 2015), and bridges (Ji et al., 2012). To evaluate, what entities and relationships are required for an adequate representation of a wireless SHM system, the relevant modeling possibilities and options in the current IFC version 4 need to be investigated (Smarsly and Tauscher, 2015). Therefore, a case study investigating the mapping a wireless SHM system with embedded auto-regressive models with exogenous inputs using the IFC standard is shown and evaluated in the following section.

Modeling a wireless SHM system using the SSN ontology and the IFC object models

The goal of this case study is to assess the modeling possibility of a wireless SHM system with embedded algorithms and distributed processing. A wireless SHM system, equipped with embedded autoregressive analysis algorithms for damage detection, is modeled, implemented, and validated. The SSN ontology and the IFC object model are chosen due to the modeling opportunities and functionality of these semantic modeling languages. While the SSN ontology, which builds on SensorML, includes multiple entities to describe the measurement process, the IFC object model provides a structure to add sensor information into building information models. As opposed to the modeling process, for the sake of clarity the wireless SHM system is first presented, followed by the illustration of the underlying models, i.e. in the following subsections, the laboratory test structure is described and the embedded algorithm is introduced. Then, the monitoring-related information is identified and modeled using the SSN ontology and the IFC object model. Finally, the capabilities of the SSN and the IFC standard with respect to modeling monitoring-related information are discussed.

Laboratory test setup and implementation

The sensor nodes used in this case study are of type “Sun SPOT” (Oracle, 2009). The sensor nodes have a 32 bit 400 Hz ARM main processor with 1 MB random access memory and 8 MB flash memory. For wireless communication, an IEEE 802.15.4-compliant TI CC2420 radio transceiver is included in the sensor nodes. The sensor board comprises an internal 3D digital-output accelerometer, type MMA7455L, which is used for collecting measurement data (accelerations) in this study.

As shown in Figure 1, the sensor nodes are deployed on a four-story frame structure. The floors, made of 0.8 mm thick steel plates, have dimensions of 25 cm (width) by 50 cm (length). The height of each story is 23 cm, while the columns are fixed on a solid block at the base of the structure. The response of the structure in terms of accelerations is measured with the wireless sensor nodes placed at the middle of each story, as shown in Figure 1.



Fig. 1: Laboratory test structure instrumented with wireless SHM system.

Autoregressive models (AR) and autoregressive models with exogenous inputs (ARX) are embedded into the wireless sensor nodes for assessing the condition of the laboratory test structure. Autoregressive analysis algorithms for damage detection have proven to be effective in structural health monitoring (Sohn et al., 2001). Autoregressive models for structural assessment, in general, involve comparison of two or more states, an “initial” (or “undamaged”) state and a “current” state of the structure. At the initial state, a prediction model for the structural response is developed to be used as a reference for comparison with the current state in order to detect structural changes. Deviations between the response obtained by the prediction model and the response obtained at the current state indicate an abnormal condition and potential damage. The prediction model employed in this study is shown in Eq. 1.

$$x(t) = \sum_{i=1}^p \phi_{xi}(t - i) + e_x(t) \quad (1)$$

where $x(t)$ is the measured response time series and $e_x(t)$ the residual error (a white noise signal). The autoregressive analysis algorithm is specified by the order p and are the AR coefficients ϕ_{xi} , determined using the Yule-Walker equations (Lynch et al., 2003).

The procedure of the structural condition assessment is described by the steps shown in Figure 2. First, one time series of accelerations obtained in the initial state is sent to a server, where a reference database is created, i.e. a family of AR coefficients, by splitting the time series in segments of the same size and progressively shifting each segment until the complete time series is covered. The error term $e_x(t)$ of Eq. 1 is affected by the excitation force, and, as a result, reconstructing $x(t)$ under all excitation conditions is not possible. Hence, the relationship between $x(t)$ and $e_x(t)$ is modeled, essentially considering $e_x(t)$ as exogenous input (ARX model), as shown in Eq. 2.

$$x(t) = \sum_{i=1}^q \alpha_i x(t-i) + \sum_{j=1}^r \beta_j e_x(t-j) + \epsilon_x(t) \quad (2)$$

Then, a time series of newly collected accelerations $y(t)$ is processed directly on the sensor node and the corresponding AR coefficients are obtained on board. Subsequently, the AR coefficients of time series $y(t)$ are sent to the server to browse the database and retrieve the closest matching AR coefficients from the segments of the initial time series $x(t)$. Upon obtaining the closest match of AR coefficients, the corresponding ARX coefficients are sent back to the sensor nodes. A damage sensitive index h is calculated as the ratio of the standard deviations of the two error terms $\sigma(\epsilon_x)$ and $\sigma(\epsilon_y)$. Deviations of h from unity are indicative of structural changes and potential damage.

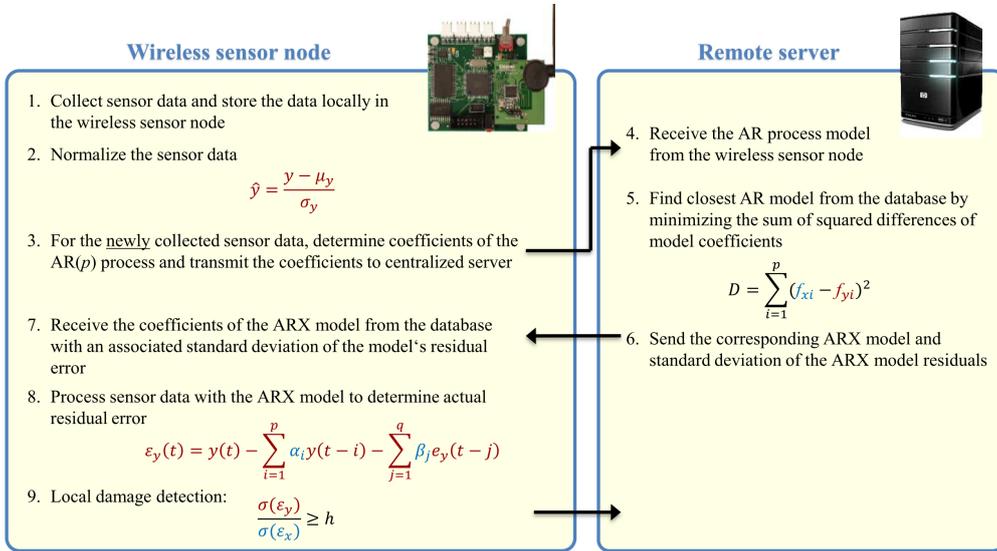


Fig. 2: Structural assessment with embedded AR-ARX algorithm.

Modeling the wireless SHM system using the SSN ontology and the IFC standard

To categorize the monitoring-related information that is needed to model the wireless SHM system, the information is divided into two main categories, global and local monitoring-related information (Smarsly and Tauscher, 2016). Global information includes, for example, the configuration and topology of the wireless sensor network, the interaction protocols used, and the overall monitoring strategy with global diagnosis methods. Also, information required for data storage and for the generation of reports is part of the global monitoring-related information. Local monitoring-related information describes, for example, the specifications regarding single sensor nodes, such as hardware and software parameters as well as the embedded algorithms.

The wireless SHM system model based on the SSN ontology is shown in Figure 3. When using the SSN ontology to model the wireless SHM system, it turns out that the specifications

of the sensors themselves can be modeled very well. To be more specific, by using the entity *ssn:Sensor*, local parameters can be defined as subclasses of *ssn:MeasurementProperty* (e.g. *ssn:Sensitivity* or *ssn:Resolution*). These properties are collected in the entity *ssn:MeasurementCapability* (Figure 3). The wireless sensor node, in this case a node of type Oracle SunSPOT, is described by the entity *ssn:SensingDevice*. Individual sensor measurements are stored in the entity *ssn:ObservationValue*.

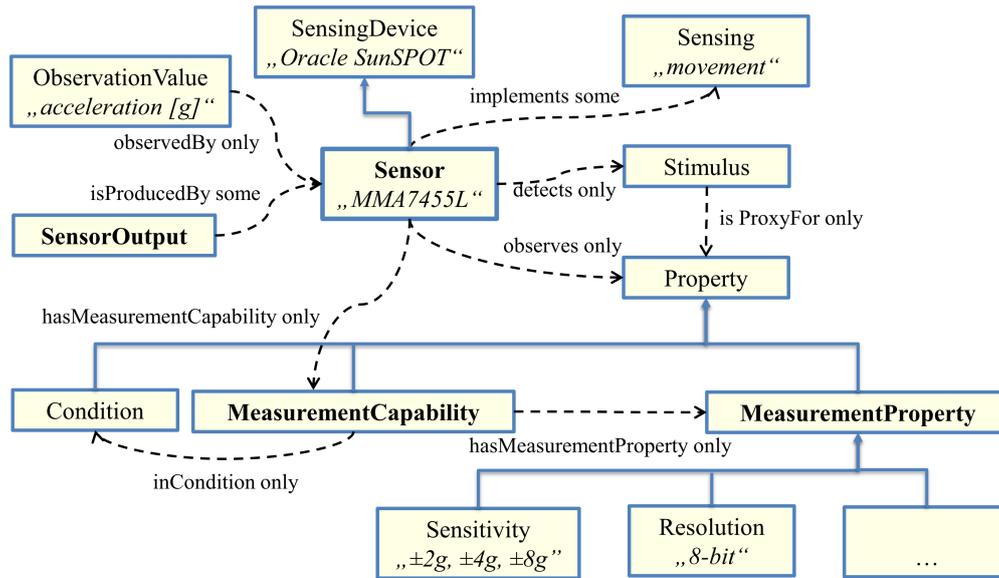


Fig. 3: Entities describing sensors using the SSN ontology

While the means provided by the SSN ontology are sufficient for modeling single sensors being part of the local monitoring-related information, modeling the global monitoring-related information is hardly possible. The entity *ssn:Deployment* is capable to describe the monitored structure as well as locations and types of sensors, but relations within the wireless sensor network or between sensors and sensor nodes cannot sufficiently be modeled using the given entities. Moreover, the embedded algorithms are not representable in the SSN ontology, representing one of the major drawbacks of the SSN ontology as a semantic model for wireless SHM systems.

The SSN ontology is capable of mapping the sensing procedure as well as sensor properties and measurement values. To integrate such semantic mapping into BIM, it would be helpful to identify such entities in the Industry Foundation Classes. The IFC object model provides means to model sensors; according to BuildingSMART, the entity *IfcSensor* must be used to map the information regarding sensors and sensor measurements. An *IfcSensor* object can be specified by another entity, the *IfcSensorType*, and its attribute *ObjectType*. A number of sensor types are predefined in IFC, such as *FIRESENSOR* or *WINDSENSOR*. However, many sensor types, such as accelerometer that is used in this case study to create a time series for the autoregressive analysis, are not available in IFC and have to be *USERDEFINED*. Other information can be mapped in the IFC object model by the use of *IfcPropertySet* (Figure 4). A lot of additional information, such as the accuracy or energy consumption of a sensor can be mapped by using the *IfcPropertySet*, but only predefined sensor types have applicable property sets to describe its characteristics. For the undefined accelerometer type, for example, the option using a user-defined property set is not straightforward. Furthermore, the relationships between the wireless

sensor nodes and the server as well as the algorithms implemented are hardly describable using property sets.

```
[...]
#1892= IFCSENSORTYPE('27H09Zl057fhS6mRbCdWcP',#41,'Occupancy', $,$,$,(#1891),$,$,.USERDEFINED.)
[...]
#1900= IFCSENSOR('2rMUVkbjj2DeNtqwD6dmKd',#41,'MMAA7455L-Accelerometer',$,'2.4V-3.6V',#1892,$)
[...]
#1906= IFCPROPERTYSET('1B$IuvAwP9PeFoRkMzaW0S',#41,'Pset_SensorTypeInformation',$,(#1900));
#1908= IFCREDEFINESBYPROPERTIES('3rFZxA5AD3gxbayNTg4zky',#41,$,$,(#1892),#1900);
#1912= IFCREDEFINESBYPROPERTIES('2XU73Ko$H6BguCUw7D2czh',#41,$,$,(#1892),#1906);
#1916= IFCPROPERTYSINGLEVALUE('Sensitivity',$,IFCTEXT('±2g, ±4g, ±8g'),$);
#1917= IFCPROPERTYSINGLEVALUE('SamplingRate',$,IFCTEXT('125Hz / 250Hz'),$);
#1919= IFCPROPERTYSINGLEVALUE('Sensor node',$,IFCTEXT('Oracle SunSPOT'),$);
#1920= IFCPROPERTYSINGLEVALUE('Layer',$,IFCLABEL('Layer: Layer 0'),$);
[...]
#1923= IFCPROPERTYSINGLEVALUE('Embedded algorithm',$,IFCTEXT('Autoregressive Model'),$);
#1925= IFCPROPERTYSINGLEVALUE('Systemclassification',$,IFCTEXT('Structural Health Monitoring'),$);
#1926= IFCPROPERTYSINGLEVALUE('Systemname',$,IFCTEXT(''),$);
[...]
#1934= IFCPROPERTYSINGLEVALUE('Family und Typ',$,IFCLABEL('Accelerometer'),$);
#1953= IFCPROPERTYSINGLEVALUE('Application / Location',$,IFCTEXT('Sensor node'),$);
#1955= IFCPROPERTYSINGLEVALUE('Measurement Output Signal Type',$,IFCTEXT('Changing Resistance'),$);
#1957= IFCPROPERTYSINGLEVALUE('Measurement Type',$,IFCTEXT('Acceleration'),$);
#1963= IFCPROPERTYSINGLEVALUE('Documentation Link',$,IFCTEXT('http://www.nxp.com/products/sensors/a
[...]

```

Fig. 4: Exemplary sensor description in IFC using PropertySets

Discussion of the results

Illustrating the feasibility of modeling monitoring-related information, Table 1 & 2 summarizes the comparison between the entities available in the SSN ontology and in the IFC object model. In general, both semantic languages provide possibilities to map single sensors and the measurement values. For undefined sensor types, the IFC object model hardly has available properties. Furthermore, both modeling languages are not designed to map or describe monitoring strategies and embedded algorithms, such as the autoregressive analysis. Therefore, new entities to map monitoring-related information of wireless SHM systems are needed.

Table 1: Comparison of entities for modeling monitoring-related information (excerpt)

Monitoring-related information	SSN ontology	IFC object model
<i>Global monitoring-related information</i>		
Hardware specifications:	<i>Nodes and Server</i> <i>ssn:System</i>	[not available]
Topology of the sensor network:	Positions (x, y, z) and story <i>ssn:Deployment (ssn:Platform)</i>	<i>IfcRelConnectsElement</i> [element to which sensor is attached]
Data storage:	<i>Type of database</i> [not available]	<i>IfcPropertyListValue</i>
Monitoring strategies	<i>Sequence of monitoring (send-recieve)</i> [not available]	[not available]

Table 2: Comparison of entities for modeling monitoring-related information (excerpt)

Monitoring-related information		SSN ontology	IFC object model
<i>Local monitoring-related information (about the sensor/wireless sensor node)</i>			
Type of sensor:	<i>MMA7455L Accelerometer</i>	<i>ssn:Sensor</i>	<i>IfcSensorTypeEnum</i> [no defined entity] <i>IfcRelDefinesByType</i>
Sensitivity:	<i>±2g, ±4g, ±8g for 8-bit Mode</i>	<i>ssn:Sensitivity (ssn:Measurement-Property)</i>	<i>IfcPropertySingleValue</i> [to be specified]
Sampling rate:	<i>125Hz / 250 Hz</i>	<i>ssn:Frequency (ssn:Measurement-Property)</i>	<i>IfcPropertySingleValue</i> [to be specified]
Power range:	<i>2.4V – 3.6V</i>	<i>ssn:OperatingPower-Range</i>	<i>IfcPropertySingleValue</i> [to be specified]
Measured values:	<i>Acceleration [g]</i>	<i>ssn:ObservationValue (ssn:SensorOutput)</i>	<i>IfcPropertyListValue</i>
Processed parameters:	<i>List of numbers [double]</i>	[not available]	[not available]
Algorithm:	<i>AR-ARX procedure</i>	[not available]	[not available]

Summary and conclusions

In this paper, existing semantic modeling languages and standards to represent monitoring-related information of wireless SHM systems have been discussed by the example of the SSN ontology and the IFC object model. It has been demonstrated that integrating monitoring-related information into building information models helps categorizing, documenting and updating monitoring-related information throughout the whole life cycle of the monitored structure. However, this paper has shown that the current IFC standard (IFC 4) does not provide sufficient entities to holistically model (and digitally represent) an overall wireless SHM system. Specifically, information related to the wirelessly communicating sensor nodes with embedded algorithms cannot be adequately modeled. Semantic modeling languages, such as the SSN ontology, may provide a sound basis, because they provide entities whose nature is closer to monitoring. However, semantic modeling languages usually focus on sensor and measurement specifications rather than on overall wireless SHM systems. Therefore, BIM standards need to be extended by a semantic model capable of representing monitoring-related information.

References

- American Society of Civil Engineers (ASCE) (2013). *2013 Report Card for America's Infrastructure*. Report. <http://www.infrastructurereportcard.org> (visited on 06/10/2016).
- Borrmann, A., Kolbe, T. H., Donaubaue, A., Steuer, H., Jubierre, J. R., and Flurl, M. (2015). „Multi-scale geometric-semantic modeling of shield tunnels for GIS and BIM applications“. In: *Computer-Aided Civil and Infrastructure Engineering* 30.4, pp. 263–281.
- BuildingSMART (2015). *BuildingSMART International Requests Comment on new Infrastructure Alignment bSI Candidate Standard*. Press Release. <http://www.buildingsmart.org/wp-content/uploads/2015/01/IFC-Alignment-Press-Release-final.pdf> (visited on 05/23/2016).
- Chen, J., Bulbul, T., Taylor, J., and Olgun, G. (2014). „A Case Study of Embedding Real-time Infrastructure Sensor Data to BIM“. In: *Construction Research Congress*, pp. 269–278.
- Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al. (2012). „The SSN ontology of the W3C semantic sensor network incubator group“. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 17, pp. 25–32.
- Daum, W. (2013). „Guidelines for structural health monitoring“. In: *Handbook of Technical Diagnostics*. Springer, pp. 539–541.
- Deutscher Bundestag (2014). *Verkehrsinfrastruktur chronisch unterfinanziert*. Press Release. Berlin, Germany. https://www.bundestag.de/dokumente/textarchiv/2014/49449197_kw07_pa_verkehr/215644 (visited on 05/22/2016).
- DIN (1999). *DIN 1076:1999-11 – Ingenieurbauwerke im Zuge von Straßen und Wegen – Überwachung und Prüfung*. Beuth Verlag GmbH. Berlin, Germany: Normenausschuss Bauwesen (NABau) within the Deutsches Institut für Normung e.V. (DIN).
- Eastman, C., Eastman, C. M., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. Hoboken, New Jersey: John Wiley & Sons.
- European Parliament (2014). *Directive 2014/24/EU of the European Parliament and of the Council*. Brussels, Belgium.
- HM Government (2015). *Digital Built Britain: Level 3 Building Information Modelling - Strategic Plan*.
- ISO (2013). *ISO 16739:2013 – Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*. Geneva, Switzerland: International Organization for Standardization (ISO).
- Ji, Y., Borrmann, A., Beetz, J., and Obergrießer, M. (2012). „Exchange of parametric bridge models using a neutral data format“. In: *Journal of Computing in Civil Engineering* 27.6, pp. 593–606.
- Lynch, J. P., Sundararajan, A., Law, K., Kiremidjian, A., and Carryer, E. (2003). „Power-efficient data management for a wireless structural monitoring system“. In: *Proceedings of the 4th International Workshop on Structural Health Monitoring*. Vol. 1. Citeseer, pp. 15–17.
- Rio, J., Ferrera, B., and Pocas-Martins, J. (2013). „Expansion of IFC model with structural sensors“. In: *Informes de la Construcción* 65.530, pp. 219–228.
- Rücker, W., Hille, F., and Rohrman, R. (2006). „Guideline for structural health monitoring“. In: *Report "F08b"*. German Federal Institute of Materials Research and Testing (BAM), Berlin, Germany.
- Smarsly, K. and Tauscher, E. (2015). „IFC-based monitoring information modeling for data management in structural health monitoring“. In: *Proceedings of the 20th Int. Conf. on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering (IKM)*.

- Smarsly, K. and Tauscher, E. (2016). „Monitoring information modeling for semantic mapping of structural health monitoring systems“. In: *Proceedings of the 16th International Conference on Computing in Civil and Building Engineering (ICCCBE)*. Osaka, Japan.
- Sohn, H., Farrar, C., Hunter, N., and Worden, K. (2001). *Applying the LANL statistical pattern recognition paradigm for structural health monitoring to data from a surface-effect fast patrol boat*. Tech. rep. Los Alamos National Lab., NM (US).
- Verkehr und digitale Infrastruktur (BMVI), B. für (2015). *Stufenplan Digitales Planen und Bauen: Einführung moderner, IT-gestützter Prozesse und Technologien bei Planung, Bau und Betrieb von Bauwerken*. Berlin, Germany.

Untersuchung von herstellerspezifischen parametrischen Bauteiltypen in einem digitalen Gebäudemodell auf Vollständigkeit der Informationen und Einhaltung bautechnischer Randbedingungen

Eric Robaszewski¹ und Jakob Kirchner²

Fachgebiet Bauinformatik, TU Berlin, Gustav-Meyer-Allee 25 13355 Berlin, Deutschland
¹E-Mail: eric.robaszewski@campus.tu-berlin.de, ²E-Mail: jakob.kirchner@tu-berlin.de

Abstract In diesem Beitrag wird ein Ansatz beschrieben, wie ein geeignetes Untersuchungsschema erstellt werden kann, um herstellerspezifische parametrische Bauteiltypen hinsichtlich ihrer Anwendbarkeit in digitalen Gebäudemodellen bewerten zu können. Hierzu werden Kategorien entwickelt und Kriterien exemplarisch aufgestellt. Anschließend werden anhand des aufgestellten Schemas ausgewählte Fenstertypen von Herstellern für das Softwaresystem Autodesk Revit untersucht. Die Ergebnisse der Untersuchung werden vorgestellt.

Keywords: Bauteiltyp, BIM-Content, Parameter, Revit

Einleitung

Ein wesentlicher Bestandteil der traditionellen Arbeitsweise im Bauwesen ist die technische Zeichnung. Hersteller bieten bereits seit vielen Jahren Detailzeichnungen von ihren Bauprodukten in Formaten wie beispielsweise *DWG* an. Bei der Umsetzung eines Bauprojektes mit der BIM-Methodik steht, im Gegensatz zur technischen Zeichnung, das Bauwerksmodell im Mittelpunkt. Erforderliche Ansichten und Schnitte lassen sich aus diesem generieren. Bei der Entwicklung solcher digitaler Gebäudemodelle können herstellerspezifische Bauteile integriert werden.

Viele Hersteller haben bereits ihre Bauteile in Form von parametrischen Bauteiltypen modelliert. Die digitalen Bauprodukte werden zunehmend auf Online-Portalen den Planern zur Verfügung gestellt. Die Planer können diese in ihren Gebäudemodellen instanzieren und so effizienter arbeiten. In (Robaszewski, 2016, S. 30 f.) wurden Portale mit kostenfreien herstellerspezifischen Inhalten genauer untersucht. Viele der Online-Portale sind innerhalb der letzten 4 Jahre entstanden. Bauteiltypen für das Softwaresystem Autodesk Revit sind am häufigsten in Online-Bibliotheken vorhanden, gefolgt von Graphisoft ArchiCAD und Bentley Microstation.

Im vorliegenden Beitrag wird ein Ansatz zum Aufstellen und Klassifizieren von Untersuchungskriterien vorgestellt. Die Kriterien können für die Bewertung der Anwendbarkeit von herstellerspezifischen Bauteiltypen in digitalen Gebäudemodellen herangezogen werden. Es

Untersuchung von herstellerspezifischen parametrischen Bauteiltypen in einem digitalen Gebäudemodell auf Vollständigkeit der Informationen und Einhaltung bautechnischer Randbedingungen

wird ein exemplarisches Untersuchungsschema für Fenstertypen von Herstellern für das Softwaresystem Autodesk Revit, sogenannte Familien, erstellt. Hierbei werden Kategorien entwickelt und Kriterien aufgestellt. Die Kriterien werden anschließend der jeweiligen Kategorie zugeordnet. Das so entwickelte Schema wird zur Untersuchung von ausgewählten freiverfügbaren Fenstertypen von Herstellern herangezogen. Es soll unter anderem geprüft werden, welche Informationen den digitalen Einbauteilen von den Herstellern mitgegeben werden. Zudem soll geklärt werden, inwieweit das parametrische Verhalten der Bauteilobjekte mit der bautechnischen Umsetzbarkeit in der Realität übereinstimmt. Die Untersuchungen dieser Arbeit sind als exemplarisch zu verstehen. Der Fokus dieses Beitrags liegt darin, Bereiche aufzuzeigen, die für eine möglichst umfassende Beurteilung der Eignung von verfügbaren Bauteiltypen relevant sind.

Parametrische Bauteiltypen

Ein *parametrischer Bauteiltyp* ist eine Klasse, mit der reale Bauteile im Computer dargestellt werden können. Ein *parametrisches Bauteilobjekt* ist eine Instanz dieses Typs. Das Objekt wird über seine Parameter und sein Verhalten beschrieben. Das Verhalten entspricht der Parameteränderung/en des Objektes, ausgelöst durch innere und/oder äußere Einflüsse. Ein Beispiel für einen möglichen inneren Einfluss ist die Modifikation des objekteneigenen Parameters Breite eines Balkenobjektes. Ein äußerer Einfluss eines Wandobjektes kann z.B. das Ausschneiden einer passenden Fensteröffnung beim Einbau eines Fensterobjektes sein.

Parameter eines Bauteiltyps können sowohl *geometrische* als auch *nichtgeometrische* Informationen repräsentieren. Geometrische Informationen sind Angaben zu den Abmessungen des Bauteils. Diese können geometriesteuernd oder nicht geometriesteuernd sein. Nichtgeometrische Informationen können beispielsweise Materialkennwerte oder Angaben zur Herstellung sein. Diese Informationen sind meist nicht ohne weiteres an dem realen Bauteil ablesbar.

Vorhandene Richtlinien und Anforderungen an Bauteiltypen

Es existieren bereits einige Richtlinien für die Entwicklung von Bauteilobjekten. Viele der vorhandenen Anforderungen sind softwarespezifisch, wie beispielsweise der *Revit Content Style Guide* (Autodesk, Inc., 2009b) und der *Metadata Style Guide* (Autodesk, Inc., 2009a) von Autodesk oder auch das *ArchiCAD GDL Referenzhandbuch* (Graphisoft, 2015) und das *GDL Cookbook* (Nicholson-Cole, 2004) von Graphisoft. Sie zeigen eine Anleitung, sowie Empfehlungen für das Entwickeln von Bauteiltypen auf.

Im *Revit Content Style Guide* wird z.B. empfohlen welche Komponenten eines Bauteilobjektes, abhängig von deren Abmessungen, in welchen der Revit internen Detaillierungsgrade *Coarse*, *Medium* und *Fine* sichtbar sein sollte, siehe Abbildung 1.

If the geometry is...	Set the detail level to...
Smaller than 1'	Fine
1' – 3'	Medium
Larger than 3'	Coarse

Abb. 1: Modellierung der Geometrie in untersch. Detaillierungsgraden (Autodesk, Inc., 2009b, S. 19)

Weiterhin wird in dem Guide auch eine maximale Verschachtelungstiefe der Familien von bis zu 2 Levels aus Performancegründen vorgeschlagen. In der *Bimstore Bible* (BIM-Store, 2015) für Revit von dem Online-Portal *Bimstore UK* wird darüber hinaus die maximale Anzahl von bis zu 6 verschachtelten Familien vorgeschrieben. Ebenfalls sind in diesen Anleitungen Konventionen für die Bezeichnung von Familien, Typen und Parameter zu finden, wie in Abbildung 2 gezeigt.

<FunctionType> _ <Subtype> _ <Manufacturer> _ <Descriptor1> _ <2D if necessary>

Abb. 2: Vorgeschriebenes Format für Familiennamen (BIM-Store, 2015, S. 13)

Auf der anderen Seite existieren auch softwareunabhängige Anforderungen, wie der *NBS BIM Object Standard* (RIBA Enterprises Limited, 2014). Ein Bauteilobjekt, das diesen Anforderungen entspricht muss beispielsweise über alle für den *IFC Export* notwendigen Parameter, sowie über die *COBie*- und die spezifischen *NBS* Parameter verfügen.

Weiterhin gibt es auch Ansätze wie in dem *BIM Handbook* (Eastman et al., 2011, S. 41 f., S. 247 ff.), die grundlegende Kriterien aufzeigen, die bei der Entwicklung von parametrischen Bauteiltypen berücksichtigt werden sollten. Hierzu zählen unter anderem die dreidimensionale Geometrie, die Repräsentation des Materials, Anschlüsse an das Haustechniknetz und zusätzliche Informationen über das Bauteil wie z.B. Leistungsbeschreibung, Lebensdauer, Wartungszyklus, Lichtdurchlässigkeit und Links zum Händler, sowie zum Hersteller.

Die vorhandene Richtlinien beziehen sich meist nur auf bestimmte Bereiche, wie beispielsweise die Parametermodellierung. Zwar werden auch bestimmte Parameter gefordert, damit die Interoperabilität sichergestellt werden kann, jedoch wird damit nicht gewährleistet, dass alle für eine spezielle Anwendung notwendigen Informationen auch tatsächlich vorhanden sind. Das bedeutet, dass beispielsweise der Parameter Lichtdurchlässigkeit zwar existieren kann, aber dieser auch mit keinem Wert belegt sein kann. Zudem sind für die Anwendbarkeit von herstellerepezifischen Bauteilobjekten weitere Bereiche relevant, wie z.B. die Korrektheit von Information, das parametrische Verhalten oder auch die Visualisierung. In diesem Beitrag wird versucht diese zu beleuchten und dessen Bedeutung anzuzeigen.

Kategorien

Die Bewertung eines Bauteiltyps wurde in verschiedene Bereiche aufgeteilt. Die aufgestellten Kriterien werden dementsprechend klassifiziert. Folgende Kategorien wurden entwickelt:

Statistische Auswertung (K1)

Bei dieser Auswertung soll ein Überblick anhand von konkreten Zahlenwerten, wie z.B. die Anzahl der Parameter, die Verschachtelungstiefe und die Anzahl der verschachtelten Bauteiltypen, gegeben werden.

Interaktion des Bauteilobjektes (K2)

Mit Kriterien dieser Kategorie soll festgestellt werden, inwieweit grundlegende Funktionalitäten des Bauteilobjektes anwendbar sind. Funktioniert das im jeweiligen Softwaresystem unterstützte parametrische Verhalten des Objektes?

Untersuchung von herstellerspezifischen parametrischen Bauteiltypen in einem digitalen Gebäudemodell auf Vollständigkeit der Informationen und Einhaltung bautechnischer Randbedingungen

Fachspezifisch (K3)

In diesem Bereich soll bewertet werden, ob die bautechnische Umsetzbarkeit durch das untersuchte Bauteilobjekt korrekt repräsentiert wird. Hierbei sind je nach Art des Bauteils fachspezifische Anforderungen und Vorschriften zu berücksichtigen. Beim Aufstellen dieser Kriterien ist es empfehlenswert, Experten aus dem entsprechenden Fachbereich mit heranzuziehen.

Support (K4)

Die Kriterien dienen der Untersuchung, ob für das digitale Bauteil zusätzliche Informationen in Form von Dokumenten, wie z.B. eine Bedienungsanleitung oder eine Erklärung der Parameter, vorhanden sind.

Vollständigkeit und Korrektheit der Informationen (K5)

Je nach Anwendung sind unterschiedliche Informationen notwendig. In diesem Bereich soll geprüft werden, ob das Bauteilobjekt die nötigen Informationen liefert, damit es für die jeweilige Anwendung wie z.B. eine bauphysikalische Simulation geeignet ist. Ein weitere mögliche Kategorie ist die Bewertung der korrekten Klassifikation wie z.B. nach IFC oder nach Omniclass.

Parametermodellierung (K6)

Parameter können unterschiedlich modelliert werden. Hierbei sind u.a. die Bezeichnung und der Typ des Parameters von Bedeutung, aber auch inwieweit Abhängigkeiten zwischen den Parametern bestehen.

Visualisierung (K7)

Eine geeignete Visualisierung von Bauteilobjekten ist wichtig für die Anwendung in digitalen Gebäudemodellen. Die Visualisierung ist ein umfangreicher Bereich und kann in weitere Unterkategorien wie z.B. die Visualisierung der Bewegungsbereiche und realitätstreue Visualisierung unterteilt werden. In diesem Beitrag wird auf diese Unterteilungen verzichtet.

Detaillierungsgrad (K8)

In diesem Bereich soll bewertet werden, in welchem Detaillierungsgrad das entsprechende Bauteil vorliegt und ob verschiedene Detaillierungsgrade von dem Bauteiltyp unterstützt werden. Hierbei sollen u.a. das *Level of Development (LOD)*, siehe (BIMForum, 2015), oder auch softwarespezifische Konzepte, wie z.B. die Detaillierungsgrade *Coarse*, *Medium* und *Fine* untersucht werden.

Die Auswahl dieser Kategorien erfolgte mit dem Ziel, zu einer möglichst umfassenden Beurteilung relevanter Aspekte kommen zu können.

Aufstellen und Klassifikation der Kriterien

Im Folgenden soll gezeigt werden, wie Kriterien aus bestimmten Kategorien aussehen können. Hierzu wurden Kriterien aufgestellt, die zur Untersuchung von Fensterobjekten herangezogen werden können. Diese werden der jeweiligen Kategorie zugeordnet.

Der Fokus dieses Beitrags wurde auf die Kategorien *K1 - K5* und *K7* gelegt. Aus diesem Grund wurden für die Kategorien *K6* und *K8* keine Kriterien entwickelt. Das so aufgestellte Schema ist als exemplarisch zu verstehen. Einige Kriterien wie z.B. *Krit.2.1* und *Krit.7.1* können speziell zur Untersuchung von Fensterobjekten herangezogen werden. Andere wiederum sind allgemeingültiger Natur und können zur Bewertung weiterer Bauteiltypen herangezogen werden.

Tabelle 1: Exemplarische Kriterien für die Untersuchung von Fensterfamilien

Kat.	Abk.	Kriterium
K1	Krit. 1.1	Verschachtelungstiefe
K1	Krit. 1.2	Anzahl verschachtelter Familien
K2	Krit. 2.1	Einbaubarkeit in Wand, automatisches Erzeugen einer Fensteröffnung
K3	Krit. 3.1	Einbaubarkeit außerhalb der Wandgeometrie nicht möglich
K3	Krit. 3.2	Einbaubarkeit in eine Wandkreuzung nicht möglich
K4	Krit. 4.1	Existenz einer Bedienungsanleitung für das digitale Bauteil
K4	Krit. 4.2	Existenz einer Erklärung der Parameter des digitalen Bauteils
K4	Krit. 4.3	Auffindbarkeit des realen Bauteils beim Hersteller
K5	Krit. 5.1	Existenz eines numerischen Wertes zur Lichtdurchlässigkeit
K5	Krit. 5.2	Existenz eines numerischen Wertes zum Wärmedurchgangskoeffizient
K5	Krit. 5.3	Existenz eines numerischen Wertes zum Gesamtenergiedurchlassgrad
K5	Krit. 5.4	Existenz eines numerischen Wertes zum Schalldämmmaß
K7	Krit. 7.1	Transparenz der Fensterscheibe des gerenderten Fensterobjekts
K7	Krit. 7.2	Kennzeichnung des Bewegungsbereiches im Schnitt
K7	Krit. 7.3	Kennzeichnung des Bewegungsbereiches in der Ansicht

Aufbau und Durchführung der Untersuchung

Es wird zunächst eine Testumgebung in Autodesk Revit 2016 modelliert. Das Projekt besteht aus verschiedenen Bauteilen, wie in Abbildung 3 dargestellt.

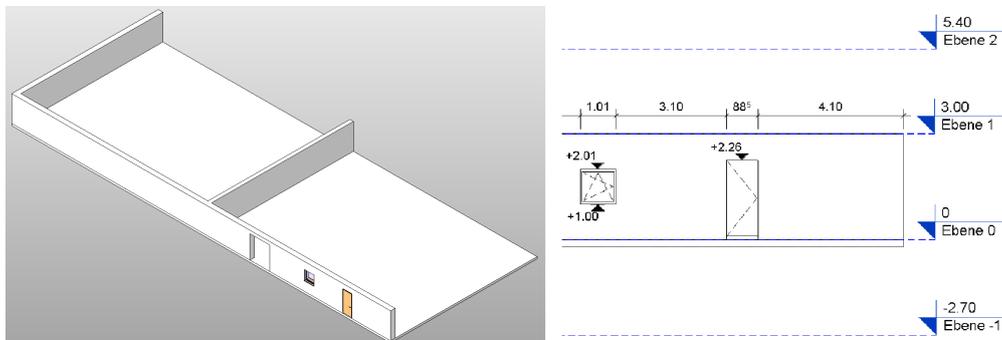


Abb. 3: Testumgebung, modelliert in Autodesk Revit 2016

Anschließend wurden verschiedene herstellerspezifische Fenstertypen ausgewählt, die in kostenfreien Online-Bibliotheken frei verfügbar sind. Es wurde versucht eine repräsentative Auswahl zu treffen, sodass Fenstertypen von verschiedenen Online-Portalen berücksichtigt werden. Zudem wurden von den Herstellern *Schüco* und *Kolbe* jeweils zwei Fenstertypen ausgewählt. Als Referenzobjekt wird die Revit Familie *Fenster 1-flg - Anschlag* zur Untersuchung herangezogen. Die Fensterfamilie ist ein generischer Bauteiltyp und entspricht keinem realen Bauteil. Der Vergleich mit der Familie soll während der Untersuchung Aufschlüsse darüber geben, inwieweit das Nichterfüllen der Kriterien aus den Kategorien *K2* und *K3* auf Einschränkungen seitens der Software zurückzuführen ist. Die zu prüfenden Familien werden von den Online-Portalen heruntergeladen, in der Testumgebung instanziiert und anhand des zuvor aufgestellten Prüfschemas ausgewertet. Die Auswahl ist in Tabelle 2 dargestellt.

Untersuchung von herstellerspezifischen parametrischen Bauteiltypen in einem digitalen Gebäudemodell auf Vollständigkeit der Informationen und Einhaltung bautechnischer Randbedingungen

Tabelle 2: Auswahl der herstellerspezifischen Fensterfamilien

Abk.	Familiename
F1	Fenster 1-flg - Anschlag
	<i>aus: Revit Content Library 2016</i>
F2	Window_Wall_Aluk_58BW_OpenOutVent
	https://www.bimstore.co.uk/ 12.05.2016
F3	efco_corporation_-_windows_-_projected_-_2900c010_10430
	http://seek.autodesk.com/ 14.05.2016
F4	Kolbe_Windows_and_Doors-Ultra_Window_Push_Out_Awning_Combination_2Wx2H-3D
	https://www.polantis.com/ 14.05.2016
F5	Window-Single-Hueck-Lambda_WS075-Tilt_and_turn
	http://bimobject.com/de 15.05.2016
F6	Schueco_AWS-75-SI+_Family-01
	<i>heruntergeladen von:</i> http://bimobject.com/de 16.05.2016
F7	Schueco AWS 90 SI+ type 1 triple glass 40mm
	http://bimobject.com/de 16.05.2016
F8	Window-Fixed-Kolbe-Ultra Series_Fixed_Doors-Inswing_Transom_Direct-Set
	http://library.smartbim.com/ 19.05.2016

Für die *Statistische Auswertung (K1)* wurde ein Revit Addin entwickelt, das auf Basis der Breitensuche die verschachtelten Familien traversiert. Für das Addin wird die Revit API 2016 genutzt. Als Ergebnis werden die Verschachtelungstiefe des Fensterobjektes und die Anzahl der verschachtelten Familien zurückgegeben.

Die Einbaubarkeit wird mithilfe von Versuchen überprüft. Zuerst wird versucht das Fensterobjekt außerhalb der Wandgeometrie zu instanziiieren.

Anschließend wird getestet, ob das Fenster innerhalb der Wandgeometrie eingebaut werden kann. Sofern das möglich ist, wird versucht das Fensterobjekt in alle möglichen Richtungen, wie in Abbildung 4 dargestellt, zu verschieben.

Hierbei wird geprüft, ob ein bautechnisch nicht umsetzbarer Zustand erzeugt werden kann, wie z.B. die Positionierung eines Fensters an einem Ort in einer Wand, an dem bereits ein Fenster oder auch eine Tür eingebaut ist. Für die Auswertung der Kriterien werden verschiedene Ansichten, Schnitte und die 3D-Darstellung betrachtet.

Nach Durchführung der Versuche werden die Kriterien der Kategorie *Vollständigkeit und Korrektheit der Informationen (K5)* ausgewertet. Hierzu werden die Parameter der jeweiligen Fensterfamilie genauer betrachtet um festzustellen, ob ein Parameter für die jeweilige Information vorhanden ist. Existiert der gesuchte Parameter und ein dazugehöriger numerischer Wert, ist das entsprechende Kriterium erfüllt. Um sicherzustellen, dass die Information korrekt ist, sollte der Wert außerdem mit den Angaben des Herstellers abgeglichen werden. In dieser Untersuchung wurde jedoch nur die Existenz bzw. die Vollständigkeit geprüft.

Für die Untersuchung des *Kriteriums 7.1* wird ein Rendering des Fensterobjektes erstellt. Hierbei werden keine zusätzlichen Einstellungen an dem Objekt vorgenommen. Das so erzeugte Bild wird anschließend ausgewertet.

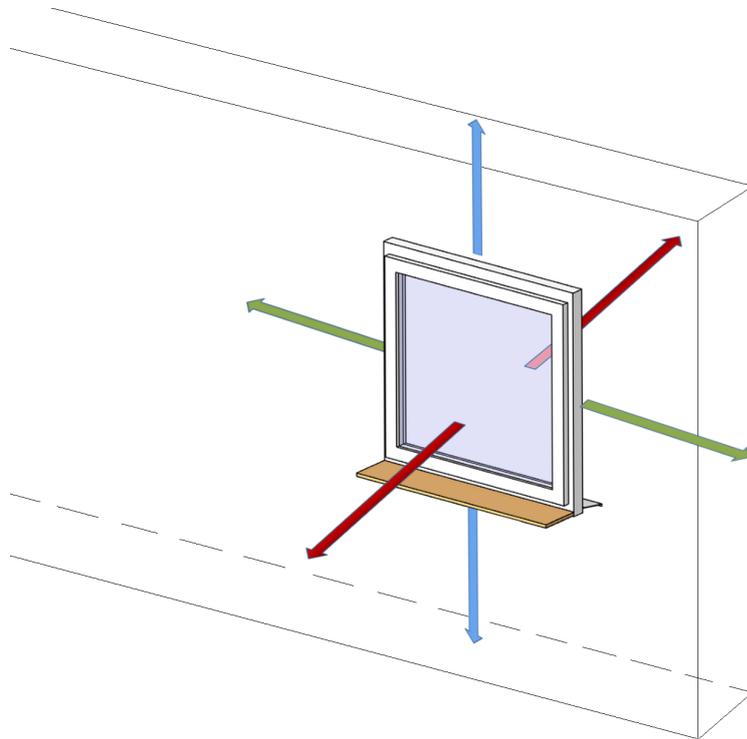


Abb. 4: Schematische Darstellung eines Verschiebungsversuchs eines zu prüfenden Fensterobjektes

Für weiteren Kriterien der Kategorie *Visualisierung (K7)* werden die Ansichten und Schnitte der Testumgebung untersucht, in denen das Fensterobjekt dargestellt wird. Hierbei wird geprüft, ob die Bewegungsbereiche, wie in Abbildung 5 dargestellt, visualisiert wurden. Die Ergebnisse der Untersuchung wurden in Tabelle 3 zusammengetragen.

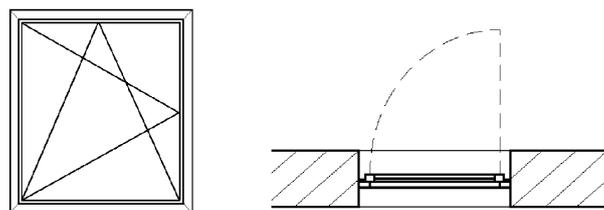


Abb. 5: Öffnungssymbole eines Fensters, links: Ansicht, rechts: Grundriss

Untersuchung von herstellerspezifischen parametrischen Bauteiltypen in einem digitalen Gebäudemodell auf Vollständigkeit der Informationen und Einhaltung bautechnischer Randbedingungen

Tabelle 3: Auswertung der Kriterien aus Tabelle 1

Krit.	F1	F2	F3	F4	F5	F6	F7	F8
1.1	0	1	1	3	2	3	1	1
1.2	0	13	4	74	15	5	6	4
2.1	•	•	•	•	•	•	•	•
3.1			•	•	•			•
3.2								
4.1	*	•			•			
4.2	*	•						
4.3	*	•		•	•	•	•	•
5.1	•							
5.2	•				•		•	
5.3	•							
5.4		•				•	•	
7.1	•		•			•	•	•
7.2					•			**
7.3	•	•			•	•	•	**

Anmerkungen:

- : Der Fenstertyp erfüllt das Kriterium.
- *: F1 ist ein generischer Fenstertyp. Die Kriterien aus der Kategorie *Support (K4)* können nicht ausgewertet werden.
- **: F8 besteht aus einem Fensterflügel, der nicht geöffnet werden kann.

Auswertung und Fazit

Die Untersuchung zeigt, dass jede der Fensterfamilien in eine bestehende Wand eingebaut werden konnte und eine dazu passende Wandöffnung erzeugt wurde. Hierbei ließen sich jedoch 4 der 8 untersuchten Fenster auch außerhalb der Wandgeometrie einbauen. Weiterhin war ein Einbau der Fenster in eine Wandkreuzung möglich.

Die untersuchten Fensterfamilien wurden sehr unterschiedlich modelliert. Wie die numerische Auswertung zeigt, variiert der Grad, wie tief die Familien verschachtelt wurden. Bei der Fensterfamilie *F4* bleibt beispielsweise fraglich, inwieweit sich die hohe Verschachtelung negativ auf das Lauzeitverhalten auswirkt und somit eine ausreichende Anwendbarkeit in digitalen Gebäudemodellen gewährleistet ist.

Auch die Familienparameter wurden unterschiedlich modelliert. Die Bezeichnungen der Parameter, die die gleiche Information repräsentieren, unterscheiden sich teilweise zwischen den untersuchten Familien und häufig existieren innerhalb einer Familie mehrere Parameter für eine Information. Die Intention des Entwicklers ist hierbei nicht immer klar ersichtlich. Erschwert wird dies zudem durch den Mangel an Zusatzinformationen, wie eine Erklärung der Parameter und eine Bedienungsanleitung des Bauteiltyps. Die Untersuchung zeigt zudem, dass zwar häufig Parameter für die unter *Kategorie K5* geforderten Informationen vorhanden sind, aber diese meist ohne Wertebelegung sind. Die Erwartung, dass die Hersteller ihren Bauteiltypen diese Informationen mitgeben würden, konnte nicht bestätigt werden.

Auch bei der Visualisierung der Bauteiltypen fallen Unterschiede auf. Unter anderem fehlen 6 von 8 der untersuchten Fensterfamilien die Kennzeichnung vom Bewegungsbereich des Fensterflügels im Schnitt und in der Ansicht. Zudem ist bei 3 von den 8 untersuchten Familien die Transparenz der gerenderte Fensterscheibe nicht erkennbar, weil die dafür notwendigen Materialeinstellungen nicht vorgenommen wurden.

Die Untersuchung in diesem Beitrag hat gezeigt, dass viele der betrachteten Bereiche durchaus Einfluss auf die Anwendbarkeit eines digitalen Bauteils in einem Bauwerksmodell haben. Für fortführende Untersuchungen wäre es sinnvoll automatisierte Prüfverfahren zu entwickeln. So könnte beispielsweise die Gleichheit der Durchführung der Versuchsprüfungen sichergestellt werden, sodass jede Verschiebung und Positionierung für beliebig viele Bauteile exakt wiederholt werden kann. Zudem wäre ein interessanter Ansatz, Bauteiltypen gemäß ihrer Eignung für gewisse Anwendungen zu zertifizieren, wie z.B. einer bauphysikalischen Simulation. So weiss der Anwender für welche Zwecke er den Bauteiltyp nutzen kann und er kann seine Auswahl leichter treffen.

Die Hersteller entwickeln ihre Bauteiltypen derzeit noch sehr unterschiedlich. Nichtsdestotrotz entsteht mit der Digitalisierung der Bauprodukte ein neuer Markt, der zunehmend an Bedeutung gewinnen wird. Wie auch für reale Bauteile gilt es daher geeignete Prüfverfahren zu entwickeln um eine ausreichende Anwendbarkeit der Bauteilobjekte in Bauwerksmodellen sicherzustellen.

Literatur

- Autodesk, Inc. (2009a). *Autodesk ®Seek Metadata Style Guide*.
- Autodesk, Inc. (2009b). *Revit Model Content Style Guide. Version 2.1*.
- BIMForum (2015). *Level of Development Specification*.
- BIM-Store (2015). *bimstore bible - Revit family creation standards. Version 13. UK Edition*.
- Eastman, C., Eastman, C. M., Teicholz, P., Sacks, R. und Liston, K. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons.
- Graphisoft (2015). *ArchiCADGDLReferenzhandbuch*.
- Nicholson-Cole, D. (2004). *The GDL Cookbook 4*.
- RIBA Enterprises Limited (2014). *The NBS BIM Object Standard*.
- Robaszewski, E. (2016). *Verfügbarkeit und Anwendbarkeit parametrischer Bauteiltypen für digitale Gebäudemodelle*. Fachgebiet Bauinformatik TU Berlin.

Automatische Prüfung und Filterung in BIM mit Model View Definitions

Ken Baumgärtel¹ und Stephan Pirnbaum²

Institut für Bauinformatik, TU Dresden, Nürnberger Straße 31a, Deutschland

¹E-Mail: ken.baumgaertel@tu-dresden.de, ²E-Mail: stephan.pirnbaum@tu-dresden.de

Abstract Zur Unterstützung der Interoperabilität mehrerer Softwareanwendungen im Bauwesen, basierend auf Building Information Modeling (BIM), werden nicht nur technische Schnittstellen oder standardisierte Datenformate, wie die Industry Foundation Classes (IFC) zum Austausch von Gebäudeinformationen (Exchange Requirements), sondern auch Model View Definitions (MVDs) verschiedener BIM-Domänen und beteiligter Fachplaner benötigt. Eine Model View ist ein spezifischer Ausschnitt einer Informationsmenge in einer vordefinierten Struktur und drückt Nutzeranforderungen an ein Gebäudemodell aus. Die MVDs werden von buildingSMART als mvdXML mit Hilfe eines XML Schemas spezifiziert. Sie erlauben die Definition bestimmter benötigter IFC-Klassen, -Attribute und -Datentypen im Gebäudemodell. Zum Beispiel können mit einer MVD Eigenschaftsmengen (Property Sets) für Wärmedurchgangskoeffizienten definiert werden, die allen Wänden zugewiesen werden müssen oder aber es wird die Existenz von Räumen und die Beziehung von Räumen zu Bauteilen mittels Raumbegrenzungen vorausgesetzt. Dieser Beitrag widmet sich einer zweistufigen Methode, um ein Gebäudemodell basierend auf einer Model View Definition prüfen und auch ein Teilmodell extrahieren zu können. Hierfür wird eine mvdXML-Datei genutzt, um daraus Filter- und Check-Abfragen mit der IFC Query Language (ifcQL) zu generieren und diese auf eine IFC-Datei anzuwenden.

Keywords: BIM, MVD, mvdXML, Filterung, Validierung, IFC, ifcQL

1 Einleitung

Softwareanwendungen nutzen MVDs, um die Bereitstellung benötigter Informationen im Gebäudemodell zu überprüfen. Eine beispielhafte Anwendung ist ifcDoc von buildingSMART, welches den Planern erlaubt MVDs in mvdXML zu erzeugen und eine IFC-Datei basierend auf diesen Anforderungen zu validieren. Die Anforderungsprüfung ist eine wichtige Aufgabe, insbesondere da Gebäudeinformationen stetig wachsen und verschiedene Baubereiche eine Vielzahl unterschiedlicher Daten benötigen, welche nicht relevant für jeden Beteiligten sind. Daher wird in diesem wissenschaftlichen Beitrag ein Konzept vorgestellt, welches die Prüfung und auch die Filterung von BIM-Daten basierend auf MVDs mit mvdXML erlaubt. Das Konzept erweitert die Prüfung des Gebäudemodells um die Möglichkeit der Reduktion der bereitgestellten Informationen zu einem neuen Teilmodell und besteht aus drei Schritten: (1) Transformation der mvdXML-Datei zu Filter- und Check-Abfragen mit Nutzung der ifcQL, (2) Prüfung des ursprünglichen Gebäudemodells mit diesen Anweisungen und (3) Ausführung

der Anweisungen und Extraktion der benötigten Informationen aus einer Original-IFC-Datei und Export einer neuen IFC-Datei. Es wird zuerst der aktuelle Stand der Forschung im Bereich der Filterung und Validierung im BIM aufgezeigt, bevor der Konvertierungsprozess basierend auf mvdXML und die Abfragesprache ifcQL vorgestellt wird. Die ifcQL bietet strukturierte Funktionen zur Abfrage von Gebäudeinformationen und ist für Ingenieure entwickelt worden, welche keine Programmierkenntnisse besitzen. Das Konzept wird mit Beispielen des entwickelten Prototyps und seinen Algorithmen zur Verarbeitung von IFC- und mvdXML-Daten gezeigt.

2 Stand der Filterung und Validierung im BIM

Die IFC sind so angelegt, dass möglichst alle Bereiche im Bauwesen abgedeckt werden können. Dieser Vorsatz führt aber auch zu Problemen verschiedenster Art. Pauwels et al. (2011) beschreiben das Schema als zu allgemein, da es für den gesamten AEC-Bereich gelten soll. Durch die vielen Ingenieursbereiche, die sich teilweise durch ihre unterschiedlichen Sichten stark beeinflussen, entstehen in einem Schema zwangsläufig optionale Datentypen. In einer Domäne können somit Informationen fachspezifisch ausgedrückt werden, während in einer anderen Domäne diese Informationen unterschiedlich dargestellt oder gar nicht benötigt werden. Dieser Umstand führt zu mehreren Möglichkeiten bei der Modellierung eines Gebäudes. Dadurch besteht die Gefahr, dass jeder Fachplaner eigene *IfcPropertySets* haben möchte, die sich unter Umständen überschneiden können und dadurch das Gebäudemodell inkonsistent wird. Die Angabe des Wärmedurchgangskoeffizienten einer Wand obwohl die Materialschichten und -dicken angegeben sind, kann beispielsweise bei Änderungen zu Inkonsistenzen führen. Würde der Konstruktionsaufbau verändert werden, so muss auch eine Anpassung des Wärmedurchgangskoeffizienten geschehen. Es besteht also eine Korrelation dieser Angaben, aber die Softwareprogramme wissen eventuell nichts davon. Dies kann zu einem redundanten Datenmodell führen, welches zu viele Optionen bietet anstatt Einschränkungen zu treffen, die gerade in der Softwareentwicklung vonnöten sind, um stabile Anwendungen und Schnittstellen implementieren zu können. Es können demzufolge Probleme durch verschiedene Interpretationen der Softwareentwickler unzuverlässige Datenaustausche auftreten (Sacks et al., 2010). Für einige Aufgaben, wie zum Beispiel der thermischen Energieberechnung eines Gebäudes, besitzt das IFC-Schema eine limitierte Semantik, um definieren zu können, welche Informationen ausgetauscht und wie sie angegeben werden sollten (Venugopal et al., 2012). Im Schema wird somit durch die Vielzahl optionaler Attribute und die Möglichkeit der Angabe von Objekteigenschaften mittels *IfcPropertySets* eine einheitliche Arbeitsweise und die Erhaltung einer bestimmten Modellqualität erschwert.

Bei der Filterung, d.h. der Informationsreduktion von Modellen, stellt sich zunächst die Frage, welche Informationen benötigt werden (Informationsanforderungen). Katranuschkov et al., 2010 klassifizieren die unterschiedlichen Informationsanforderungen durch folgende Arten von *Model views*: a.) *Domain Model View*, b.) *Ad-hoc Model View* und c.) *Multimodel View*. Bei der *Domain model view* werden die aus einem beliebigen Gebäudemodell zu extrahierenden Informationen mittels eines vordefinierten Subschemas beschrieben. *Domain model views* entsprechen dem durch buildingSMART eingebrachten Ansatz der *Model View Definitions* (Hietanen, 2006), welche die Informationsanforderungen für ein spezifisches Austauschszenario auf der Grundlage eines *Information Delivery Manuals (IDM)* definieren (Wix und Karlshoej, 2010).

Die *Ad-hoc Model View* bezieht sich auf Informationen aus einem konkreten Modell. Die zu

extrahierenden Informationen werden hier nicht im Voraus, sondern spontan spezifiziert. In der Multimodel View werden, ähnlich der Ad-hoc Model View, konkrete Informationsanforderungen spezifiziert. Diese beziehen jedoch zusätzliche Informationen aus anderen Modellen (z.B. Kosten) mit ein, die sich mit dem Gebäudemodell verknüpfen lassen (z.B. Kosten aller Wände in der zweiten Etage). Die Entwicklung und Nutzung der o.g. Model Views kann somit hinsichtlich der Aspekte Zeit- und Detailierungsgrad unterschieden werden, indem die Definition welche (Teil-) Datenmenge zu extrahieren ist entweder vorher, d.h. losgelöst von der eigentlichen Reduktion der Daten eines konkreten Modells, oder basierend auf Informationen eines konkreten Modells erfolgen kann (Windisch et al., 2012). Damit die, durch die o.g. Modellsichten definierten, Informationsanforderungen realisiert werden können (Informationsextraktion) bedarf es der Umsetzung verschiedener Filterarten, welche in (1) Filtern auf Schemaebene, (2) Filtern auf Klassenebene, (3) Filtern auf Objekt- bzw. Instanzebene und (4) Filtern auf Schlussfolgerungsebene unterschieden werden. Da, bedingt durch die zeitlich vorgelagerte Definition, keine Kenntnis über konkrete Instanzen bzw. modellspezifische Aspekte (z.B. Bauteilabmessungen, verwendete Materialien) vorliegen können, ist der mögliche Detailierungsgrad bei der Definition von Domain Model Views entsprechend geringer.

3 Filterung und Validierung mit mvdXML

Eine Model View Definition wird zur Erstellung eines Subschemas der IFC genutzt. Dadurch können eine oder mehrere Exchange Requirements definiert werden, um den Datenaustausch in einer bestimmten Qualität zu ermöglichen. Durch die Möglichkeit der unterschiedlichen Modellierungen mit IFC, zum Beispiel der Verwendung verschiedener Geometrirepräsentationen, und dem stetigen Wachstum neuer BIM-Softwareanwendungen werden Model Views immer wichtiger. Die Model View Definitions können hierbei als Vertrag zwischen Anwendern und Programmen gleichermaßen verstanden werden. Sie werden zum Beispiel auch zur Zertifizierung von BIM-Software durch buildingSMART verwendet. Die benötigten Detailebenen und ausgetauschten Daten können exakt mit Hilfe von Klassen, Attributen und Property Sets beschrieben werden. Hierfür wird die offizielle Softwareanwendung ifcDoc von buildingSMART genutzt (buildingSMART, 2016). Mit ifcDoc lassen sich die MVDs als mvdXML exportieren und Anforderungen auch durch importieren eines Gebäudemodells prüfen. Zhang et al. (2014) entwickelten hierfür den Prototypen mvdXMLChecker der Model View Definitions, ausgedrückt durch mvdXML, einlesen und Gebäudemodelle basierend auf dem IFC-Schema prüfen kann (Zhang et al., 2014; Zhang et al., 2015). Schlagen ein oder mehrere Prüfungen in der Software fehl, so wird ein Bericht generiert, der alle Probleme auflistet (Laa, 2016). Der Prototyp wird derzeit für den BIMserver genutzt (opensourcebim.org, 2016) und kann auch problemlos in anderen Programmen integriert werden, um Model View Definitions auf Gebäudemodelle anwenden zu können.

Die Filterung von IFC-Dateien ist mit ifcDoc und dem mvdXMLChecker allerdings nicht möglich. Dabei eignet sich die Beschreibung der Model Views vorzüglich zur Extraktion der definierten Exchange Requirements in ein Teilmodell. Die Gesamtdatenmenge kann so minimiert werden, um nur die benötigten Informationen bereitzustellen. Dies hat auch den Vorteil, dass nur relevante Projektdaten kommuniziert und vertrauliche Informationen somit exkludiert werden können. Die Filterung wird in dem im Folgenden vorgestellten Konzept durch den mvdXML-Converter und die Nutzung des ifcQL-Processors umgesetzt. Es wird zuerst auf den Mapping-Prozess und danach auf die Abfragesprache ifcQL eingegangen.

3.1 Mapping-Prozess

Durch einen zweistufigen Prozess wird die Anwendung einer mvdXML zur Extraktion eines Teilmodells ermöglicht. Abbildung 1 verdeutlicht die Prozesskette:

1. Die Nutzung des mvdXML-Converters zur Umwandlung einer mvdXML-Datei in eine äquivalente ifcQL-Datei. Dabei kann angegeben werden, ob eine Validierungsanfrage oder eine Filteranfrage generiert werden soll.
2. Die Anwendung des ifcQL-Processors durch den Import einer IFC-Datei und der ifcQL-Datei mit den konvertierten Befehlen. Die Ausgabe ist dann entweder eine reduzierte IFC-Datei (Filterung) oder eine LOG-Datei (Validierung).

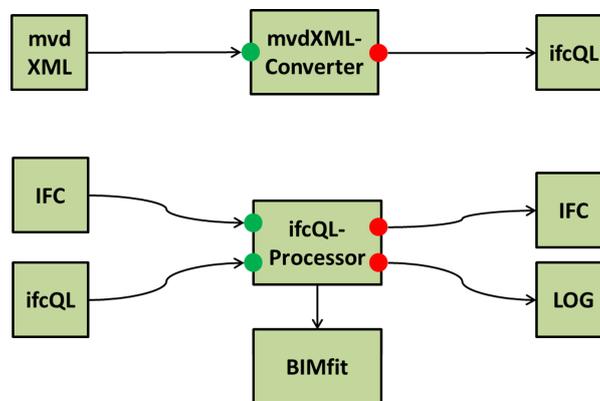


Abb. 1: Die Prozesskette im Konvertierungsprozess

Die zweite Stufe wird zusammen mit der ifcQL im nächsten Abschnitt erläutert. Bei der ersten Prozessstufe wird wie folgt vorgegangen:

Einlesen der mvdXML-Datei Es wird von dem mvdXML-Converter der XML-Inhalt der Model View geparkt.

Instanziierung der MVD-Objekte basierend auf dem Inhalt Die XML-Elemente werden ausgewertet und die jeweiligen Java-Klassen gesucht und instanziiert. Die mvdXML Spezifikation Version 1.0 wird herangezogen, um ein Netz der definierten Exchange Requirements aufzubauen (Hietanen, 2006).

MVD-Objekte werden schrittweise in ifcQL-Befehle umgewandelt Nachdem der Objektgraph mit Exchange Requirements aufgebaut wurde, wird Knoten für Knoten dieser Graph in die Struktur der ifcQL umgewandelt.

Export der ifcQL-Befehle Zum Schluss werden die ifcQL-Befehle in der richtigen Reihenfolge in eine ifcQL-Datei exportiert, um ein Gebäudemodell entweder validieren oder filtern zu können.

Das Element <Templates> beinhaltet alle Regeln, welche zur Validierung oder Filterung angewendet werden sollen. Es kann aus mehreren <ConceptTemplate> Elementen bestehen, die wiederum aus mehreren <SubTemplates> zusammengesetzt sein können, um die Strukturierung zu verbessern (Abbildung 2). Für jedes bestehende <ConceptTemplate> wird ein neuer ifcQL-Befehl erstellt (*filter* oder *check* je nach Anwendung).

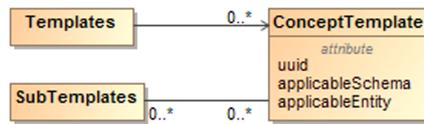


Abb. 2: Die Beziehung zwischen Templates, SubTemplates und ConceptTemplates

Abbildung 7 zeigt ein mvdXML-Beispiel zur Filterung von Wänden, welche mindestens ein Fenster besitzen. Darin und in Abbildung 2 ist zu sehen, wie die Angabe eines `<ConceptTemplate>` Elements in ifcDoc erfolgt. Insbesondere das XML-Attribut `applicableEntity` ist von Bedeutung. Darin wird der IFC-Typ angegeben auf den sich die Anfrage bezieht. Weiterhin wird die Angabe von `<Rules>` veranschaulicht. Die hier angegebenen Elemente werden als Suchanfrage eines Pfades beginnend von der `applicableEntity` benutzt. Das Element `<AttributeRule>` beinhaltet einen `AttributeName`, welches das korrespondierende IFC-Attribut kennzeichnet. Das Attribut `Cardinality` wird ebenfalls verarbeitet und kann 0, 0 : 1, 1 oder 1 : n sein (mögliche Repräsentationsformen in mvdXML und ifcQL, siehe Tabelle 1). Wie in dem Beispiel in Abbildung 7 zu sehen ist, kann eine `<AttributeRule>` auch mehrere `<EntityRule>` Elemente beinhalten, welche wiederum aus `<AttributeRule>` Elementen bestehen kann. Eine `<EntityRule>` hat die Attribute `EntityName` und `Cardinality`.

Tabelle 1: Überführung der Kardinalitäten

Kardinalität	mvdXML	ifcQL
1 : n	<code>_asSchema</code>	ONETOMANY
0	Zero	ZERO
0 : 1	ZeroToOne	ZEROTOONE
1	One	ONE
1 : n	OneToMany	ONETOMANY

Zur besseren Strukturierung ist es möglich, `<Views>` einzusetzen, um die konvertierten `<Templates>` in mehrere Ausgabedateien zu exportieren. Jedes `<ModelView>` Element kann mehrere `<ExchangeRequirement>` Elemente beinhalten. Für jede Kombination dieser Elemente erzeugt der mvdXML-Converter eine ifcQL-Datei, die nach der jeweiligen Modellsicht benannt ist. Abbildung 3 veranschaulicht die Beziehungen von `<Concept>` Elementen zu `<ModelView>` Elementen. Jedem `<Concept>` muss ein Name (`name`), der IFC-Typ (`applicableRootEntity`) und ein `<Template>` zugeordnet sein. Das Attribut `applicableRootEntity` kann genutzt werden, um `applicableEntity` eines `<ConceptTemplate>` Elements modifizieren zu können.

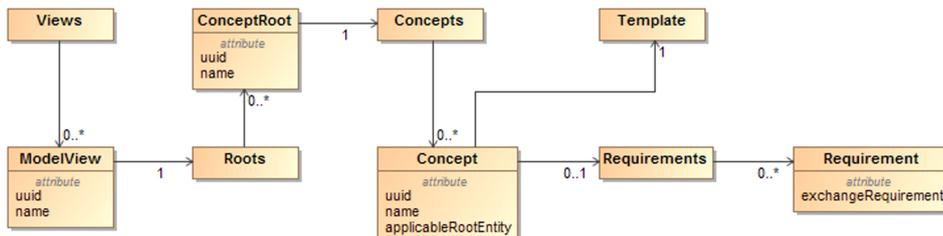


Abb. 3: Die Beziehung zwischen ModelViews und ExchangeRequirements

3.2 Die Abfragesprache ifcQL

Zur Definition von MVDs und deren Anwendung auf IFC-Daten wird eine relationale Algebra in einer eigens dafür entwickelten Syntax verwendet. Sie ähnelt der Sprache SQL und wird IFC Query Language (ifcQL) genannt. Der Hauptzweck ist nicht nur (1) die Validierung von Gebäudedaten, sondern auch (2) die Extraktion der spezifizierten Informationen (=Informationsmenge) aus dem Gebäudemodell und der Erstellung einer Version des Gebäudemodells in einer neuen IFC-Datei. Es lässt sich somit ein reduziertes Gebäudemodell erstellen, welches mit weiteren Anwendungen ausgetauscht werden kann. Die Abfragesprache ermöglicht den Zugriff auf Gebäudeinformationen ohne Programmierkenntnisse besitzen zu müssen. Die ifcQL integriert, ähnlich dem Datenbankkonzept, das CRUD-Prinzip mit den Funktionen:

Create Erzeugung eines Datensatzes.

Read Lesen eines Datensatzes.

Update Verändern eines Datensatzes.

Delete Löschen eines Datensatzes.

Die MVD mit ifcQL wird in einer oder mehreren Dateien definiert, wobei ein Befehl pro Zeile angegeben ist. Dabei kann festgelegt werden, ob sich eine neue Zeile auf die vorhergehende bezieht oder eine neue Menge erzeugt wird. Die ifcQL stellt eine Vielzahl an Funktionen bereit, um ein Gebäudemodell nicht nur validieren, sondern auch reduzieren, erweitern oder verändern zu können. In ifcQL sind folgende Befehle enthalten:

filter zur Reduktion der Informationsmenge.

check zur Validierung einer Informationsmenge.

add zur Erzeugung neuartiger Entitäten zur Informationsmenge.

insert zum Hinzufügen von Informationen zur Informationsmenge (zum Beispiel Property Sets).

update zur Modifikation einer bestehenden Informationsmenge.

delete zur Entfernung einer Teilmenge aus der Informationsmenge.

load zum Importieren bestehender MVDs als Teil einer MVD.

Die Befehle werden in einer eigenen Syntax definiert und fangen mit “~” an. Die Angaben von bestimmten Entitätstypen, Attributen oder anderen Bedingungen werden danach mit “-” definiert. Das Syntaxdiagramm für die initiale Angabe von Befehlen ist in Abbildung 4 zu sehen. Ähnlich dem SQL-Prinzip mit den Befehlen *SELECT*, *FROM* und *WHERE*, kann die Informationsmenge eines Gebäudemodells verarbeitet werden. Das *FROM* bezieht sich hierbei auf die Menge an IFC-Entitäten durch Angabe von Typen oder Identifikatoren. Das *SELECT* ist jeweils einer der oben genannten Befehle und das *WHERE* bezieht sich meist auf die konkrete Angabe von gesetzten Objektattributswerten, zum Beispiel die Angabe des Attributs *GlobalId* und dem gesuchten Bezeichner, mit denen die Auswahl beeinflusst wird.

Die beiden wichtigsten Befehle sind *check* und *filter*. Sie sind in den Syntaxdiagrammen in den Abbildungen 5 und 6 zu sehen. Bei der Validierung können zum einen Objekte mit bestimmten Bedingungen (Objektsemantik) und zum anderen Relationen zwischen Objekten (Relationssemantik) überprüft werden. Dies erlaubt nicht nur die Angabe von Attributswerten,

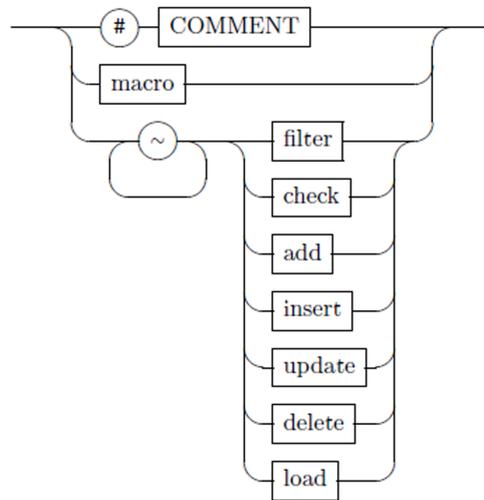


Abb. 4: ifcQL-Befehle

sondern auch die Einordnung von Objekten. Mit dem *relationcheck* ist es beispielsweise möglich, die Zuordnung von Räumen zu einer bestimmten Etage prüfen zu können. Das Ergebnis der Prüfung ist eine Log-Datei, welche alle Prüfungen und deren Ergebnisse beinhaltet.

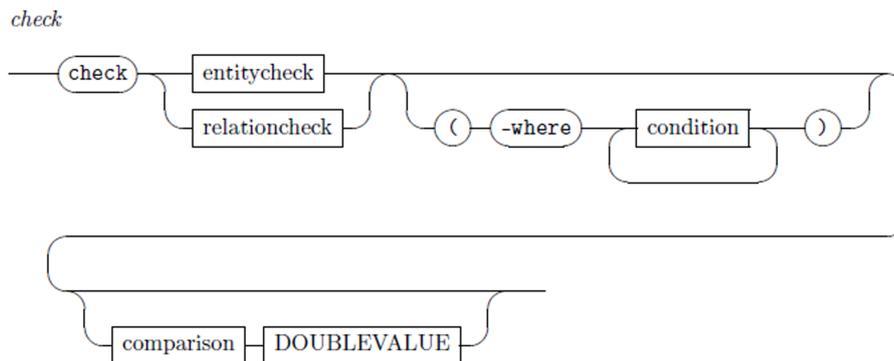


Abb. 5: Check-Befehl

Bei der Filterung ist das Resultat ein Teilmodell, dessen Informationsgehalt dem zugeordneten Subschema entspricht. Das Filtern auf Klassenebene reduziert, wie auch das Filtern auf Schemaebene, BIM-Modelle auf Basis von Typeinschränkungen und Relationen, bezieht sich jedoch auf konkrete Modelle (z.B. filtere alle Bauteile eines bestimmten Typs aus dem Gebäudemodell). Der *filter*-Befehl ist deshalb geprägt von der Angabe eines Typs ("-type") bzw. einer bestimmten *GlobalId* zur Suche eines exakten Objekts ("-guid") oder aber zur Exkludierung (*exclude*) eben dieser Teilmenge von der Ergebnismenge. Mit "-recursive" können zusätzlich noch alle Entitäten zur Ergebnismenge eingeschlossen werden, welche mit der Entität direkt verlinkt sind.

Die ifcQL bietet noch viel mehr Möglichkeiten mit den Daten eines Gebäudemodells arbeiten

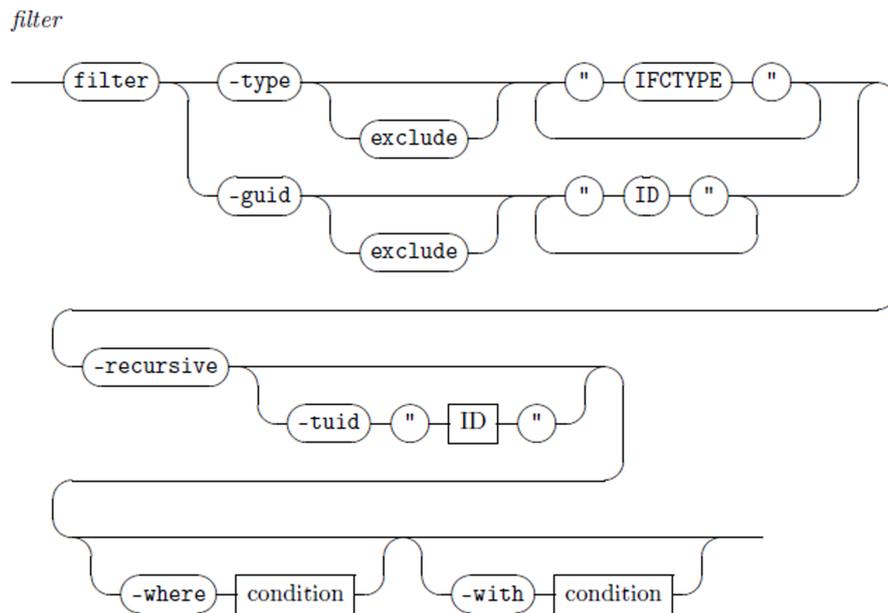


Abb. 6: Filter-Befehl

zu können. Die ifcQL basiert auf BIMfit, welches ein Java-basiertes Framework ist, mit dem STEP-Daten programmatisch manipuliert werden können (Wülfing et al., 2012).

4 Anwendungsbeispiel

Wie bereits erwähnt, ist in Abbildung 7 eine einfache MVD dargestellt, die festlegt, dass alle Wände Fenster besitzen müssen. Es wird ein Pfad von dem Typ *IfcWall* durch Angabe der einzelnen Attributnamen angegeben, um die jeweilig verbundenen Fensterobjekte zu erreichen. Dieses <ConceptTemplate> wird dem Exchange Requirement "WallsWithWindows" zugewiesen und einer Modellsicht angehängen. Die dazugehörige Umwandlung in den (a) Filter- bzw. (b) Check-Befehl ist in Abbildung 8 dargestellt. Mit dem ifcQL-Processor können nun beliebige Gebäudemodelle eingelesen und reduziert bzw. geprüft werden. In Abbildung 9 ist ein Gebäude (a) in der Originaldatei vor und (b) nach der Informationsreduktion basierend auf der MVD zu sehen. Wie in der MVD definiert, sind nur alle Wandobjekte in der Informationsmenge aufgeführt, welche auch wirklich Fenster haben. Alle anderen Bauteile wurden aus der Ergebnismenge gefiltert.

Bei dem Export einer neuen IFC-Datei durch definierte Filteranweisungen wird in allen Fällen ein syntaktisch und semantisch korrektes Gebäudemodell ausgegeben, welches in einem IFC-Viewer angeschaut werden kann. Hierbei wird der IFC Model Implementation Guide von buildingSMART beachtet (Liebich, 2009). Die räumliche Struktur mit *IfcProject*, *IfcBuilding* und *IfcBuildingStorey* mit deren Relationen bleibt zum Beispiel in jedem Falle erhalten.

```

<mvdXML xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  uuid="6080c980-c0fd-4c54-80e5-3a801cc9556e" name="" status="sample"
  xsi:schemaLocation="http://buildingsmart-tech.org/mvd/XML/1.1
  http://www.buildingsmart-tech.org/mvd/XML/1.1/mvdXML_V1.1_add1.xsd"
  xmlns="http://buildingsmart-tech.org/mvd/XML/1.1">
  <Templates>
    <ConceptTemplate uuid="3019" name="WallWithFillings"
      status="sample" applicableSchema="IFC4" applicableEntity="IfcWall">
      <Rules>
        <AttributeRule AttributeName="HasOpenings">
          <EntityRules>
            <EntityRule EntityName="IfcRelVoidsElement">
              <AttributeRules>
                <AttributeRule AttributeName="RelatedOpeningElement">
                  <EntityRules>
                    <EntityRule EntityName="IfcOpeningElement">
                      <AttributeRules>
                        <AttributeRule AttributeName="HasFillings">
                          <EntityRules>
                            <EntityRule EntityName="IfcRelFillsElement">
                              <AttributeRules>
                                <AttributeRule AttributeName="RelatedBuildingElement">
                                  <EntityRules>
                                    <EntityRule EntityName="IfcWindow" />
                                  </EntityRules>
                                </EntityRule>
                              </EntityRules>
                            </EntityRule>
                          </EntityRules>
                        </AttributeRule>
                      </EntityRules>
                    </EntityRule>
                  </EntityRules>
                </AttributeRule>
              </EntityRules>
            </EntityRule>
          </EntityRules>
        </AttributeRule>
      </Rules>
    </ConceptTemplate>
  </Templates>
  <Views>
    <ModelView uuid="4de5" name="FBI2016" status="sample" applicableSchema="IFC4">
      <ExchangeRequirements>
        <ExchangeRequirement uuid="a41a" name="WallsWithWindows"
          status="sample" applicability="both" />
      </ExchangeRequirements>
      <Roots>
        <ConceptRoot uuid="97dc" name="IfcWall" status="sample" applicableRootEntity="IfcWall">
          <Concepts>
            <Concept uuid="5c96" name="Wall" status="sample" override="false">
              <Template ref="3019" />
              <Requirements>
                <Requirement applicability="import" requirement="mandatory" exchangeRequirement="a41a" />
                <Requirement applicability="export" requirement="mandatory" exchangeRequirement="a41a" />
              </Requirements>
            </Concept>
          </Concepts>
        </ConceptRoot>
      </Roots>
    </ModelView>
  </Views>
</mvdXML>

```

Abb. 7: Beispiel einer mvdXML mit Filterung von Wänden (IfcWall) mit Fenstern (IfcWindow)

```

#Wall(5c96)                                #Wall(5c96)
-filter -type "Ifcwall" -with path          ~check -type "Ifcwall" having path
ONETOMANY "HasOpenings"("Ifcrelvoidselement") ONETOMANY "HasOpenings"("Ifcrelvoidselement")
.ONETOMANY "RelatedOpeningElement"("Ifcopeningelement") .ONETOMANY "RelatedOpeningElement"("Ifcopeningelement")
.ONETOMANY "HasFillings"("Ifcrelfillselement") .ONETOMANY "HasFillings"("Ifcrelfillselement")
.ONETOMANY "RelatedBuildingElement"("Ifcwindow") .ONETOMANY "RelatedBuildingElement"("Ifcwindow")

```

Abb. 8: Beispielhafte Gegenüberstellung der Filterung und Validierung in ifcQL

5 Zusammenfassung

MVDs werden immer wichtiger, weil der Informationsgehalt in einem Gebäudemodell weiter zunimmt, die Fachplaner unterschiedliche Sichtweisen auf ein Gebäude besitzen und ein einheitlicher Weg zum Austausch benötigter Gebäudedaten in einem BIM-Projekt unterstützt werden sollte. In diesem Beitrag wird das Konzept der Nutzung von Model View Definitions zur Validierung, aber auch Reduzierung von Gebäudemodellen, basierend auf mvdXML vorgestellt. Die Validierung mittels mvdXML wurde bereits von Zhang et al. (2014) und Zhang et al. (2015) vorgestellt. Die Filterung mit mvdXML wurde allerdings bisher nicht unterstützt. Dabei eignen sich die IFC vorzüglich zum Austausch von Gebäudeteilmodellen. Somit kann zwischen den Fachplanern nur die benötigte Informationsmenge ausgetauscht werden, um deren Aufgaben erfüllen zu können. Dies minimiert den Datentransfer und vereinfacht teilweise auch das Gebäude. Das ist im Kontext von Gebäudesimulationen sehr vorteilhaft. So können einzelne Etagen oder Räume einer thermischen Energiesimulation übermittelt werden ohne das

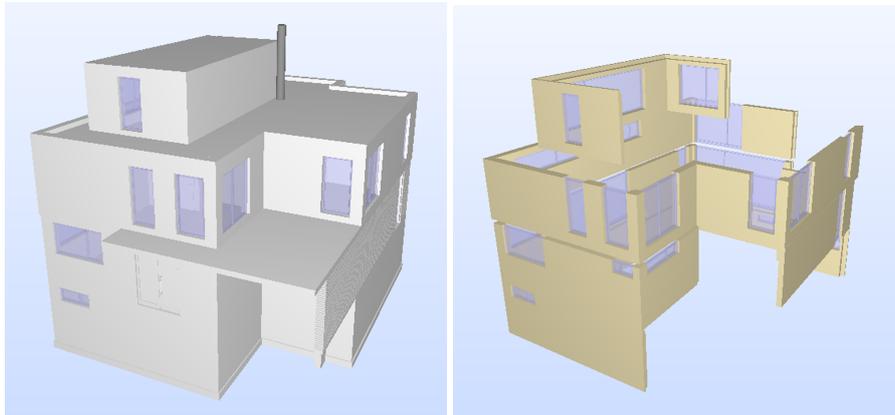


Abb. 9: Beispielhafte Gegenüberstellung vor und nach Nutzung des mvdXML-Converters und ifcQL-Processors

gesamte Gebäudemodell übergeben zu müssen.

Für die Umsetzung dieses Prozesses erfolgt eine zweistufige Konvertierung. Zuerst wird eine mvdXML mit dem mvdXML-Converter eingelesen und in ifcQL-Befehle übersetzt. Dabei muss vom Anwender festgelegt werden, ob eine Validierungs- oder Filterungsbeschreibung gewünscht ist. Die ifcQL bietet hierfür unterschiedliche Befehle an, welche in dem Schritt automatisch erzeugt werden. Sie wurde entwickelt, um Ingenieuren ohne Programmierkenntnisse die strukturierte Abfrage von Gebäudeinformationen zu ermöglichen. Danach erfolgt im zweiten Schritt die Anwendung der erzeugten Abfragen auf ein Gebäude mit dem ifcQL-Processor. Die Ausgabe ist letztlich ein Validierungsbericht oder ein reduziertes Gebäudemodell in IFC (derzeit werden IFC2x3 und IFC4 unterstützt). Die Java-Programme, ifcQL-Processor und mvdXML-Converter, lassen sich in beliebige Softwareanwendungen integrieren. Deren Dokumentation und die Prototypen selbst sind unter <https://openeebim.bau.tu-dresden.de/bimfit.html> zu finden.

Literatur

- buildingSMART International Ltd. (2016). *ifcDoc*. <http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-download-page> (besucht am 15. 07. 2016).
- Hietanen, J. (2006). *IAI Official IFC Model View Definition Format*.
- Katranuschkov, P., Weise, M., Windisch, R., Fuchs, S. und Scherer, R. J. (2010). „BIM-based generation of multi-model views“. In: *27th CIB W78 International Conference*.
- Laat, R. de (2016). *mvdXMLChecker*. <https://github.com/opensourceBIM/mvdXMLChecker> (besucht am 15. 07. 2016).
- Liebich, T. (2009). *IFC 2x Edition 3 Model Implementation Guide*.
- opensourcebim.org (2016). *BIMserver.org*. <http://bimserver.org> (besucht am 15. 07. 2016).
- Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R. und Van Campenhout, J. (2011). „A semantic rule checking environment for building performance checking“. In: *Automation in Construction* 20.5, S. 506–518.

- Sacks, R., Kaner, I., Eastman, C. M. und Jeong, Y.-S. (2010). „The Rosewood experiment—Building information modeling and interoperability for architectural precast facades“. In: *Automation in Construction* 19.4, S. 419–432.
- Venugopal, M., Eastman, C. M., Sacks, R. und Teizer, J. (2012). „Semantics of model views for information exchanges using the industry foundation class schema“. In: *Advanced Engineering Informatics* 26.2, S. 411–428.
- Windisch, R., Wülfing, A. und Scherer, R. (2012). „A generic filter concept for the generation of BIM-based domain-and system-oriented model views“. In: *9th European Conference on Product and Process Modelling (ECPPM), Reykjavik, Iceland*. Hrsg. von G. Gudnason und R. Scherer. CRC Press, S. 311–319.
- Wix, J. und Karlshoej, J. (2010). *Information Delivery Manual Guide to Components and Development Methods*.
- Wülfing, A., Windisch, R. und Baumgärtel, K. (2012). „BIMfit—Ein modulares Softwarewerkzeug zur Abfrage und Filterung von Gebäudeinformationsmodellen“. In: *Proceedings des Forum Bauinformatik 2012*. Bochum, Deutschland.
- Zhang, C., Beetz, J. und Weise, M. (2014). „Model view checking: automated validation for ifc building models“. In: *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2014*. Hrsg. von Mahdavi. Vienna, Austria: CRC Press, S. 123.
- Zhang, C., Beetz, J. und Weise, M. (2015). „Interoperable validation for IFC building models using open standards“. In: *ITcon—Journal of Information Technology in Construction* 20, S. 24–39.

Automatisierte Prozessneugenerierung basierend auf Ergebnissen aus der Baufortschrittskontrolle

Alex Braun¹, Simon Vilgertshofer² und Simon Daum³

Computergestützte Modellierung und Simulation, TU München, Deutschland

¹E-Mail: alex.braun@tum.de, ²E-Mail: simon.vilgertshofer@tum.de,

³E-Mail: simon.daum@tum.de

Abstract Die automatisierte Baufortschrittskontrolle ist derzeit ein zentraler Bereich computerorientierter Forschung im Bauwesen. Dabei wird der Baufortschritt in situ mittels Laserscanning oder photogrammetrischer Methoden erfasst und mit einem 4D Building Information Model abgeglichen. Ein derartiges Modell zeichnet sich durch die kombinierte Vorhaltung geplanter Realisierungszeiträume für Bauteile und ihrer Geometrierepräsentationen aus. Somit kann der vor Ort erfasste *as-built* Status eines Bauteils mit seinem *as-planned* Status innerhalb des 4D-Modells verglichen werden. Treten Abweichungen zwischen tatsächlichen und antizipierten Baufortschritt auf, wird der Prozessplan automatisiert und bauteilgenau angepasst. Dieser Beitrag diskutiert das beschriebene Verfahren und schlägt eine Umsetzung mittels graph-basierter Methoden vor.

Keywords: Baufortschrittskontrolle, Prozesse, Graphen

1 Motivation

In der Ausführungsphase eines Bauprojekts spielt das Baustellenmonitoring eine wichtige Rolle und beinhaltet die Dokumentation von Arbeitspaketen und einzelnen Ausführungsschritten. Somit müssen sämtliche Vorgänge detailliert überwacht werden. Da Monitoring derzeit weitestgehend manuell durchgeführt wird, ergibt sich ein hoher Zeitaufwand. Um dem entgegen zu wirken, ist die Einführung einer automatisierbaren Methodik, die den gesamten Baufortschritt überwacht, erstrebenswert. Insbesondere während großer und sehr großer Bauprojekte ist eine Automatisierung der Überwachungsprozesse für Bauherren lohnenswert. Ein Grund hierfür ist die steigende Anzahl an Subunternehmern bei derartigen Projekten.

Die Bauüberwachung und insbesondere die automatisierte Baufortschrittskontrolle ist ein Forschungsgebiet, das aktuell intensiv untersucht wird. Da sich Building Information Modeling (BIM) zunehmend etabliert, eröffnen sich in diesem Bereich neue Möglichkeiten. So erleichtern die in der BIM-basierter Planung vorgehaltenen semantischen, zeitlichen und räumlichen Informationen die Identifizierung von Bauteile und deren Zuordnung an entsprechende Bauprozessen im Bauablaufplan.

Zur Erfassung des Ist-Zustandes auf der Baustelle werden unterschiedliche Technologien eingesetzt. Darunter fallen Messungen mit Laserscannern und photogrammetrische Aufnahmen,

aber auch die Nutzung von RFID-Tags (Omar und Nehdi, 2016). Durch den Abgleich des sich ergebenden Ist-Bauzustands und der BIM-basierten raum-zeitlichen Planung können Aussagen über tatsächlich vorhandene Bauteile getroffen werden. Daraus lässt sich wiederum ableiten, ob ein Bauteil zum geplanten Zeitpunkt fertiggestellt worden ist.

Die Erfassung des Ist-Zustands auf der Baustelle wurde bereits intensiv erforscht. Der Abgleich der ermittelten Daten mit dem digitalen Gebäudemodell kann auf unterschiedliche Arten erfolgen. Bosché et al. (2010) und Kim et al. (2013) und Turkan et al. (2013) nutzen dazu das Iterative Closest Point (ICP)-Verfahren, bei dem die 3D Geometrie in Punktwolken transformiert und mit der detektierten Punktwolke abgeglichen wird. Golparvar-fard et al. (2009) setzen unter anderem das RANSAC-Verfahren zum Soll-Ist-Abgleich ein. In Braun et al. (2015) werden Punkte direkt mit den Geometrieoberflächen abgeglichen. Alle genannten Verfahren ermöglichen die Erkennung einzelner Bauteile. Folglich kann eine Aussage bezüglich der Fertigstellung eines Bauteils zum Beobachtungszeitpunkt t getroffen werden.

Bisher wurden jedoch keine umfangreichen Untersuchungen angestellt, inwiefern dieses gewonnene Wissen in den Prozessplan integriert werden kann. Insbesondere fehlen generelle Methoden mit denen bestimmt werden kann, wie sich Verzögerungen und Änderungen auf den Prozessplan auswirken. Um die Konsistenz des Prozessplans zu gewährleisten, müssen individuelle Prozessabhängigkeiten berücksichtigt werden.

Zur Darstellung von Prozessplänen eignen sich besonders sogenannte Gantt-Diagramme. Diese enthalten neben der jeweiligen Prozessdauer auch Zwangsfolgen, die durch Abhängigkeitspfeile dargestellt werden. Gantt-Diagramme beinhalten jedoch keine interne Logik und sind nicht direkt rechnergestützt auswertbar. Daher stellen die Autoren einen graphbasierten Ansatz zur Neuberechnung von Bauablaufplänen vor.

2 Aktueller Forschungsstand

Bauprozessplanung und deren Abhängigkeiten wurden bereits von vielen Forschergruppen intensiv untersucht. Erste Erkenntnisse zur automatisierten Prozessplangenerierung sind in Fischer und Aalami (1996) enthalten. Dabei wurden Prozessmuster entworfen, um das Wissen um Bauprozesse hierarchisch darzustellen. Tiefer gehende Untersuchungen wurden in diesem Zusammenhang von Tauscher (2011) vorgestellt. Hier wurden Ähnlichkeiten in Prozessen identifiziert, um in einem wissensbasiertes System Prozesse mit Informationen aus vorangegangenen Prozessen zu bereichern. Dabei werden sowohl geometrische als auch materialspezifische Ähnlichkeiten berücksichtigt.

Bauprozesse korrelieren nicht zwingend mit Bauteilen. Beispielsweise werden mehrere Bauteile in einem Prozess kombiniert. Dies muss bei einer graphbasierten Betrachtung von Prozessen entsprechend beachtet werden. Kritischer jedoch sind große Bauteile, die in mehreren Arbeitsschritten erstellt werden. Um diese Bauteile in Prozessplänen im Kontext der Bauablaufüberwachung sinnvoll zu erfassen, wurde von Huhnt und Enge (2007) sowie Enge (2010) eine Zusammenfassung bzw. Zerlegung einzelner Objekte in passende Bauablaufobjekte vorgestellt.

Erste Untersuchungen zur automatisierten Neuberechnung von Bauprozessplänen sind in Huhnt et al. (2008) dargelegt. Dabei wurden jedoch fiktive Beobachtungsergebnisse angenommen und keine tatsächliche Bauprozessüberwachung ausgeführt. Außerdem wurde hauptsächlich auf spezielle Änderungsfälle eingegangen. Dazu zählt das Entfernen einzelner Prozesse

und sich dadurch ergebende Auswirkungen. Im Rahmen des hier vorgestellten Ansatzes wird stattdessen primär auf temporale Aspekte, insbesondere auf Verzögerungen eingegangen.

Aktuelle Arbeiten im Bereich der Prozessplanung beschäftigen sich außerdem mit Ähnlichkeiten von Bauprozessen (Sigalov und König, 2015). Auch hier kommen graphbasierte Ansätze zum Einsatz.

3 Theorie

Im Rahmen der hier vorgestellten Methoden sind insbesondere die Graphentheorie als auch das Bauprozessmanagement relevant. Im Folgenden werden daher allgemeine Grundlagen der Graphentheorie diskutiert und ihr Bezug zum entwickelten Ansatz erläutert. Anschließend werden die benötigten Aspekte der Bauprozesstheorien angesprochen.

3.1 Graphentheorie

Grundsätzlich sind im Rahmen der automatisierten Baufortschrittskontrolle zwei unterschiedliche Graphen von besonderer Bedeutung.

Einerseits können technologischen Abhängigkeiten zwischen verschiedenen Bauteilen durch Knoten und Kanten dargestellt werden. Der sich so ergebende Graph wird im Folgenden als Abhängigkeitsgraph bezeichnet. Er ermöglicht, technologisch zwingend vorhandene Bauteile trotz messtechnischer Nichterkennung als "gebaut" festzulegen. Zusätzlich wird der Bauablaufplan durch einen Graphen abgebildet. Er enthält die einzelnen Prozesse, in die sich der gesamte Bauablauf gliedert. In diesem Ablaufgraph werden die zeitliche Abfolge sowie die Zusammenhänge zwischen den einzelnen Prozessen detailliert beschrieben.

Die graphbasierte Repräsentation von technologischen Abhängigkeiten und Bauablaufplan hat zum Vorteil, dass eine kombinierte rechnergestützte Interpretation ermöglicht wird. Sich ergebende Veränderungen im Bauablaufplan können mit Hilfe von Graphersetzungsoperationen in den Ablaufgraph propagiert werden. Letztendlich lässt sich somit eine aktualisierte Version des Bauablaufplans automatisch generieren.

Um die genannten Graphen aufzubauen und zu verändern, wird das Prinzip der Graphersetzung angewandt. Mit Graphersetzungsgesetzen können Teile eines Graphen durch einen neuen Subgraph ersetzt werden. Dabei besteht eine Graphersetzungsgesetz aus einem Mustergraph und einem Ersetzungsgesetz. Wird eine Gesetz auf einen Graphen angewandt, so wird dieser auf das Vorkommen des Mustergraphen hin durchsucht. Existiert ein zum Mustergraphen homomorpher Subgraph, wird dieser durch den in der Gesetz definierten Ersetzungsgesetz ausgetauscht bzw. erweitert. Die einfachsten Formen von Graphersetzungsgesetzen sind Knoten- und Kantenersetzungen (Rozenberg, 1997). Eine Menge von Graphersetzungsgesetzen, die mit einem zugehörigen Metamodell einen Graph aufbauen und verändern kann, wird als Graphersetzungssystem bezeichnet. Derartige Systeme können mit entsprechenden Softwarewerkzeugen implementiert und die Gesetz dadurch automatisiert anwendbar gemacht werden.

Grundsätzlich sind sowohl der Abhängigkeits- als auch der Ablaufgraph gelabelte gerichtete Graphen (Digraphen) $G = \{V, E\}$. Die Kanten $(v, w) \in E$ eines gerichteten Graphen sind dementsprechend gerichtete Kanten. Im Abhängigkeitsgraphen werden einzelne Bauteile als Knoten repräsentiert, während durch die Kanten die Abhängigkeiten der Bauteile untereinander definiert werden (siehe Abbildung 1). Ist ein Bauteil b von einem anderen Bauteil a abhängig, ist diese Abhängigkeit durch eine Kante $e = (a, b)$ dargestellt, die ohne Label auskommt. Die

Knoten a und b repräsentieren dabei entsprechend die Bauteile, wobei die Zuordnung zu einem konkreten Bauteil im Gebäudemodell durch das Label des Knotens gewährleistet wird. Im Ablaufgraph werden die Prozesse durch Knoten repräsentiert (siehe Abbildung 2). Durch das Label des Knotens wird der entsprechende Prozess im Gebäudemodell referenziert. Zusätzlich sind die Bauteile, die während des Prozesses gebaut werden, durch Attribute dieses Knotens festgelegt. Die Kanten zwischen den Prozessen beschreiben einerseits wie die Prozesse aufeinander folgen und legen zusätzlich den zeitlichen Abstand zwischen dem Ende des vorgelagerten Prozesses und dem Beginn des nachfolgenden Prozesses fest. Die Dauer t wird dabei als Attribut der verbindenden Kante gespeichert.

Abbildung 3 visualisiert die im Abhängigkeitsgraphen enthaltenen Elemente.

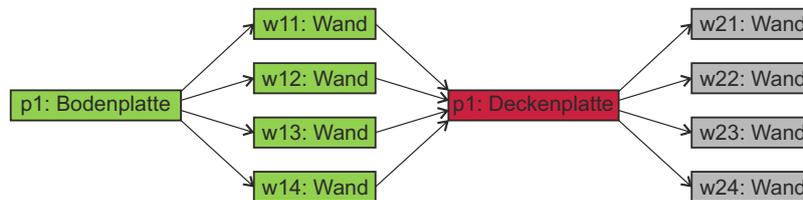


Abb. 1: Graphbasierte Darstellung der technologischen Abhängigkeiten.

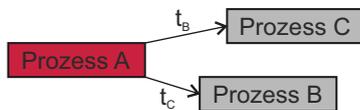


Abb. 2: Graphbasierte Darstellung des Bauablaufs.

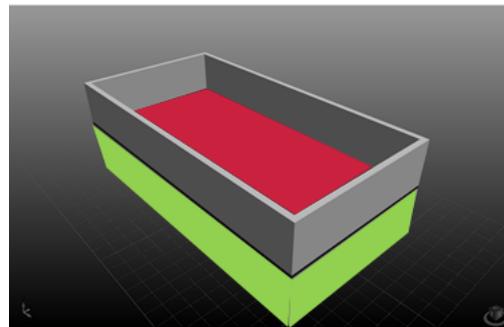


Abb. 3: Visualisierung der in Abbildung 1 enthaltenen Elemente.

3.2 Prozessmanagement

Während der Bauprozessplanung spielen vor allem Ressourcen und Abhängigkeiten eine bedeutende Rolle.

Der Begriff Ressource ist im Baubereich ein Überbegriff für verschiedene Vorgänge beziehungsweise Handlungen. Ressourcen können sowohl Materialien, wie zum Beispiel die Menge an verfügbarem Beton, jedoch auch die Menge an verfügbaren Arbeitskräften sein. Die Verfügbarkeit von Ressourcen ist eine entscheidende Randbedingung in der Prozessplanung. Für die Planer gilt es, einen sinnvollen Kompromiss zwischen verwendeten Ressourcen und minimalem Zeitplan einzugehen. Beispielsweise kann für zeitkritische Aufgaben die Ressource Arbeitskraft erhöht werden, indem man mehr Personal einstellt. Dies verursacht jedoch einen erhöhten Kostenaufwand. Selbiges gilt für die Verfügbarkeit von Schalungswänden auf der Baustelle. Auch hier könnte für eine maximale Auslastung eine hohe Anzahl an Wänden bereitgestellt werden. Die mit der Anschaffung einer hohen Stückzahl einhergehenden Mehrkosten können

allerdings nicht gerechtfertigt werden, wenn abgesehen von Spitzenauslastungszeiten keine Verwendung stattfindet.

Die für die Bauprozessplanung ebenfalls zu beachtenden Abhängigkeiten werden nach Würfel et al. (2007) folgendermaßen kategorisiert:

- *technologisch bedingte, zwingende Abhängigkeiten*
Die wichtigsten und in keinem Fall umgehbaren Abhängigkeiten. Das Dach eines Hauses kann nicht gebaut werden, bevor nicht die Wände darunter fertiggestellt worden sind.
- *vorgegebene, externe Randbedingungen*
Hierzu gehört ein fest vorgeschriebener Fertigstellungstermin oder einzelne Zwischenabnahmen.
- *kapazitätsbedingte Abhängigkeiten*
Zu den kapazitätsbedingten Abhängigkeiten zählen vor allem Ressourceneinschränkungen wie das Fehlen von Baumaterialien und Baugeräten sowie Personalengpässe.
- *terminplantechnische Abhängigkeiten*
In diese Kategorie gehören Pufferzeiten zur Entzerrung des Terminplans und sonstige Ressourcenanpassungen.

4 Prozessneuberechnung

Im Kontext der automatisierten Baufortschrittskontrolle wird die Graphentheorie nun auf den Prozessplan angewendet. Dabei wird ein sogenannter Precedence Relationship Graph (PRG) erstellt. Dieser Graph beinhaltet sämtliche technologischen Abhängigkeiten. Beispielhaft ist dieser Graph in Abbildung 4 für ein Bauprojekt dargestellt. Hierbei handelt es sich, wie in Abschnitt 3 beschrieben, um einen gerichteten, gelabelten Graphen.

Als Datengrundlage für die Neuberechnung wird eine Menge A an Bauteilen zu einem Zeitpunkt t als gebaut klassifiziert. Zur Erfassung des Ist-Zustands und der zugrundeliegenden Methoden sei auf Braun et al. (2015) verwiesen. In diesem Anwendungsfall werden die erkannten Bauteilmengen als gegeben und vollumfänglich betrachtet. Zu beachten ist, dass keine fortlaufende Überwachung stattfindet, sondern die Beobachtungen immer den Bauzustand zu einem gewissen Zeitpunkt widerspiegeln. Dadurch ergibt sich eine Unschärfe zwischen den einzelnen Beobachtungszeitpunkten. Diese Unschärfe beträgt stets $t_i - t_{i-1}$. Im konkreten Anwendungsfall beträgt die Unschärfe zwischen ein und zwei Wochen. Dies bedeutet, dass die Neuberechnungen auch immer nur eine Genauigkeit identisch zur vorhandenen Beobachtungsdichte bzw. Unschärfe haben können.

Im Allgemeinen wurden die folgenden Prozessänderungen identifiziert und untersucht:

- *Temporale Änderungen*
Verzögerungen oder Verfrühungen, welche sich auf nachfolgende Prozesse auswirken.
- *Ähnlichkeitsbasierte Änderungen*
Wiederkehrende Prozesse können basierend auf ihren Eigenschaften (Materialien, Dauer, geometrische Aspekte) detektiert und entsprechend behandelt werden.
- *Temporale, prozessoptimierende Änderungen*
Änderungen, welche den kritischen Pfad beeinflussen und eine Optimierung des Gesamtprozessplanes zulassen.

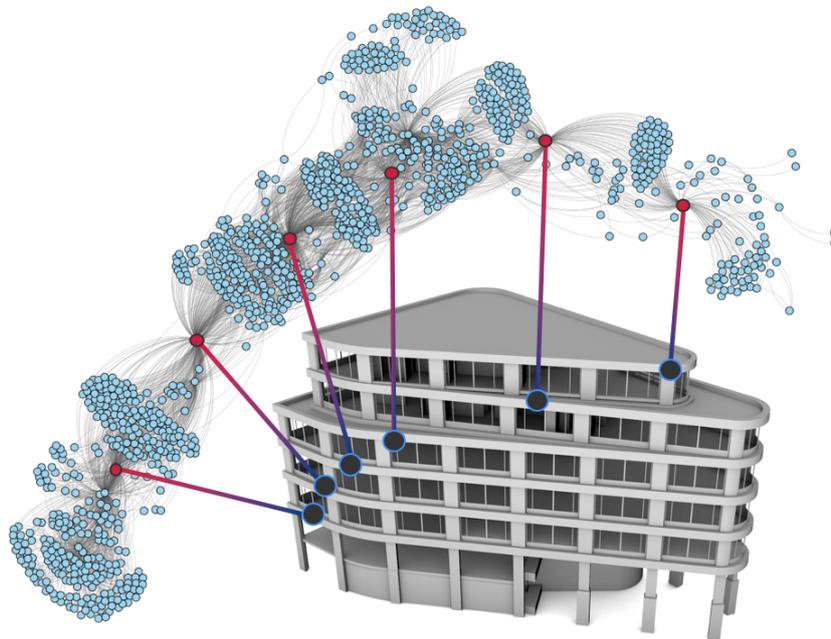


Abb. 4: Precedence Relationship Graph (PRG) am Beispiel des Bauprojekts "Karlstrasse".

Die Bauteile werden, wie in Abschnitt 3 definiert, den jeweiligen Bauprozessen zugeordnet. Diese Menge wird mit der Menge der zum Beobachtungszeitpunkt t als geplant deklarierten Bauteile verglichen. In diesem Schritt wird festgestellt, in welchem Prozessschritt sich die Baustelle gerade befindet. Dabei wird der Abhängigkeitsgraph verwendet und der Endzeitpunkt des letzten fertiggestellten Prozesses gesucht, dem erkannte Bauteile zugeordnet werden konnten. Für den Fall, dass sämtliche Bauteile dieses letzten Prozesses detektiert wurden, wird dieser als aktueller Zustand t_d deklariert.

Die Differenz $\delta t = t - t_d$ symbolisiert nun die Abweichung vom Prozessplan, wobei die Unschärfe u hier nicht berücksichtigt wurde. Für den Fall $\delta t > u$ werden nun mittels des PRGs sämtliche nachfolgenden Prozesse und deren zugehörige Bauteile identifiziert. Diese können nun um δt verschoben werden. Durch diesen Schritt werden temporale Änderungen erfasst und in der Neuberechnung berücksichtigt.

In einem weiteren Schritt können nun einzelne Prozesse detailliert untersucht werden. Dabei wird auf der Makroebene die konkrete Prozessdauer einzelner Prozesse untersucht und deren Dauer analysiert. Ähnliche, wiederkehrende Bauteile bzw. Prozesse können basierend auf ihren Materialien und geometrischen Dimensionen identifiziert werden. Für den Fall einer Änderung der Prozessdauer kann so die Dauer der nachfolgenden, ähnlichen Prozesse ebenfalls analog verlängert werden. Dies sollte jedoch in einem interaktiven Prozess dem Prozessplaner vorgeschlagen werden und nur bei dauerhaft zu erwartenden Verzögerungen angewendet werden. Ein konkreter Fall wäre eine bisher nicht berücksichtigte, wiederkehrende Lieferzeitverlängerung von Materialien. Bei einem einmaligen Auftreten dieser Verzögerung sollte die Neuberechnung nicht ausgeführt werden.

Als Ausblick kann mittels der vorhandenen Daten der Bauprozessplan auch auf ein erneutes Optimierungspotential überprüft werden. Dies ist jedoch nicht im Fokus dieses Forschungsansatzes und wird daher nicht weiter behandelt. Stattdessen wird auf die umfangreiche Literatur in diesem Bereich verwiesen.

5 Case Study

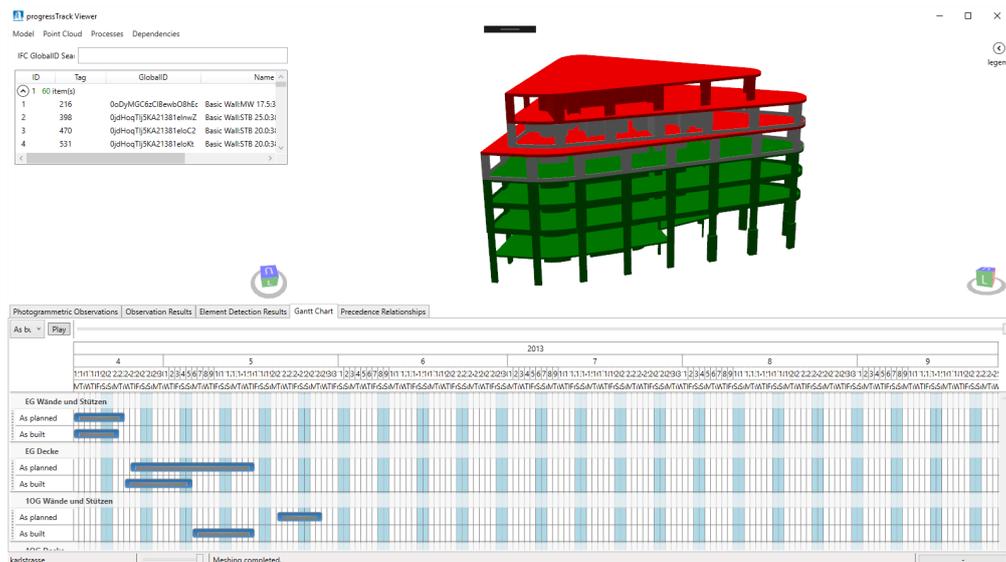


Abb. 5: As-planned vs. as-built Abgleich am Beispiel eines Projektgebäudes. Grün markierte Bauteile wurden vor dem ursprünglichen Zeitplan, rot markierte Bauteile nach dem ursprünglichen Zeitplan gebaut.

Wie bereits erläutert, können im Rahmen der automatisierten Baufortschrittskontrolle Aussagen über das Vorhandensein einzelner Bauteile getroffen werden. Diesbezüglich wurde in drei Fallstudien die generelle Erfassung des Ist-Zustandes detailliert betrachtet. Basierend auf den Ergebnissen sollen nun Prozessänderungen nachverfolgt und neu berechnet werden.

Wie in Abbildung 5 sichtbar, wurde ein Bauprojekt während seiner Bauphase überwacht und die Prozessdauer nach einem Soll-Ist-Abgleich ausgewertet. Dabei wurden die Bauteile nach der Differenz der tatsächlichen Fertigstellung zur geplanten Fertigstellung δt kategorisiert. Bauteile, die früher als geplant fertiggestellt wurden, sind *grün* markiert. Bauteile, die zu spät fertiggestellt wurden, sind *rot* markiert. Der zugehörige Bauprozessplan wurde entsprechend der Erkennungsdaten neu berechnet. Der daraus erzeugte Bauprozessplan ist in Abbildung 6 zu sehen.

6 Zusammenfassung

In diesem Beitrag wurde eine Methodik zur automatisierten Prozessneugenerierung vorgestellt. Aus der Baufortschrittsüberwachung gewonnene Daten wurden mit dem vorhandenen Bauprozessplan abgeglichen. Im Falle von Abweichungen wurde eine Neugenerierung angestoßen.

Automatisierte Prozessneugenerierung basierend auf Ergebnissen aus der Baufortschrittskontrolle

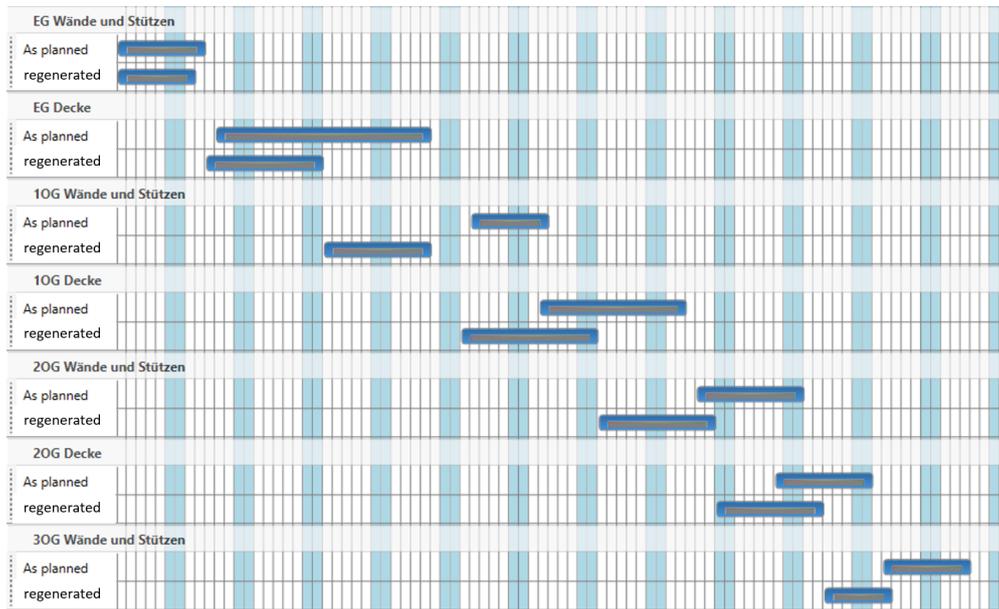


Abb. 6: Neuberechneter Prozessplan basierend auf den erkannten Bauteilen durch die Baufortschrittsüberwachung.

Dabei sind insbesondere technologische Abhängigkeiten zu beachten, um die Integrität des Prozessplanes zu gewährleisten. Diese Abhängigkeiten werden in der vorgeschlagenen Methode durch Graphen repräsentiert und können so über entsprechende Graphanfragen automatisiert ausgewertet werden. Die benötigten Anfragen wurden in einem Graphersetzungssystem implementiert.

Das Verfahren wurde an drei Bauprojekten im Rahmen von Case Studies auf seine Robustheit hin überprüft. Potential zur Weiterentwicklung besteht aus Sicht der Autoren insbesondere in der nachträglichen Optimierung der Bauprozesse. Bei der Neuberechnung wird aktuell nicht geprüft, ob es sich bei dem entstehenden Prozessplan um den optimalen Ablauf handelt oder ob sich durch einen Austausch von Prozessen (sog. Swaps) ein verbesserter oder sogar optimaler Ablauf generieren lässt.

7 Danksagungen

Die präsentierten Forschungsarbeiten werden durch die Deutsche Forschungsgemeinschaft (DFG) unterstützt (STI 545/6 und BO 3575/4-1). Wir möchten uns bei folgenden Unternehmen und Organisationen für die freundliche Kooperation während der Case Studies bedanken: Leitner GmbH & Co Bauunternehmung KG, Kuehn Malvezzi Architekten, Staatliches Bauamt München, Baugesellschaft Brunner + Co., Baukran Logistik GmbH, Baureferat H5 der Landeshauptstadt München, Baugesellschaft Mickan mbH & Co KG, h4a Architekten, Wenzel + Wenzel, Stadtvermessungsamt München. Des Weiteren danken wir dem Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften für die Unterstützung und Bereitstellung der Recheninfrastruktur, die für diese Publikation wesentlich war.

Literatur

- Bosché, F., Turkan, Y., Haas, C. T. und Haas, R. (2010). „Fusing 4D Modelling and Laser Scanning for Construction Schedule Control“. In: *26th ARCOM Annual Conference and Annual General Meeting*. ISBN: 0955239044.
- Braun, A., Tuttas, S., Borrmann, A. und Stilla, U. (2015). „A concept for automated construction progress monitoring using BIM-based geometric constraints and photogrammetric point clouds“. In: *ITcon 20*, S. 68–79.
- Enge, F. (2010). „Muster in Prozessen der Bauablaufplanung“. Diss., S. 210. ISBN: 9783832290535.
- Fischer, M. und Aalami, F. (1996). „Scheduling with computer-interpretable construction method models“. In: *Journal of Construction Engineering and Management* December, S. 337–347.
- Golparvar-fard, M., Pena-Mora, F. und Savarese, S. (2009). „D4AR - a 4 dimensional augmented reality model for automation construction progress monitoring data collection, processing and communication“. In: *Journal of Information Technology in Construction* 14.June, S. 129–153.
- Huhnt, W. und Enge, F. (2007). „Consistent information management for structuring construction activities“. In: *Bringing ITC Knowledge to Work, Proceedings of...*
- Huhnt, W., Richter, S. und Enge, F. (2008). „Modification Management for Construction Processes“. In: *Tsinghua Science and Technology* 13.SUPPL. 1, S. 185–191. ISSN: 10070214. DOI: 10.1016/S1007-0214(08)70147-0.
- Kim, H., Anderson, K., Lee, S. und Hildreth, J. (2013). „Generating construction schedules through automatic data extraction using open BIM (building information modeling) technology“. In: *Automation in Construction* 35, S. 285–295. ISSN: 09265805. DOI: 10.1016/j.autcon.2013.05.020. <http://linkinghub.elsevier.com/retrieve/pii/S0926580513000873>.
- Omar, T. und Nehdi, M. L. (2016). „Data acquisition technologies for construction progress tracking“. In: *Automation in Construction*. ISSN: 09265805. DOI: 10.1016/j.autcon.2016.06.016.
- Rozenberg, G. (1997). *Handbook of Graph Grammars aand Computing by Graph Transformation*. Bd. 1. World Scientific. ISBN: 9810228848. DOI: 10.1142/9789812384720.
- Sigalov, K. und König, M. (2015). „Similarity estimation of BIM-based schedules“. In: *Proceedings of the EG-ICE Workshop on Intelligent Computing in Engineering*, S. 60–69. ISBN: 9780000000002.
- Tauscher, E. (2011). „Vom Bauwerksinformationsmodell zur Terminplanung - Ein Modell zur Generierung von Bauablaufplänen“. Diss.
- Turkan, Y., Bosché, F., Haas, C. T. und Haas, R. (2013). „Toward Automated Earned Value Tracking Using 3D Imaging Tools“. en. In: *Journal of Construction Engineering and Management* 139.4, S. 423–433. ISSN: 0733-9364. DOI: 10.1061/(ASCE)CO.1943-7862.0000629.
- Würfele, F., Bielefeld, B. und Gralla, M. (2007). *Bauobjektüberwachung - Kosten - Qualitäten - Termine - Organisation - Leistungsinhalt - Rechtsgrundlagen - Haftung - Vergütung*. Wiesbaden: Vieweg. ISBN: 978-3-8348-0055-8. DOI: 10.1007/978-3-8348-9402-1.

BIM-basierte Kollaborationsplattform mit objektorientierter Versionierungshistorie und Produktdatenkatalog zur Integration energetisch aktiver Fassadenelemente in Bauprozesse

Anna Wagner¹, Steffen Franz² und Christian Leifgen³

*Institut für Numerische Methoden und Informatik im Bauwesen, TU Darmstadt,
Franziska-Braun-Str. 7, Deutschland*

¹E-Mail: wagner@iib.tu-darmstadt.de, ²E-Mail: franz@iib.tu-darmstadt.de,

³E-Mail: leifgen@iib.tu-darmstadt.de

Abstract In diesem Bericht werden Methoden vorgestellt, die eine ganzheitliche Integration energetisch aktiver Fassadenelemente vereinfachen sollen. Dafür werden eine digitale Produktdatenmodellierung vorgeschlagen, eine Kollaborationsplattform konzipiert und eine objektorientierte Versionierungshistorie entwickelt. Für die Produktdatenmodellierung ist der IFC-Standard entsprechend seines Schemas so erweitert worden, dass allgemein multifunktionale Bauteile damit beschrieben werden können. Für die Kollaborationsplattform wurden Anforderungen gesammelt und ein Konzept mit drei Ebenen (Layern) erarbeitet. Hierfür wurden verschiedene Funktionen wie gemeinsame Datenhaltung, Produktdatenkatalog und Kommunikationsmanagement vorgesehen. Die verwendeten Programme im gesamten, verteilten System können autonom über ein Multi-Agenten-System (MAS) kommunizieren und Aufgaben bearbeiten, um den Planer zu unterstützen. Das MAS erzeugt außerdem eine objektorientierte Versionierungshistorie, die sicherstellen soll, dass der Informationsverlust im Laufe des Planungsprozesses minimal ist. Dies wird dadurch erreicht, dass für neu hinzugefügte Modellkomponenten Vorgängerelemente gesucht und verlinkt werden.

Keywords: BIM, MAS, IFC, Plattformentwicklung, Versionierung

1 Einleitung

Durch die Einführung des Erneuerbare-Energien-Gesetzes (EEG) im Jahr 2000 wurde von der deutschen Bundesregierung der Grundstein zur Erweiterung der regenerativen Energien gelegt (EEG, 2014). Die in Deutschland hauptsächlich zur Stromerzeugung verwendeten Energiequellen sind Windkraft und Sonnenenergie (Henning und Palzer, 2015). Da diese Energien große Flächen für ihre Konversion in Strom benötigen, wären die geforderten Ziele mit weiteren Einschnitten in die Umwelt, z.B. durch Waldrodung, verbunden. Zur Vermeidung dessen könnten bereits existente Flächen auf Dächern oder Fassaden zur Verarbeitung von Solarenergie in nutzbare Energie genutzt werden.

Die Integration energetisch aktiver Fassadenelemente in Bauprozesse ist derzeit jedoch komplex und arbeitsintensiv. So können diverse Fachplaner mit ihren Änderungen die Effizienz des Systems beeinflussen, weswegen eine enge Kollaboration notwendig ist. Hierfür müssen sowohl Kommunikation als auch Datenaustausch koordiniert, homogenisiert und archiviert werden. Zudem müssen die Systeme frühzeitig dimensioniert und geplant werden. Dies verlangt eine konsistente Datenhaltung ohne Informationsverluste.

Um den Integrationsprozess zu unterstützen, wird im Zuge des Forschungsprojektes *SolCon-Pro* (SolConPro, 2016) eine Kollaborationsplattform entwickelt und in dieser Arbeit vorgestellt. Als Basis der Plattform dienen Methoden des *Building Information Modelling* (BIM), die eine kooperative Planung von Gebäuden erleichtern. Einerseits sollen die *Industry Foundation Classes* (IFC) als Austauschformat digitaler Gebäudemodelle zwischen den Projektpartnern über die Kollaborationsplattform verwendet werden. Weil die IFC aktuell keine aktiven Bauteile darstellen können, wird eine Erweiterung des Standards vorgeschlagen. Andererseits wird ein *Multi-Agenten-System* (MAS) entwickelt, welches Kommunikation und Datenaustausch zwischen den Projektpartnern koordinieren soll. Darüber hinaus soll das MAS die Datenhaltung optimieren und Datenverluste durch das Erzeugen einer objektorientierten Versionierungshistorie (OVH) verhindern.

Im nächsten Kapitel werden verwandte Arbeiten vorgestellt, das darauf folgende Kapitel präsentiert die Produktdatenmodellierung, die mit einer Erweiterung des IFC-Standards realisiert wurde. Anschließend wird ein Überblick der Architektur der Kollaborationsplattform und des MAS gegeben, woraufhin die objektorientierte Versionierungshistorie erklärt wird. Abschließend findet eine Diskussion mit Ausblick über die entwickelten Konzepte statt.

2 Verwandte Arbeiten

2.1 Energetisch Aktive Fassadenelemente (EAF)

Fassadenelemente, die als energetisch aktiv bezeichnet werden, dienen zusätzlich zu ihrer Funktion als Gebäudehülle dem Zweck Sonnenenergie in nutzbare Energieformen wie Wärme und Strom umzuwandeln. Dementsprechend teilen sich die Elemente in *Building Integrated Photovoltaic* (BIPV) und *Building Integrated Solar Thermal* (BIST) auf. Die Techniken können zudem zu *Photovoltaic-Thermal* (PVT) kombiniert werden.

Aufgrund ihrer abweichenden Funktionsweisen müssen BIPV und BIST unterschiedlich in die Bauprozesse eingebunden werden. Zwar sind z.B. BIPV Elemente nur wenige Überschneidungen verschiedener Fachrollen vorhanden. Diese reagieren dafür empfindlich auf Verschattungen, sodass bei Veränderungen in der Gebäudehülle stets geprüft werden muss, ob die Änderungen einen Einfluss auf die Effizienz des Systems haben. Im Vergleich sind BIST Module gegen Verschattung und Umwelteinflüsse robuster, ihre Systeme hingegen sind komplexer und besitzen eine höhere Interdisziplinarität mit einem Schwerpunkt auf der technischen Gebäudeausrüstung (TGA).

2.2 Industry Foundation Classes (IFC)

Die IFC sind ein im Bauwesen und Facility Management verwendeter Standard zur digitalen Beschreibung von Gebäudemodellen und dienen hauptsächlich als Austauschformat. Entsprechend können sie von gängigen Modellierungsprogrammen im- und exportiert werden. Neben

der Geometrie werden zusätzlich semantische Informationen der Elemente gespeichert; sollten benötigte Beschreibungen im Standard fehlen, ist es möglich Objekten weitere Eigenschaften hinzuzufügen. Die IFC werden von der BuildingSmart Foundation weiterentwickelt; die aktuellste Version ist IFC4 Add2, die die wesentlichen Objekte eines Gebäudes, aber keine Verkehrsflächen oder Gelände abbilden kann (International Ltd., 2016b).

2.3 Gebäudemodell-Versionierung

Die Versionierung eines Gebäudemodells ist seit der Einführung der BIM-Methode ein bekanntes Problem. Sie ist nicht nur fachlich, sondern auch rechtlich von Bedeutsamkeit, da es mit ihrer Hilfe möglich wird klar festzustellen, wer zu welchem Zeitpunkt welche Veränderung am Modell getätigt hat.

Um eine Versionierung zu erreichen, darf das Modell im Zuge einer Aktualisierung nicht überschrieben werden. Stattdessen wird es als veraltet markiert und mit einem Zeitstempel versehen abgelegt. Während einem direkten Vergleich zweier aufeinanderfolgenden Versionen können so Veränderungen identifiziert werden. Es gibt bereits Software-Lösungen, die automatisiert Abweichungen zwischen Versionen detektieren (Schapke et al., 2015).

2.4 BIMserver

Bei dem BIMserver handelt es sich um eine Java-basierte Open-Source-Software, die eine vollständig veränderbare, Web-basierte Serverumgebung zum Speichern und Austauschen von Gebäudemodellen erzeugt. Als Basis der Datenhaltung dienen die IFC; sie werden unverändert abgelegt und gespeichert. Der BIMserver ist mit weiteren Datenformaten wie CityGML, BCF oder COBie kompatibel. Er besitzt einige Funktionen wie z.B. Versionierung, Zusammenführung verschiedener Modelle oder Kollisionsprüfung. Diese Services können vom Entwickler zusätzlich angepasst und weiterentwickelt werden (BIMserver, 2016).

2.5 Mefisto

Mefisto war ein Forschungsprojekt, das zwischen 2009 und 2012 lief. Ziel des Projektes war ein Management- und Informationssystem zu erzeugen, das eine Prozess-basierte und Risiko-kontrollierte Zusammenarbeit in Bauplanung und -management ermöglicht. Es wurde im Rahmen des "Horizon 2020" Programms der deutschen Regierung gefördert (Mefisto, 2016).

Als Ansatz des Projektes diente die Annahme, dass es ein zentrales Gebäudemodell gibt, in dem Informationen aus allen beteiligten Fachrollen hinterlegt sind, Fachplaner jedoch weiterhin an eigenen Fachmodellen arbeiten. Dies wurde mit den Konzepten von Multi-Modellen, einer gemeinsamen Projektplattform und einer Informationslogik umgesetzt. Als Multi-Modelle verstehen sich Ordner, in denen die verschiedenen Fachmodelle hinterlegt und miteinander verknüpft werden. Die Projektplattform wurde mit einem Webservice-Interface ausgestattet und dient dem Austausch der Multi-Modelle. Die Kollaborationsprozesse werden über eine Geschäftslogik koordiniert.

2.6 Multi-Agenten-Systeme (MAS)

Ein MAS ist definiert als ein System, welches aus einer Mehrzahl von Agenten besteht, die in einem gemeinsamen Umfeld existieren und miteinander interagieren, um Probleme zu lösen. Agenten können in diesem Zusammenhang sowohl menschlich als auch Software-Agenten sein. Laut Shehory und Sturm (2014) sind Software-Agenten als Software-Einheiten zu verstehen, die aufgrund von Anfragen oder Informationsgewinn durch andere Agenten beginnen Aufgaben zu bearbeiten. Sie müssen in einer Umgebung laufen, die gewöhnlich als Plattform realisiert ist. Diese Plattform kann wiederum mehrere sogenannte *Container* beinhalten, in denen die Software-Agenten selbst platziert sind.

Software-Agenten können verschiedene Eigenschaften besitzen. Die für das Konzept des MAS wichtigsten werden an dieser Stelle kurz aufgelistet:

- Soziabilität
- Intelligenz
- Kommunikationsfähigkeit
- Mobilität
- Autonomie

Wenn ein Software-Agent diese Eigenschaften besitzt, ist es ihm u.a. möglich mit anderen Agenten zu interagieren und sich zwischen Plattformen bzw. Containern zu bewegen. Außerdem kann er eigenständig nach Lösungen für das gegebene Problem suchen und sein Umfeld beobachten, wodurch er Informationen und Wissen generieren kann. Zur Ausprägung der genannten Verhaltensweisen kann er mit anderen Software-Agenten kommunizieren, was durch ein allgemeingültiges Protokoll definiert sein muss. Ein solches Protokoll wurde von der *Foundation for Intelligent Physical Agents* (FIPA) als Standard entwickelt.

FIPA-Standards

Die FIPA wurde 1996 mit dem Ziel gegründet Standards für die Architektur von Software-Agenten und MAS zu generieren, damit eine hohe Interoperabilität heterogener MAS Frameworks entstehen kann. Ein Standard ist der der *Agent Communication Language* (ACL) (FIPA, 2002). In ihm wird festgelegt, wie Nachrichten, die zwischen Software-Agenten ausgetauscht werden sollen, aufgebaut und verschickt werden müssen.

Zudem ist es in dem Standard möglich, direkt auf vorangegangene Nachrichten zu antworten, was zur Folge hat, dass komplexe und lange Unterhaltungen zwischen Software-Agenten realisierbar sind. So können diese besser miteinander interagieren oder verhandeln.

2.7 SolConPro

Die vorgestellte Arbeit ist Teil des Forschungsvorhabens "Ganzheitliche Integration energetisch aktiver Fassadenelemente in Bauprozesse" (SolConPro). Das Vorhaben wird in Kooperation von der Technischen Universität Darmstadt, dem Fraunhofer Institut für Solare Energiesysteme und Ed. Züblin AG bearbeitet und durch das Bundesministerium für Wirtschaft und Energie gefördert.

Im Zuge des Forschungsprojektes werden mehrere vergangene Bauvorhaben analysiert, um wiederkehrende oder problematische Prozesse zu abstrahieren. Des Weiteren wird eine Produktdatenmodellierung für energetisch aktive Fassadenelemente definiert und eine Kollaborationsplattform konzeptioniert.

Es wurden bereits Berichte zu diesem Forschungsvorhaben von Leifgen et al. (2016), Franz et al. (2016) und Wagner und Rüppel (2016) veröffentlicht, die jeweils einen Teilbereich des Projektes beleuchten. Der vorliegende Bericht dient als Überblick des derzeitigen Entwicklungsstands seitens der TU Darmstadt.

3 Digitale Produktdatenmodellierung Energetisch Aktiver Fassadenelemente

Der frühe Einsatz von EAF im Planungsprozess setzt bei der Verwendung moderner Methoden, wie dem BIM, die Existenz einer digitalen, maschinenlesbaren Beschreibung voraus. Viele Produkthersteller bieten daher virtuelle Versionen ihrer Produkte an. Diese sind in den meisten Fällen in einem proprietären Datenformat für eine bestimmte Software entwickelt und nicht allgemeingültig verwendbar. Außerdem steht häufig lediglich die geometrische Repräsentation im Vordergrund. Semantische Informationen, welche zwingend für eine ganzheitliche Betrachtung benötigt werden, könnten bereitgestellt werden, das Fehlen klarer Vorgaben für Attributsbezeichnungen sowie deren Umfang erschwert dies jedoch. Weiterhin herrscht unter den Herstellern Unsicherheit über die Detaillierung, mit der sie ihre Produkte digital beschreiben sollen. Eine über-ambitionierte, hochgradig detaillierte Beschreibung führt in der Regel zu einer Überforderung der Planungssoftware. Die genannten Erkenntnisse lassen sich in drei wesentliche Aspekte zusammenfassen: Eine Produktbeschreibung muss für eine anwendungsneutrale Verwendung auf einem offenen Standard basieren. Zusätzlich müssen Attribute allgemeingültig, einheitlich und eindeutig bezeichnet werden. Zuletzt muss der Detaillierungsgrad in Abhängigkeit des Standes der Planung und der damit verbundenen geforderten Informationen wählbar sein.

Den genannten Anforderungen kann bereits begegnet werden, da notwendige Standards und Technologien existieren. In Deutschland ist im Bauwesen die Richtlinie VDI 3805 maßgebend für die herstellerunabhängige Beschreibung von Produkten der TGA. Zur Zeit befindet sich diese in der Überführung in einen internationalen Standard, der ISO 16757, der zur Beschreibung der Daten auf das offene Datenaustauschformat IFC (ISO 16739) zurückgreift. Die DIN 4002 legt Regeln für die anwendungsneutrale Beschreibung von Produktmerkmalen auf Basis internationaler (ISO 13584-42) und europäischer (DIN EN 61360-2, IEC 61360-1/2) Standards fest. Durch das EU-Forschungsprojekt ISES (ISES, 2011) wird deren Einsatzmöglichkeit für die Erstellung einer Produktdatenbank aufgezeigt. Neben den technischen Rahmenbedingungen ist eine eindeutige, sprachunabhängige Bezeichnung der Eigenschaften essentiell. BuildingSMART stellt hierfür mit dem *buildingSMART Data Dictionary* (bSDD) (International Ltd., 2016a) eine auf der ISO 12006-3 basierende Ontologie zur Verfügung. Eine Detaillierung in Abhängigkeit des Planungsfortschrittes kann mit Hilfe des *Level of Development* (LOD), wie es im BIM-Leitfaden für Deutschland (orientiert an den Vorgaben des Amerikanischen Architektenverbands) beschrieben wird, erreicht werden.

Im Rahmen des Forschungsprojektes *SolConPro* finden genannte Standards, Normen und Vorgaben bei der Entwicklung der digitalen Beschreibung der Produkte Anwendung. So wird die technische, semantische und maschinenlesbare Beschreibung mit Hilfe der IFC realisiert.

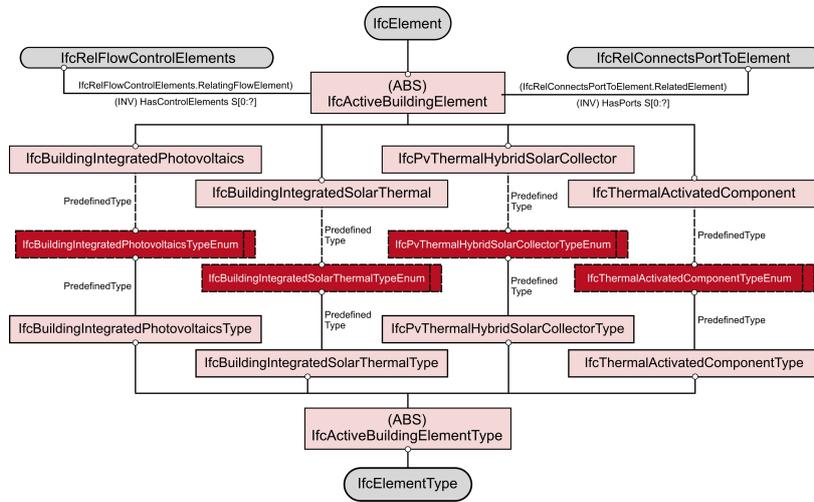


Abb. 1: Vorgeschlagene IFC-Schema Erweiterung zur Abbildung energetisch aktiver Bauteile

Eingangs wurde die Mehrfachfunktion der EAF angesprochen. Da sie sowohl Energie umwandeln und somit Teil der TGA sind als auch als Bauteil und Element der Gebäudehülle statische und ästhetische Funktionen übernehmen, ist eine klare und korrekte semantische Einordnung in die bestehende IFC-Struktur nicht möglich. Das Problem trifft zudem für bspw. energetisch aktivierbare Bauteile wie Kühldecken zu, weswegen eine Erweiterung des bestehenden IFC-Schemas vorgeschlagen wird, sodass diese modernen Elemente abgedeckt werden können. Abbildung 1 ist der entsprechende Vorschlag zur Einführung der abstrakten Entität *IfcActiveBuildingElement* zu entnehmen. Erbende Entitäten sorgen für die semantisch korrekte Wiedergabe. Die Erweiterung orientiert sich an den Konzepten der IFC-Architektur, indem das Typisierungskonzept gemäß korrespondierender Typ-Entitäten umgesetzt ist.

Im nächsten Schritt werden die benötigten Enumerationen in Abhängigkeit der relevanten Technologien erstellt und Merkmale in Abstimmung mit dem bSDD definiert. Das aufgezeigte Vorgehen und die vorgeschlagene Erweiterung des IFC-Schemas ermöglichen eine standardisierte, offene und anwendungsneutrale Beschreibung der EAF. Dadurch wird die frühzeitige Integration der Elemente in die Planung sowie Simulation vereinfacht und die Grundlage für eine ganzheitliche Betrachtung der Komponenten geschaffen.

4 Konzept der entwickelten Kollaborationsplattform

Aufbauend auf den zuvor genannten Überlegungen zur Integration von EAF in Bauprozesse wird in diesem Abschnitt das Konzept einer verteilten Softwareumgebung vorgestellt, die eine kollaborative Zusammenarbeit aller am Projekt Beteiligten unterstützt. Dies soll dadurch erreicht werden, dass Schnittstellen geschaffen und komplexe Abläufe (teil-)automatisiert durchgeführt werden können. Ein Hauptaugenmerk liegt auf der Einbeziehung der im vorherigen Abschnitt beschriebenen Produktdaten und dem im nachfolgenden Kapitel erläuterten Multi-Agenten-System. Um die Herangehensweise an die Entwicklung der Plattform und deren Aufbau zu verdeutlichen, werden verschiedene Sichtweisen und Anforderungen aufgezeigt. Einen Gesamtüberblick der im Folgenden erläuterte Plattform bietet Abbildung 3.

4.1 Inhaltliche Sicht

Grundlage für die Entwicklung der Kollaborationsplattform ist eine Anforderungsanalyse, die eine Übersicht über die einzubeziehenden Inhalte erlaubt. Dabei wurden grundlegende Bereiche ermittelt, die im Rahmen des Forschungsprojektes berücksichtigt werden sollen. Den zentralen Baustein der Plattform bildet ein Gebäudemodellserver, der Koordination und Austausch der Modellstände ermöglichen soll. Zur Abstimmung verschiedener Fachmodelle wird eine persistente und eindeutige Versionierung mittels einer zentralen Verwaltung geschaffen. In einer zweiten Komponente müssen Produktmodelle verwaltet werden können. Darunter fällt neben deren Speicherung auch eine Eingabe von neuen Produkten (bspw. vom Hersteller). Hier stellt das Hinterlegen von Produktdetails (wie z.B. Funktionalbeschreibungen) eine Herausforderung an die Plattform dar. Denn dieses Fachwissen der Hersteller soll nicht öffentlich einsehbar sein, wird jedoch zu einer ganzheitlichen Planung benötigt. Weitere Bereiche, die in der Plattformentwicklung zu berücksichtigen sind, sind die Kommunikation der Beteiligten und die Anbindung von Simulations- und externen Softwaretools. Insbesondere müssen Modellierungssoftwares, wie bspw. Autodesk Revit, einbezogen werden.

4.2 Ablauforientierte Sicht

Ein Ziel des Forschungsprojektes ist es, die Bauprozesskette über den gesamten Lebenszyklus des Gebäudes hinweg zu berücksichtigen. Unter dem Aspekt eine prozessorientierte Herangehensweise zu ermöglichen, wird nicht die Honorarordnung für Architekten und Ingenieure herangezogen, sondern die von der 5D Initiative entwickelte „process map“ genutzt (Initiative, 2016), die das Bauprojekt in fünf Phasen gliedert (s. Abbildung 2). Eng mit der Betrachtung der Projektphasen verbunden und für die Erstellung der Plattform von besonderer Bedeutung ist die vorliegende Informationsdichte. Diese lässt sich durch die verschiedenen LOD beschreiben. Somit kann definiert werden, in welcher Projektphase welche Informationen vorhanden sein müssen.



Abb. 2: Prozessgliederung und Level of Development

4.3 Kooperationsbasierte Sicht

Die nächste Sichtweise auf die Plattform bildet der Aspekt der Kooperation. Um das Projekt zum Erfolg zu bringen, muss die Vielzahl der an einem Bauprojekt Beteiligten in der Lage sein, Modelle, Nachrichten oder allgemeine Informationen einfach und mit Bezug zum jeweiligen Kontext auszutauschen. Die in Deutschland üblichen Geschäftsmodelle und Vertragsstrukturen führen jedoch zu konfrontativem Verhalten der Beteiligten. Daher ist ein Ziel der Plattformentwicklung, Strukturen zu schaffen, die Kommunikation und Kollaboration ermöglichen und fördern. Gleichzeitig soll Beteiligten die Sicherheit gegeben werden, dass andere nur Informationen abgreifen können, die für sie bestimmt und freigegeben sind.

4.4 Programmiertechnische Sicht

Die Entwicklung größerer Softwaresysteme setzt eine geplante und strukturierte Herangehensweise voraus, damit das System langfristig den gestellten und sich stetig ändernden Anforderungen genügen kann. Eine in der Softwareentwicklung übliche Vorgehensweise ist die Trennung der Softwarekomponenten in mehrere logische Ebenen (engl. *layer*). Die Anzahl dieser Layer ist dabei nicht fest, sondern muss abhängig von den Anforderungen im Projekt definiert werden (Microsoft, 2016). Während der Entwicklung sollen weiterhin offene Standards und Dateiformate (z.B. IFC) verwendet werden. Die genutzten Softwaretools müssen entweder mit diesen kompatibel sein oder offene Programmierschnittstellen bereitstellen, mit deren Hilfe relevante Informationen ausgetauscht werden können.

4.5 Entwurf

Der in Abbildung 3 dargestellte Entwurf basiert auf den oben genannten Überlegungen und zeigt die wichtigsten inhaltlichen Komponenten. Die Plattform soll modular aufgebaut und entsprechend der Layer-Architektur in drei Layer der Präsentations-, Steuerungs- und Datenschicht gegliedert werden. Die Gebäude- und Produktdaten, Kommunikationsstrukturen sowie Teile der Simulationen sollen auf einem zentralen Server angesiedelt sein. Andere Bereiche der Plattform werden direkt bei den zuständigen Akteuren vorhanden sein. Basis für Gebäudedaten bildet eine Implementierung des Modellservers BIMserver. Als Weboberfläche ist das erweiterbare Modul "BIMvie.ws" vorhanden. Modelldaten werden IFC-basiert in einer Datenbank gespeichert. Die Verwaltung der Produktdaten bildet eine weitere Säule der Projektplattform. Sie soll ebenfalls IFC-basiert erfolgen und eine zum BIMserver ähnliche Struktur aufweisen: in der Präsentationsschicht eine webbasierte grafische Oberfläche zur Eingabe und Verwaltung der Produkte, in der Steuerungsschicht eine objektorientierte Programmierschnittstelle und eine Datenbank zur persistenten Speicherung der Informationen. Die anderen Komponenten werden ebenfalls der Drei-Layer-Architektur folgend strukturiert, sodass ein modulares und flexibel erweiterbares Gesamtsystem entsteht.

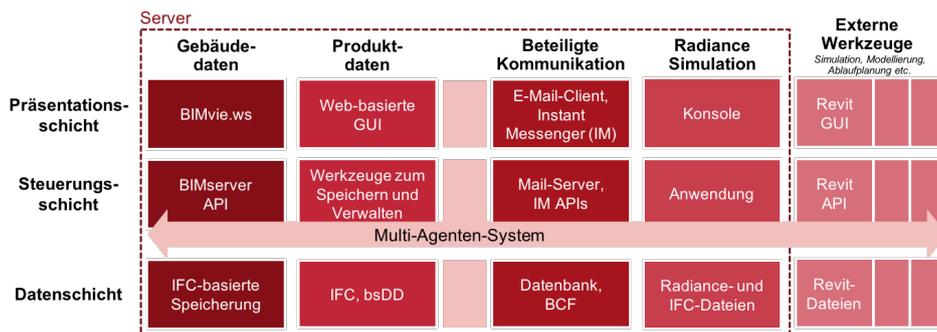


Abb. 3: Plattformübersicht

Zur Verbindung der auf dem Projektserver selbst angesiedelten Komponenten mit externer Software, wird ein Multi-Agenten-System implementiert, das im folgenden Abschnitt weiter erläutert wird.

5 Integriertes Multi-Agenten-System

Die Architektur des verbindenden MAS ist Abbildung 4 zu entnehmen. Auf der Kollaborationsplattform sind zwei Container angesiedelt. Ein Container (*project container*) verbindet sich mit dem BIMserver und kann dadurch auf hinterlegte Informationen zu dem Gebäudemodell zugreifen. In dem anderen (*main container*) befindet sich der Großteil der Agenten inklusive der zugrundeliegenden Logik des MAS. Von dort aus können Agenten auf lokale Produktdatenkataloge und lokal abgelegte Ontologien zugreifen, externe Produktdatenkataloge können mittels Webservice Schnittstellen angesprochen werden. Zudem können die Agenten mit Agenten auf anderen Containern kommunizieren. So wird von jeder Applikation, die ein Projektteilnehmer verwendet ein eigener Container erzeugt, der ihn mit der Hauptplattform verbindet (*container A, B & C*). Voraussetzung hierfür ist, dass angewandte Programme eine Programmierschnittstelle besitzen.

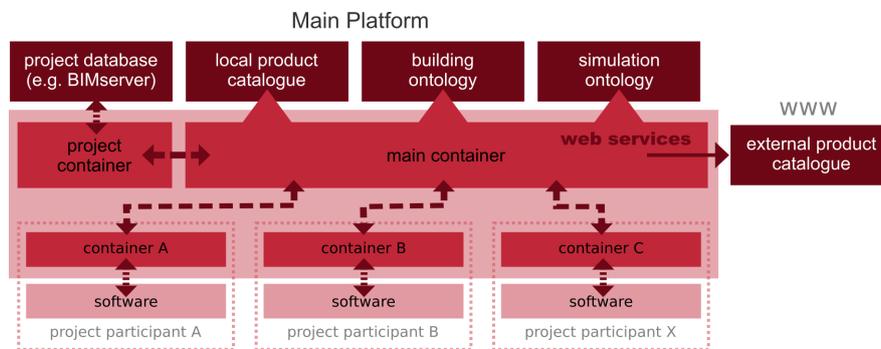


Abb. 4: Architektur des MAS mit Darstellung der Kommunikationswege

Die "Client-Agenten", die auf den von den Anwendern gestarteten Containern leben, beobachten verknüpfte Programme und können Informationen von Agenten auf der Hauptplattform erfragen bzw. die Durchführung von Analysen oder Simulationen anstoßen. Von der Hauptplattform mitgeteilte Ergebnisse können wiederum direkt im Programm den Nutzern angezeigt werden.

Weiterhin gibt es "Element-Agenten", die jeweils ein Bauteil des Gebäudemodells repräsentieren. Mit einer Hierarchie, die von dem *Directory Facilitator* (DF) verwaltet wird, kann mit den Agenten jedes LOD nach Belieben angesprochen werden. Die Element-Agenten sind auf dem Hauptcontainer lokalisiert, sodass sie auf alle Informationen zugreifen können. Laut Cortese et al. (2002) ist außerdem die Performanz des Nachrichtentransports innerhalb eines Containers am besten.

Im Weiteren wird auf die objektorientierte Versionierungshistorie eingegangen, die durch das MAS umgesetzt ist.

5.1 Objektorientierte Versionierungshistorie

Unter Verwendung aktiver Bauteile ist eine gewöhnliche Versionierungshistorie häufig nicht ausreichend, weil ein hoher Informationsverlust zwischen verschiedenen Planungsstufen existiert. Dies ist darin begründet, dass Beziehungen zwischen Elementen unterschiedlicher Versionen

von der globalen Identifikationsnummer, im Fall der IFC *Global Unique Identifier* (GUID) genannt, abhängt. Sollte jedoch ein Bauteil durch ein neues ersetzt werden, wird eine neue GUID erzeugt, wodurch eine Beziehung zwischen den Bauteilen nicht mehr erkannt werden kann. Da bereits zu Beginn der Planung das verwendete System der EAF dimensioniert und festgelegt werden muss, werden solche Eigenschaften oftmals als Textinformation an ein simples Bauteil (z.B. einem Fenster) angehängt. Bei einem Austausch des Fensters zu einem späteren Zeitpunkt gehen somit die bereits formulierten Anforderungen verloren.

Als Grundlage der OVH dienen die "Gelbe Seiten Services" des DF: Dort kann ein Agent beliebig viele Services anmelden, die er öffentlich anbietet und nach denen andere Agenten suchen können. So müssen Agenten von veralteten Bauteilen nicht gelöscht werden, ihre Services müssen lediglich mit dem Zusatz ihrer Versionsnummer versehen werden, wodurch ersichtlich wird, dass das Bauteil einen Nachfolger besitzt. Außerdem sind in den Services die Funktionalitäten des Elementes (z.B. Fassadenelement), ihr(e) Einbaustockwerk(e) und eine grobe Beschreibung des Einbauortes (z.B. Fassade Süd) hinterlegt.

Um einen Informationsverlust zu verhindern, muss in Folge jeder Erzeugung eines Bauteils geprüft werden, ob es in der vorigen Version Bauteile gab, die von ihm ersetzt werden. Das Vorgehen hierfür ist in Abbildung 5 zu erkennen. Entsprechend des erläuterten Ansatzes ändert der DF aufgrund der Mitteilung von F dessen Services. Fragen neue Element-Agenten, wie BIPV, nach Vorgängern, suchen diese in den gelben Seiten des DF unter allen Agenten, die aktuell sind oder der letzten Version entsprechen, nach semantischen Übereinstimmungen in den Funktionalitäten (an mindestens einer Stelle) und dem Einbauort. Anschließend senden sie jedem Agenten der erhaltenen Menge eine Nachricht mit geometrischen und semantischen Daten ihres Bauteils. Stellt einer der Element-Agenten daraufhin fest, dass die Informationen übereinstimmen, teilt er es dem neuen Agenten mit und beide Agenten registrieren bei dem DF ihren gefundenen Vorgänger bzw. Nachfolger als Service.

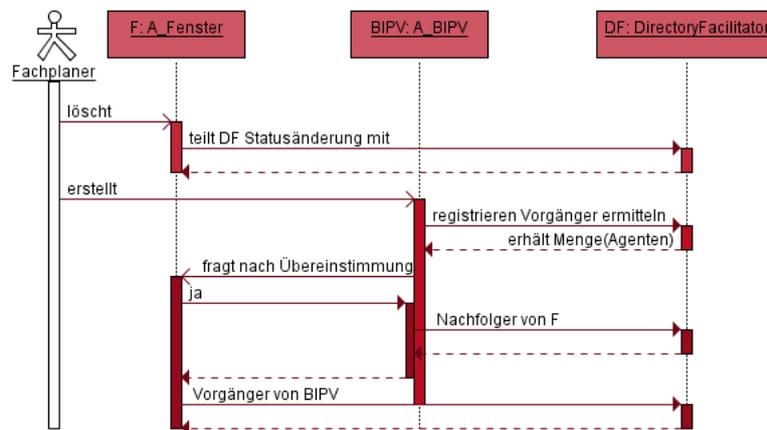


Abb. 5: Sequenzdiagramm zur Registrierung der Vorgänger bzw. Nachfolger

Abbildung 6 zeigt einen weiteren Vorteil, der sich durch die OVH ergibt. In diesem Szenario hat der Fertigungsplaner eine Wand in zwei Betonierabschnitte geteilt. Die Elemente haben, wie in Abbildung 5 dargestellt, ihre Vorgänger bzw. Nachfolger bereits gefunden und als Service dem DF gemeldet. Da der Architekt die Unterteilung der Betonierabschnitte nicht benötigt, wird ihm für seine Planung weiterhin die gesamte Wand als ein einziges Bauteil (W1) angezeigt. Wenn der Architekt nun die Betoneigenschaften der Wand ändert, würde dies ohne die OVH zu dem Problem führen, dass der Fertigungsplaner diese Änderung nicht wahrnehmen würde,

weil dieser nur die Wände W2 und W3 betrachtet. Wegen der OVH teilt der Agent von W1 die Änderungen seinen Nachfolgern mit, sodass der Fachplaner die korrekten Betoneigenschaften erhält.

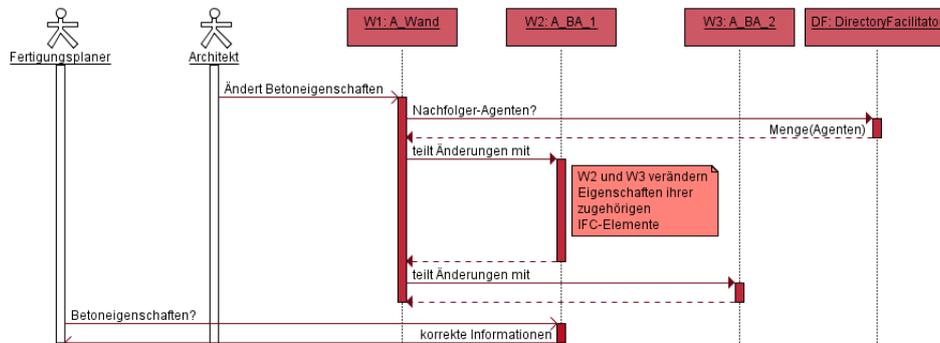


Abb. 6: Sequenzdiagramm zur Veranschaulichung der Vorgehensweise der OVH

6 Diskussion und Ausblick

Die vorgestellte Produktdatenmodellierung ermöglicht eine standardisierte Abbildung EAF. Sie wird von einer Erweiterung des IFC-Standards ermöglicht. Die Erweiterung ist derzeit nicht in die offiziell unterstützte Version der IFC integriert, es werden aber in Absprache mit BuildingSmart Anstrengungen mit diesem Ziel unternommen. Zur Verwendung innerhalb des Forschungsprojektes werden die notwendigen, zusätzlichen Eigenschaften für den Produktdatenkatalog über Hinzufügen von PropertySets realisiert. Nach einer erfolgreichen Umsetzung des Vorschlags in eine offizielle Version des Standards, müssen Normen und Richtlinien formuliert werden, die Hersteller multifunktionaler Bauteile dazu bewegen ihre Produkte in dieser Form digital zur Verfügung zu stellen. Bisher sind solche Richtlinien nicht existent. Die Integration der Produktdatenkataloge in die Projektplattform und deren webbasierte Eingabemaske motiviert Anwender und Hersteller aktiver Bauelemente dazu den vorgeschlagenen Standard zur Beschreibung von EAF zu nutzen.

Die vorgestellte Plattform ermöglicht eine Kollaboration und gemeinschaftliche Datenhaltung. Diese zieht, ebenso wie die objektorientierte Versionierungshistorie, rechtliche Fragen und Probleme mit Gewährleistung und Haftung mit sich, die bei dem traditionellen Austausch von Papierdokumenten nicht vorhanden sind. Seit der Veröffentlichung des "Stufenplan Digitales Planen und Bauen" (Dobrindt, 2015) ist jedoch davon auszugehen, dass in näherer Zukunft Regelungen für das gemeinsame Bearbeiten eines digitalen Dokumentes in Deutschland eingeführt werden.

Weiterhin müssen die im Forschungsprojekt identifizierten Prozesse, die zur Integration aktiver Fassadenelemente notwendig sind, in die Plattform integriert werden. Durch deren Modularität stellt dies kein Problem dar. Weitere Neuerungen in dem Feld der (solar) aktiven Bauteile könnten ebenfalls als Erweiterung der Plattform zugefügt werden. Das ist zusätzlich zu der Plattform-Architektur aufgrund der Verwendung eines MAS möglich, welchem neue Agenten und Agententypen hinzugefügt werden können, ohne einen Neustart des Systems notwendig zu machen. So lassen sich während dem Verlauf eines Projektes Anpassungen umsetzen ohne Informationsverluste riskieren zu müssen.

BIM-basierte Kollaborationsplattform mit objektorientierter Versionierungshistorie und Produktdatenkatalog zur Integration energetisch aktiver Fassadenelemente in Bauprozesse

Das MAS birgt neben seinen Vorteilen auch den Nachteil, dass nicht alle Anwendungsprogramme von der Plattform unterstützt werden können, da eine Programmierschnittstelle eine Voraussetzung für dessen Anbindung ist. Es ist daher zu untersuchen, ob Agenten ohne direkte Einbindung in Software-Applikationen, sondern nur über das Lesen von Dateien, vorgenommene Änderungen überwachen können.

Literatur

- BIMserver (2016). *BIMserver*. <http://www.bimserver.org/> (besucht am 15. 03. 2016).
- Cortese, E., Quarta, F., Vitaglione, G., Lab, T. I., Direzionale, C., Message, J. und System, T. (2002). *Scalability and Performance of JADE Message Transport System*.
- Dobrindt, A. (2015). *Stufenplan Digitales Planen und Bauen*. Techn. Ber. Berlin: Bundesministerium für Verkehr und digitale Infrastruktur. https://www.bmvi.de/SharedDocs/DE/Publikationen/DG/stufenplan-digitales-bauen.pdf?__blob=publicationFile (besucht am 17. 07. 2016).
- EEG (2014). *Erneuerbare Energien Gesetz 2014 Gesetz für den Ausbau erneuerbarer Energien*. <http://www.bmwi.de/BMWi/Redaktion/PDF/G/gesetz-fuer-den-ausbau-erneuerbarer-energien> (besucht am 15. 03. 2016).
- FIPA (2002). *FIPA ACL Message Structure Specification*. Englisch. Standard SC00061G. Geneva, Switzerland: FIPA, S. 11. <http://www.fipa.org/specs/fipa00061/SC00061G.pdf> (besucht am 25. 02. 2016).
- Franz, S., Rüppel, U., Kuhn, T. E. und Teizer, J. (2016). „A Multi-Agent-Based Platform for the Early Integration of Photovoltaic Systems in Building Facades“. In: *ICCCBE*.
- Henning, H.-M. und Palzer, A. (2015). *Was kostet die Energiewende? - Wege zur Transformation des deutschen Energiesystems bis 2050*. de. Techn. Ber. Freiburg: Fraunhofer ISE.
- Initiative, 5. (2016). *5d Initiative*. <http://www.5d-initiative.eu/process.html>.
- International Ltd., buildingSMART (2016a). *buildingSMART Data Dictionary*. <http://bsdd.buildingsmart.org/>.
- International Ltd., buildingSMART (2016b). *IFC Overview*. <http://www.buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary> (besucht am 04. 12. 2016).
- ISES (2011). *Intelligent Services For Energy-Efficient Design and Life Cycle Simulation*. <http://ises.eu-project.info/>.
- Leifgen, C., Kuhn, T. E., Rüppel, U. und Teizer, J. (2016). „BIM-based Collaboration Platform for the Holistic Integration of Energy Active Facade Components“. In: *ICCCBE*.
- Mefisto (2016). *Mefisto*. <http://www.mefisto-bau.de/overview.html>.
- Microsoft, M. (2016). *Three-tier Application Model*. <https://msdn.microsoft.com/en-us/library/aa480455.aspx>.
- Schapke, S.-E., Beetz, J., König, M., Koch, C. und Borrmann, A. (2015). „Kooperative Datenverwaltung“. de. In: *Building Information Modelling*. Wiesbaden: Springer Vieweg, S. 207–236. ISBN: 978-3-658-05605-6.
- Shehory, O. und Sturm, A. (2014). „Chapter 1, A Brief Introduction to Agents“. In: *Agent-oriented software engineering : reflections on architectures, methodologies, languages, and frameworks*. Berlin [u.a.]: Springer. ISBN: 3-642-54431-2. <http://d-nb.info/1047007045/04>.
- SolConPro (2016). *solconpro*. de/en. <http://www.solconpro.de/>.
- Wagner, A. und Rüppel, U. (2016). „Cooperative Design Methods for Energy Active Facade Elements Based on BIMserver and Multi-Agent Services using JADE“. In: *EG-ICE*.

Entwicklung eines Model Generators für private Haushalte auf Basis der IWU-Gebäudetypologie als Grundlage für EnergyPlus Gebäudesimulationen

Christian Eller

*Institut für Numerische Methoden und Informatik im Bauwesen, Franziska-Braun-Straße 7
64287 Darmstadt, Germany
E-Mail: eller@iib.tu-darmstadt.de*

Abstract Durch die europaweite Einführung der Smart-Meter Zähler, ist es möglich individuelle Energieverbrauchsdaten automatisiert messen zu können. Somit stehen in Zukunft detaillierte Energieverbräuche im privaten Haushalt zur Verfügung, die verwendet werden können, um individuelle Energiesimulationen im Gebäudebestand durchführen zu können.

In diesem Zusammenhang beschreibt dieses Paper die Entwicklung einer Software, die es ermöglicht simulationsfähige Modelle von Einfamilienhäusern zu erstellen, um reale Simulationen über Energie- und Heizverbräuche zu tätigen. Die OpenStudio C#-API wurde verwendet, um ein Modell mit Konstruktionen, verschiedenen Räumen, Bewohnern, Heizanlage, elektronischen Geräten und Lichtern zu generieren.

Die Eingabe des Nutzers konnte durch eine große Datenbank auf die Information des Gebäudetyps und Baujahr reduziert werden. Die Kompatibilität mit EnergyPlus wurde hergestellt und die Ergebnisse in OpenStudio veranschaulicht und verglichen. So konnte gezeigt werden, dass es möglich ist, mit der Verwendung der IWU-Gebäudetypologie, einen Generator zu erzeugen, der simulationsfähige Modelle entwickelt, welche aussagekräftige Ergebnisse liefern. Dazu sind vom Nutzer keine Fachkenntnisse in Bezug auf Baukonstruktion, energetische Berechnungen, Softwareprogrammierung oder der Simulationssoftware EnergyPlus notwendig.

Keywords: Smart-Meter, EnergyPlus, OpenStudio, Energiesimulation

Einleitung

Am 1. Juli 2011 beschloss der Bundestag den Atomausstieg Deutschlands und ermöglichte damit die Energiewende. Diese Wende hat Einfluss auf die Energiegewinnung und deren Verbrauch. Der größte Anteil des Energieverbrauchs entfällt auf das Heizen im Gebäudebestand (BMWi, 2014). Deshalb ist es nötig eine Strategie zu entwickeln, um die Energieeffizienz im Gebäudebestand und Neubau zu steigern. Das „Gesetz zur Digitalisierung der Energiewende“ beschreibt die Energiewende als einen erheblichen Umbau der Elektrizitätsversorgung und die Veränderung in der Verbrauchsmessungen (BMWi, 2005).

Dabei ist die Rede von sogenannten Smart-Meter Zählern. Die Nutzung der daraus gewonnenen Daten ist allerdings erst am Anfang der Entwicklung. Die Frage ist, welcher Mehrwert sich durch die Auswertung der Daten für die angesprochenen Parteien und für die Umwelt generieren lassen. Es geht um die Kombination der Verbrauchsdaten mit den Haushalts-Kenndaten, um damit anhand von Modellen, Aufschlüsse über Strom- und Wärmeverbrauch von Gebäuden zu erhalten. Dadurch sollen Potenziale zur Verbesserung und Sanierung im Gebäudebestand leicht erkennbar gemacht und aufgedeckt werden.

Ziel

Ziel dieses Generators ist es, eine einfache, umfassende und realitätsnahe Modellierung eines Einfamilienhauses zu erstellen, welche anhand von gemessenen Smart-Meter Energieverbräuchen den Energiehaushalt des Gebäudes ermitteln kann. Dabei soll das Modell auf den Grundlagen der Daten der Gebäudetypologie des Instituts für Wohnen und Umwelt (=IWU) (IWU, 2005) basieren und mit der Simulationssoftware EnergyPlus kompatibel sein. Grundsätzlich sollen durch den Nutzer nur wenige Angaben über das abzubildende Gebäude eingegeben werden, um auch Laien die Nutzung des Tools ermöglichen zu können und ein simulationsfähiges und realitätsnahes Modell zu generieren.

Die Entwicklung dieses Softwaretools, die Bereitstellung der nötigen Daten, sowie die Auswertung der Simulationsergebnisse ist der Inhalt der weiteren Arbeit.

Grundlagen

IWU-Gebäudetypologie

Als 1993 das IWU durch die Bundesregierung den Auftrag erhielt Energieeinsparpotentiale durch Wärmeschutzmaßnahmen für den deutschen Gebäudebestand zu bestimmen, erstellte diese eine Gebäudetypologie (IWU, 2005). Diese enthält verschiedene Bauklassen aller Jahrzehnte seit 1900, die Daten über die Wärmeleitfähigkeit der angewendeten Konstruktionen, durchschnittlichen Fenster-, Grundriss- und Bauteilflächen und Beheizungsgrad des Kellers und Daches. Somit ermöglicht die Gebäudetypologie neben der Entwicklung von Heizwertberechnung, Energieeinsparpotenzialen und Szenarien, den schnellen Zugriff auf typische Flächen und Wärmeleitfähigkeiten des Gebäudes. Es existiert dadurch eine belastbare Datengrundlage für verschiedene Gebäudestrukturen.

EnergyPlus

EnergyPlus ist eine Software zur energetischen Simulationen ganzheitlicher Bauwerke. Sie betrachtet Wärme-, Kühl-, Licht-, Wassersysteme und prozessbasierte Belastungen eines Gebäudes. Die Software ermöglicht nach (NREL, 2014) zahlreiche Funktionalitäten:

- **Integrierte, parallele Analysen** von unterschiedlichen Thermalen Zonen und Heizungs-, Lüftungs- und Klimatechnik in Räumen unterschiedlicher Eigenschaften und Bedingungen.

- **Berechnung der Wärmebilanz**, welche von Strahlungs- und Konvektionswärme der Oberflächentemperaturen, sowie thermischen Behaglichkeit und Kondensation abhängt.
- **Einstellung frei definierbare Zeitschritte einzelner Komponenten**, um eine hohe Präzision und dynamische Prozesse in der Wechselwirkung von unterschiedlichen Zonen und der Umwelt zu erreichen.
- **Kombinierte Wärme- und Stoffaustauschmodellierung**, welches eine Simulation von Luftbewegung auch zwischen den Zonen ermöglicht.
- **Erweiterte Fenstermodellierung**, die es ermöglicht Jalousien und unterschiedliche Verglasungsaufbauten zu erstellen und somit die Schicht für Schicht absorbierte Solarenergie zu ermitteln.
- **Berechnung der Beleuchtungsstärke und Ablendung**.
- **Komponentenbasierte Heizungs-, Lüftungs- und Klimatechnik (HLK engl. HVAC)**, die sowohl Standard- und neuartige Systemkonfigurationen unterstützt.
- Auswahl aus einer **großen Anzahl an HLKs und Lichtanlagen** mit Erweiterungsmöglichkeit.
- **Funktionales Interface**, um Importe und Exporte zu anderen Engines zu ermöglichen.
- **Ergebniszusammenfassung**, detaillierte und Nutzer definierte Ausgabe der Resultate über die einzelnen Zeitabschnitte.

EnergyPlus bietet die Möglichkeit über das Dateiformat Input Data File (=IDF) Daten einzulesen oder auszugeben. Die Simulationsergebnisse werden über SQL, CSV und HTML Softwareübergreifend ausgegeben, damit sie leicht visualisiert werden können. Diese Formate sind hilfreich, da EnergyPlus keine eigene User Interface besitzt, um Nutzereingaben einzuarbeiten oder Ergebnisse zu veranschaulichen.

OpenStudio

Die fehlende visuelle Oberfläche von EnergyPlus bietet OpenStudio. Die Software wurde als Open Source Projekt vom National Renewable Energy Laboratory (=NREL) entwickelt. Sie verbindet EnergyPlus mit der Lichtanalysesoftware „Radiance“. OpenStudio beinhaltet nicht nur die Möglichkeit EnergyPlus fähige IDF-Dateien zu erstellen, sondern auch in einem Results-Viewer und Parametric Analysis Tool die Ergebnisse zu veranschaulichen und zu analysieren. Die Software wurde dafür entwickelt, Gebäude des öffentlichen Raumes wie z.B. Schulen und Krankenhäuser auf ihren energetischen Verbrauch und daraus resultierenden Kosten zu analysieren. OpenStudio enthält eine Graphische Oberfläche, kann aber auch über seine C#- oder Ruby-API und als SketchUp Plug-In genutzt werden. Die Geometrien können mit Hilfe des CAD-Programms SketchUp hinzugefügt werden. Durch diese Entwicklung kann EnergyPlus benutzerfreundlich genutzt werden.

Ein OpenStudio-Model besteht aus zahlreichen Komponenten, die in verschiedenen Detaillierungsstufen angegeben werden können. Hierfür bietet die Abbildung 1 eine Übersicht. Wichtig für eine funktionierende Simulation sind die Generierung von Räumen, sogenannten Spaces, Errichtung von thermalen Zonen, sowie das Heiz- und Kühlsystem und die Wetterdaten.

OpenStudio folgt einem hierarchischen Aufbau [siehe Abbildung 1]. So besteht das Modell aus einem oder mehreren Construction Sets, welche eine Sammlung aller Konstruktionen wie

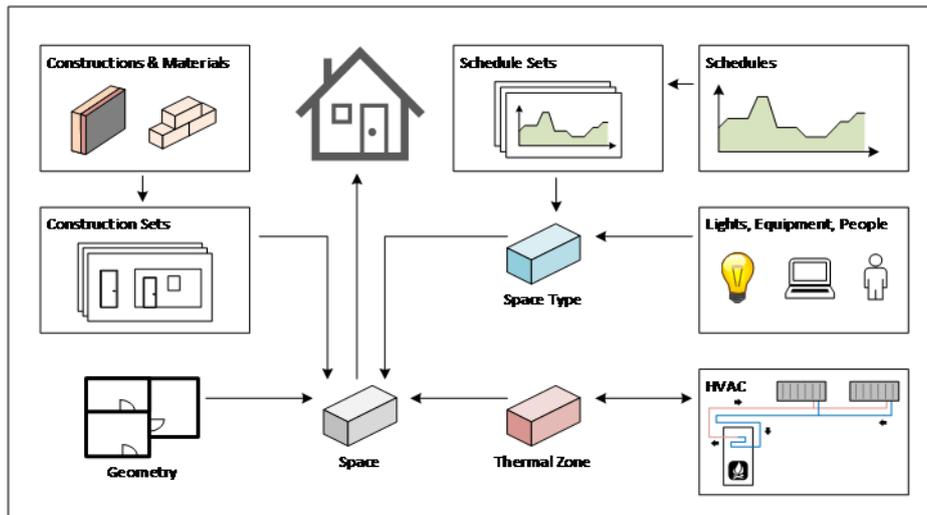


Abb. 1: Vereinfachte Übersicht der OpenStudio Modellentwicklung (Irmeler und Eller, 2016)

Wand, Decke, Türen, Fenster etc. beinhaltet. Dies kann dem Gebäude oder nur einem Teil, wie dem Stockwerk hinterlegt werden. Dadurch kann durch einen einfachen Austausch des Sets die vollständige Bauweise des Gebäudes oder eines Stockwerks geändert werden. Die Konstruktion setzt sich aus unterschiedlichen Schichten zusammen, die als Materialien hinterlegt werden. Diese Sets werden dem Space, einem Raum, hinzugefügt, welcher mit Hilfe der Geometrien das Gebäude abbildet. Ein Space spiegelt eine thermische Zone wieder, welche mit Heiz-, und Kühlkomponenten wie Heizkörpern, einem Heizkühl- und Lüftungssystem hinzugefügt werden kann. Das Heizkühl- und Lüftungssystem besteht aus einzelnen als Kreislauf verknüpften Komponenten wie Heizboiler, Heizpumpe und Rohrleitungen. Durch die Verknüpfung mit den Spaces stehen diese der energetischen Berechnung zur Verfügung. Die Einflüsse auf den Space werden durch die außen und innen liegenden Bedingungen beschrieben. Dies sind unter anderem die Wetterbedingungen, aber auch durch die Nutzer und Verbraucher im Gebäude. Genauer sind das elektronische Verbraucher, wie Licht und Geräte, aber auch Personen, die sich über den Tag darin aufhalten. Alle Verbraucher werden über ihre Anzahl, einem Grundverbrauch und Graphen, die das zeitliche Vorhandensein oder die Höhe des Verbrauchs beschreiben definiert. Diese Zeitabläufe werden als Schedules dem Modell zur Verfügung gestellt und in sogenannten Schedule Sets ähnlich der Construction Sets vereint.

Vorhandene Lösungsansätze

Die Building Science Group versucht aktuell eine webbasierte Applikation namens „Virtual Pulse“ zu erstellen, welche es ermöglichen soll Modelle in OpenStudio zu generieren und diese durch EnergyPlus zu simulieren (building Science Group, 2016). Die Ergebnisse sollen aufbereitet dem Nutzer im Browser angezeigt und zum Download zur Verfügung stehen. Allerdings beschränkt sich die Modellentwicklung auf Gebäude der Standardbibliothek von OpenStudio. Diese beinhaltet zum derzeitigen Stand keine privaten Haushalte sondern öffentliche Gebäude, weshalb dieses Tool für diese Problemstellung nicht genutzt werden kann.

Den Versuch private Haushalte abzubilden unternimmt auch (NREL, 2014) mit ihrer Software Building Energy Optimization (=BEopt). BEopt ist eine Software um Ein- und Mehrfamilien-

häuser durch EnergyPlus auf ihre laufenden Kosten simulieren zu können. Die Entwicklung der Geometrie erfolgt über gesetzte Standards und einfaches Platzieren von variablen Rechtecken. Fenster und Türen werden automatisch hinzugefügt. Ebenso findet eine reiche Vorauswahl an den Nutzungsinformationen statt. Der Standort des Gebäudes, die äußeren Umweltbedingungen und die verwendeten Heizanlagen und Betriebsrohstoffe werden von der Software festgelegt. Diese sind veränderlich, aber nur im Rahmen der vorliegenden Auswahlmöglichkeiten. Diese Auswahl beschränkt sich auf den amerikanischen und kanadischen Raum, wodurch die Abbildung europäischer Gebäude nicht möglich ist. Die Daten über Konstruktionen, deren Eigenschaften, und Nutzungszeiträume im Alltag stammen vom Building America Housing Simulation Protocol, welches vom Office of Energy Efficiency & Renewable Energy erstellt worden ist. Das definierte Gebäude wird durch EnergyPlus simuliert und die Ergebnisse der Kosten angezeigt. Ziel der Software ist es, die geringsten Nutzungskosten verschiedener Gebäudestrukturen zu zeigen und diese Struktur zu optimieren. Eine energetische Analyse ist nicht vorgesehen und die Einflussnahme auf das Modell sehr beschränkt. Deshalb ist es sinnvoll einen eigenen Generator zu entwickeln.

Konzept

Einer der entscheidenden Hauptanforderungen an die Software ist es, keinerlei Voraussetzung in Fachkenntnisse an den Nutzer zu fordern und die Nutzerfreundlichkeit durch geringe Eingaben sicher zu stellen. Eine realitätsnahe Modellierung erfordert wiederum eine Vielzahl an Angaben und der nötigen Fachkenntnisse in den unterschiedlichsten Bereichen eines Gebäudes. Um diese Divergenz aufzulösen, ist die Idee viele Daten in einer Datenbank verfügbar zu machen und im Hintergrund zu nutzen. Diese Daten werden durch unterschiedliche Fachquellen wie unter anderem dem statistischen Bundesamts und dem Bauteilkatalog des Zentrums für Umweltbewusstes Bauen gewonnen [siehe Abbildung 2].

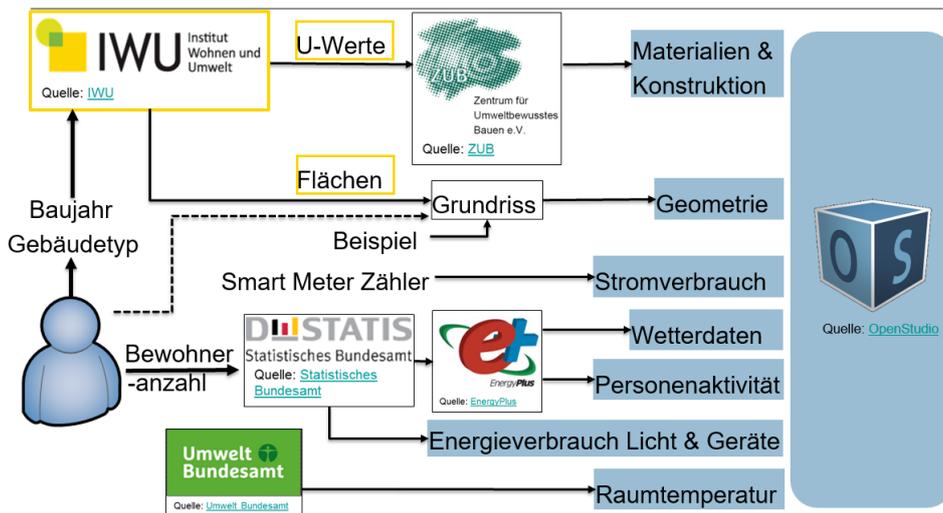


Abb. 2: Konzept zur Minimierung der Nutzereingaben durch Nutzung von Datensätzen

Die vorliegende Konstruktionsstruktur der Gebäude aus der IWU soll in eine Datenbank übertragen werden. Ebenso sollen Daten über elektronische Lastfälle einzelner Haushalte, elektronische Verbräuche und Zeitpläne über den Aufenthalt von Bewohnern in die Datenbank

Entwicklung eines Model Generators für private Haushalte auf Basis der IWU-Gebäudetypologie als Grundlage für EnergyPlus Gebäudesimulationen

eingepflegt werden. Durch diese Datenbank sollen alle Daten zur Laufzeit vorhanden sein. Zusätzlich mit den Eingaben des Nutzers über eine Graphical User Interface (=GUI), die entweder nur die Informationen über Baujahr und Gebäudeklasse abfragt, oder in einer Erweiterung detaillierte Informationen einfordert, wird das Gebäudemodell mit Geometrie, Konstruktionen, Heizungssystem und Nutzungsart individuell erstellt.

Außerdem soll durch das Softwaretool die erweiterte Möglichkeit gegeben sein, einzelne Verbraucher, wie elektronische Endgeräte und Personen des Haushaltes hinzuzufügen. Das Modell soll durch OpenStudio simuliert und dem Nutzer die Ergebnisse visuell angezeigt und im Hintergrund abgespeichert werden [siehe Abbildung 3].

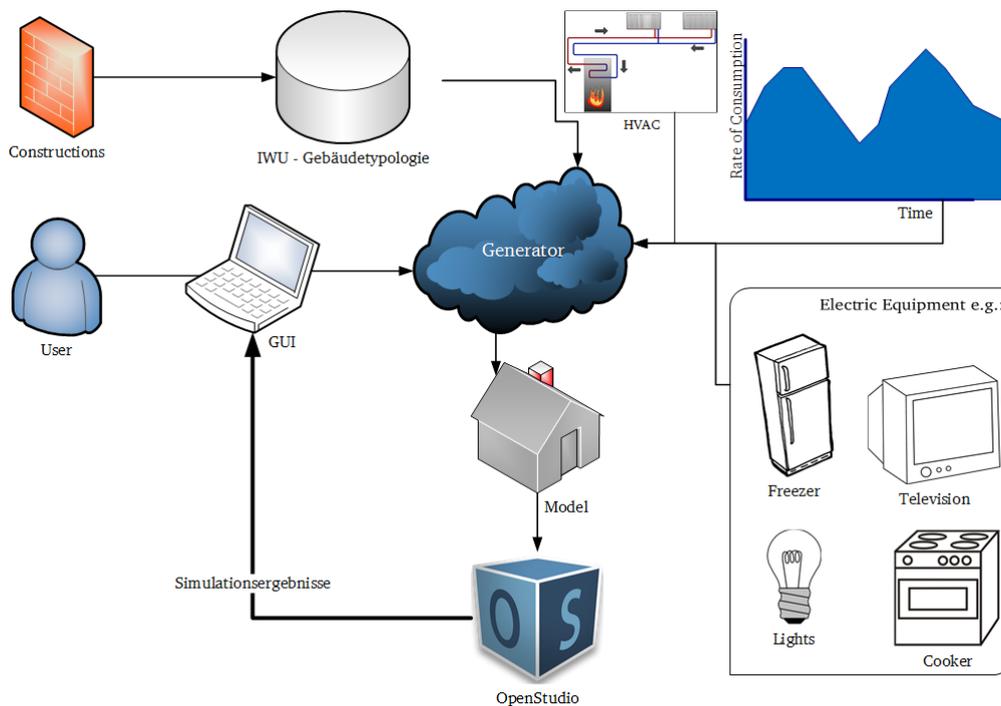


Abb. 3: Konzept zur Generierung des Softwaretools

Umsetzung

Die Umsetzung erfolgte über die Erstellung einer Datenbank [siehe Abbildung 4] und der Entwicklung einer eigenen Software. Diese Software besitzt eine GUI für die Abfrage der Nutzereingaben und Ausgabe des Entwicklungsfortschritts und Ergebnisse [siehe Abbildung 6]. Außerdem wurde in die Software eine Schnittstelle zur Datenbank und eine Schnittstelle zur OpenStudio C#-API implementiert, um Daten abfragen und in veränderter Form übertragen zu können. Da die API von OpenStudio in C# geschrieben worden ist, wurde die Software in C# umgesetzt.

Zuerst wurden die Informationen der Gebäudetypologie (IWU, 2005) mit Hilfe des Bauteilkatalogs erweitert, um Konstruktionen und Materialien für die jeweiligen Kennwerte der Gebäudetypologie zu erhalten. Anschließend wurden diese Informationen in die Datenbank

übertragen. Diese beinhaltet wie Abbildung 4 zeigt, die Eigenschaften unterschiedlicher Materialien und deren Zusammensetzung zu Konstruktionen, wie auch die Zusammenstellung zu Construction Sets. Daneben enthält die Gebäudetypologie die Informationen der Flächen, welche mit den Construction Sets verknüpft und hinterlegt werden. Neben diesem großen Datensatz beinhaltet die Datenbank die Informationen über Zeitabläufe und genutzten Zeiten unterschiedlicher Aktivitäten. Die Zeitabläufe für den Stromverbrauch sollen durch die Smart-Meter Zähler erstellt werden. Da diese zum Zeitpunkt der Ausarbeitung nicht zur Verfügung standen, wurden die Daten der Kreiswerke Main-Kinzig verwendet und als Rohdaten eingepflegt. Diese Rohdaten werden mit Hilfe der Nutzereingaben individuell angepasst. Durch diese Rohdaten existieren viertelstündliche Verbrauchswerte, wodurch genaue Aussagen über die verschiedenen Jahreszeiten getroffen werden können. Für den absoluten Verbrauch stehen die Daten des Stromsparspiegels des BMWi zur Verfügung (Berlin, 2016) und wurden zusätzlich in die Datenbank aufgenommen.

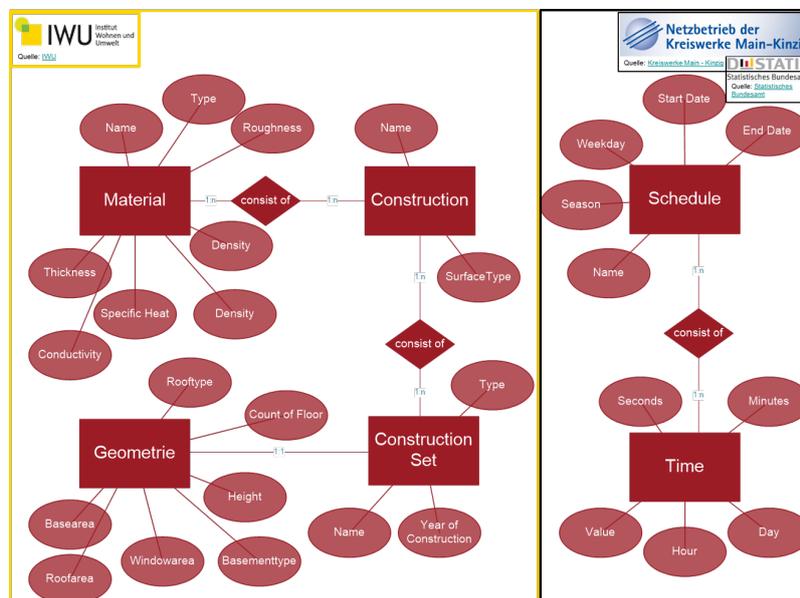


Abb. 4: Datenbank der Software

Für die Abbildung des Nutzungsverhaltens werden Daten des Statistischen Bundesamtes verwendet. Dieses erstellt in regelmäßigen Abständen eine Statistik über die Verwendung von Zeit für unterschiedliche Tätigkeiten verschiedener Personen- oder Altersgruppen (Statistisches Bundesamt, 2013). Diese Daten enthalten absolut Werte und wurden der Datenbank hinzugefügt und mit Hilfe der Stromverbrauchskurven zu Zeitverläufen konstruiert. Es wird angenommen, dass Aktivitäten wie Hausarbeit, Fernsehen und Computerspielen mit dem Stromverbrauch korrelieren, dagegen aber Schlafen, Hausaufgaben machen, etc. nicht zusammenhängen. Der Einfluss dieser Abläufe wurde untersucht und wird in der Analyse beschrieben. Durch diese Eingaben steht dem Generator, um die Prototypen der Modelle zu generieren, eine große Datenmenge zur Verfügung.

Als nächstes wurde die GUI [siehe Abbildung 6] entwickelt, um die Nutzereingaben abzufragen. Hierbei unterscheidet der Generator über eine einfache und eine erweiterte Eingabe. In erster Form werden nur das Baujahr und die Anzahl der Bewohner abgefragt. Anhand dieser Informationen wird das Construction Set aus der Datenbank gewählt und dem Benutzer

die einfache Geometrie des Außengrundrisses angezeigt. Er hat nun die Möglichkeit diese zu ändern, da die Fläche über das Verhältnis 1:3 in Breite und Länge vereinfacht berechnet wurde. Danach kann die Erstellung des Modells gestartet werden. In der erweiterten Eingabe hat der Nutzer zusätzlich die Möglichkeit die Fensterflächen abhängig der Himmelsrichtung, die Ausrichtung des Gebäudes zum Norden, das Heizmaterial, den Beheizungsgrad von Keller und Dach, die Anzahl der Bewohner aufgeteilt in Altersgruppen, den individuellen Stromverbrauch und die Eintragung von elektronischen Geräten einzupflegen. Dabei sieht der Nutzer vorerst die Standardauswahl, welche bei der einfachen Eingabe genutzt wird. Dies sind die Fensterflächen und Beheizungsgrad für Keller und Dach aus der Datenbank. Es wird angenommen, dass die Ausrichtung auf 30 Grad, als Heizmaterial Gas ist und die Personen sich aus mindestens einem Erwachsenen ergibt. Der Stromverbrauch ergibt sich aus der Anzahl der Bewohner und der Datengrundlage des statistischen Bundesamts (Berlin, 2016). Danach wird das Modell erstellt und kann simuliert werden.

Bei der Generierung des Modells werden zunächst die Fläche konstruiert, zu Räumen verknüpft und mit den Konstruktionen aus der Datenbank über die Schnittstelle zur OpenStudio C#-API hinzugefügt. Das Erzeugen der Räume wurde vereinfacht, indem dafür die Annahme über Raumverhältnisse abhängig der vorher ermittelten Länge und Breite festgelegt wurde [siehe Abbildung 5].

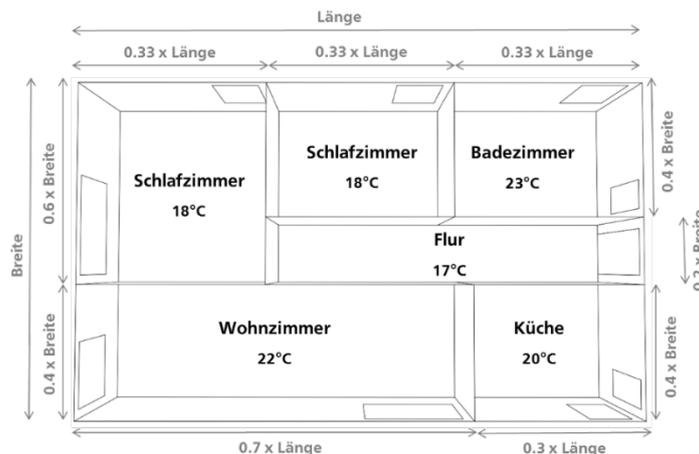


Abb. 5: Grundriss für Standard Einfamilienhaus

Nachdem dieser Grundriss für die Anzahl der Stockwerke, welche auch in der Gebäudetypologie (IWU, 2005) enthalten ist, erzeugt worden ist, werden die Fensterflächen an den jeweiligen Wandseiten gleichmäßig platziert. Dabei hilft ein selbst entwickelter Algorithmus, der die Fenster nur platziert, wenn die Fläche ausreicht, und der versucht eine gleichmäßige Aufteilung über alle Wände und Stockwerke zu gewährleisten. Bei mangelnder Fläche, wird der Nutzer darauf hingewiesen.

Sobald die Geometrie vorhanden ist, werden die einzelnen Räume im Heizkreislauf zusammengeschlossen und über die OpenStudio C#-API in das Modell hinzugefügt, wodurch eine nachträgliche Bearbeitung in OpenStudio möglich ist.

Zu guter Letzt werden die Zeitverläufe aus der Datenbank mit der Anzahl der Personen zusammengesetzt und hinzugefügt und die Personen sowie Geräte als Verbraucher hinzugefügt. Für den Fall das keine manuelle Eingabe über die erweiterte GUI getroffen wurde, werden in

die Küche ein Herd, ein Geschirrspüler und Kühlschrank hinterlegt. In das Wohnzimmer wird ein Fernseher und ein Computer hinzugefügt, das Badezimmer enthält eine Waschmaschine und jeder Raum wird mit zwei Lichtern ausgestattet. Der Nutzer wird über die erfolgreiche Modellgenerierung informiert und hat nun die Wahl, das Modell in OpenStudio zu öffnen und somit zu bearbeiten und/oder es direkt zu simulieren. Im zweiten Fall wird über die Schnittstelle zur OpenStudio C#-API die nötigen EnergyPlus Dateien erstellt und über OpenStudio simuliert.

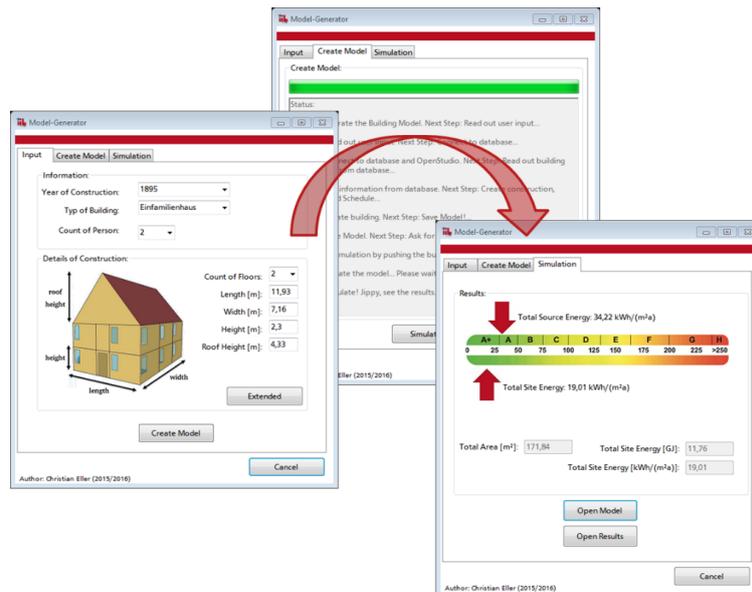


Abb. 6: GUI des Generators

Ein Fortschrittsbalken der GUI zeigt den Fortlauf der Simulation an. Anschließend werden die Ergebnisse der Simulation im Anbetracht des End- und Primärenergieverbrauchs dem Nutzer in der EnEV-Skala (=Energieeinsparverordnung) visuell dargestellt (Tuschinski, 2016) [siehe Abbildung 6]. Für eine detaillierte Einsicht in die Ergebnisse, können die Resultate von EnergyPlus angezeigt werden. Da EnergyPlus mit den Primärenergiefaktoren des amerikanischen Raums rechnet, wurde eine weitere Textdatei generiert, die die Berechnung mit den aktuellen Primärenergiefaktoren der EnEV anzeigt.

Als Ergebnis existiert nach der Nutzung des Generators die OpenStudio Datei des Modells, die Textdatei mit den aufbereiteten Ergebnissen, die IDF-Datei des Modells und die Ergebnisdatei der Simulation.

Analyse

Um den Generator validieren zu können, wurden für jedes Jahrzehnt ein Modell erstellt. All diese Modelle werden von drei Personen bewohnt. Zum Vergleich wurden die Jahresendenergieverbräuche in kWh/m^2a aus BMWi (2014) entnommen. Den Vergleich der Simulationsergebnisse der Endenergieverbräuche zeigt Abbildung 7. Es ist deutlich erkennbar, dass die Simulationsergebnisse bis 1978 den Endenergieverbräuchen der BMWi (2014) entsprechen. Die Abweichung beträgt hier nur 4 %. Es kann von einem realitätsnahen Modell gesprochen

Entwicklung eines Model Generators für private Haushalte auf Basis der IWU-Gebäudetypologie als Grundlage für EnergyPlus Gebäudesimulationen

werden. Ab dem Jahr 1980 liegen die entwickelten Modelle in ihrem Verbrauch deutlich unter dem Verbrauch der BMWi (2014). Die Abweichung beträgt 42 %. Dies ist darauf zurückzuführen, dass die Konstruktion aufgrund der eingesetzten Dämmung in diesem Zeitraum immer weniger Einfluss auf den Heizverbräuche besitzt. Der Einfluss von inneren Faktoren wie dem Trinkwasserkreislauf steigt. Allerdings wurde den Modellen dieser, aufgrund der fehlenden statistisch belegbaren Informationen, ohne Daten hinzugefügt und ist somit bei der Simulation vernachlässigt worden. Dennoch zeigt sich, dass die Ergebnisse einer gewünschten Prognose entsprechen und das die Ergebnisse des BMWi (2014) reproduziert und repräsentiert werden können.

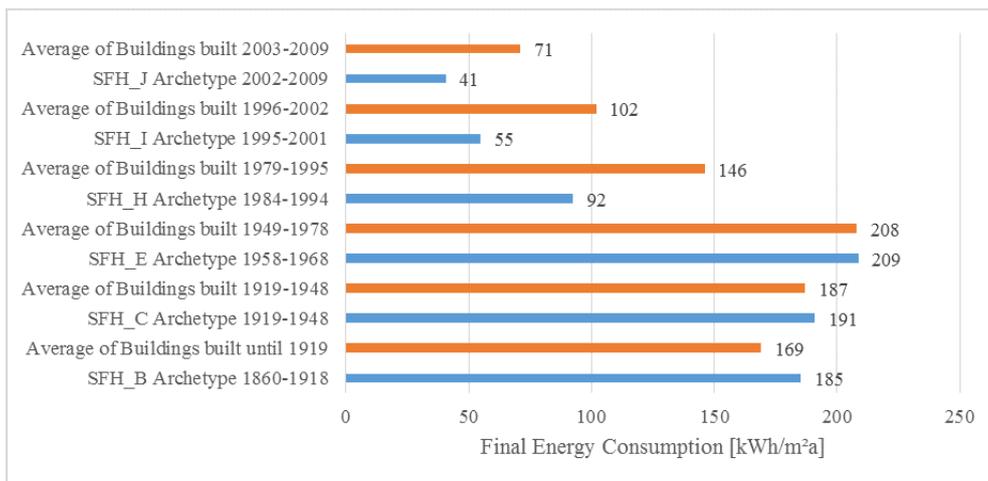


Abb. 7: Vergleich Endenergieverbrauch ermittelt durch BMWi (2014) (orange) und den entwickelten Prototypen der verschiedenen Jahrzehnte (blau) (Irmler und Eller, 2016)

Das Modell ermöglicht nun eine detaillierte Einsicht in alle Zonen und Räume, wodurch die Auswirkungen der Verbraucher und Umwelteinflüsse möglich ist. Je nach Jahreszeit ergeben sich somit andere Verlaufskurven. Diese können visualisiert werden.

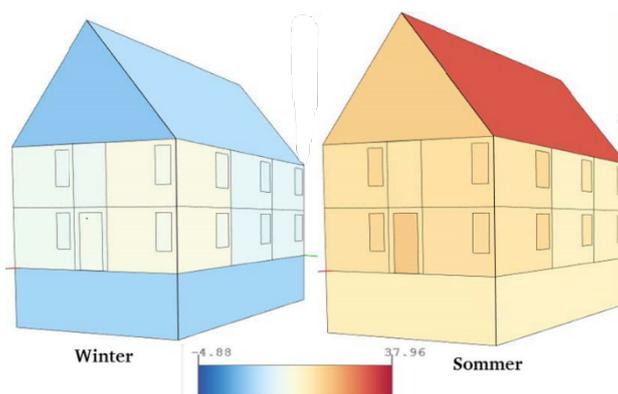


Abb. 8: Modellanzeige mit Anzeige der inneren Oberflächentemperatur im Winter und Sommer

Abbildung 8 zeigt die innere Oberflächentemperatur der einzelnen Spaces des gesamten generierten Modells im Vergleich von Winter und Sommer. Dabei ist deutlich erkennbar, dass das Kellergeschoss ohne Beheizung im Erdreich im Winter kalt und im Sommer leicht erwärmt

ist, da Sonneneinstrahlung nur indirekt Einfluss hat. Die Sonneneinstrahlung macht sich im Dachraum besonders bemerkbar und unterschiedliche Konstruktionen, wie Fenster und Türen unterscheiden sich eindeutig in ihrem Wärmezustand von den Wänden. Den Einfluss der technischen Geräte die sich hier hauptsächlich in den markierten Räumen befinden, zeichnet sich durch eine durchschnittlich höhere Raumtemperatur aus. Es wurde zur Validierung eine unterschiedliche Anzahl und Kombination von Verbrauchern hinzugefügt. Dadurch ergab sich, dass elektronische Geräte einen höheren, die Personen dagegen einen sehr geringen Einfluss haben. Diese Einflüsse entsprechen der Realität und sind nachweisbar, weshalb von einem realitätsnahen Modell gesprochen werden kann.

Fazit

Mit Hilfe von OpenStudio und dessen C#-API konnte dieser Generator entwickelt werden, der anhand der IWU-Gebäudetypologie Konstruktionen und Materialien für EnergyPlus aufbereitet und in eine vordefinierte Geometrie einarbeitet. Diese Geometrie konnte für EnergyPlus lesbar entwickelt und mit den nötigen Eigenschaften sowie Verknüpfungen zueinander ergänzt werden. Des Weiteren hat der entwickelte Generator die Möglichkeit, Menschen und ihr statistisches Verhalten in einer Gebäudehülle abzubilden, sowie Verbraucher wie elektronische Geräte mit einem realistischen Nutzungszeitplan zu integrieren, damit diese in die Simulation einbezogen werden können. Dabei wurde der Einfluss dieser Verbraucher untersucht und festgestellt, dass die elektronischen Geräte für die Energieverbräuche eine große Rolle, die Menschen hingegen, aufgrund ihrer geringen Anzahl im Wohngebäude und ihrem Nutzungsverhalten eine sehr kleine Rolle spielen.

Die Analyse der Simulationsergebnisse unterschiedlicher Modelle aller IWU-Konstruktionen zeigt, dass die Modellierung in dieser Form vollständig funktioniert und gute Resultate zurückgibt. Allerdings sind die Ergebnisse sehr von den Geometrien der Modelle, sowie der Anzahl der Geräte und zusätzlicher Verbraucher abhängig, wodurch von einem realen Modell noch nicht gesprochen werden kann. Dennoch kann man davon ausgehen, dass es erfolgreich war einen Generator zu entwickeln, der mit den Daten der IWU-Gebäudetypologie, realitätsnahe Modelle erstellen kann, welche durch OpenStudio und EnergyPlus betrachtet und simuliert werden können.

Ausblick

Mit dem entwickelten Generator ist es nun möglich, ein realitätsnahes Modell zu erstellen. Dennoch wäre es sinnvoll das Modell zu erweitern. Mit der Anpassung der elektronischen Verbräuche an die Messungen der Smart-Meter Zähler wird auch dieser Generator die Möglichkeit besitzen individuellere Gebäudenutzungen ins Modell einzuarbeiten und dieses somit direkt zu individualisieren.

Eine Erweiterung zur Einarbeitung der Grundrisse unterschiedlicher Gebäudearten wäre der nächste Schritt der Individualisierung dieses Generators und würde den Weg einer schnellen, einfachen und individuellen Simulation eines jedes Wohngebäudes in Deutschland wieder um einen weiteren Schritt verkürzen.

Des Weiteren wäre es nun interessant den Generator mit zum Beispiel Measures-Makros für OpenStudio- zu erweitern, damit kleine Veränderungen und deren Auswirkungen auf das Modell durchgeführt und untersucht werden können. So könnte zum Beispiel bei Sanierungen

Entwicklung eines Model Generators für private Haushalte auf Basis der IWU-Gebäudetypologie als Grundlage für EnergyPlus Gebäudesimulationen

der Konstruktionstyp Fenster vor dem effektiven Einbau auf dessen Auswirkung getestet und durchgerechnet werden.

Literatur

- Berlin, I. (2016). *Stromspiegel für Deutschland*. <http://www.die-stromsparinitiative.de/fileadmin/bilder/Stromspiegel/broschuere/Stromspiegel-2016-web.pdf>.
- BMWi (2005). *Gesetz zur Digitalisierung der Energiewende*. <https://www.bmwi.de/Redaktion/PDF/E/entwurf-eines-gesetzes-zur-digitalisierung-der-energiewende,property=pdf,bereich=bmwi2012,sprache=de,rwb=true.pdf>.
- BMWi (2014). *Sanierungsbedarf im Gebäudebestand*. Bundesministerium für Wirtschaft und Energie. <https://www.bmwi.de/BMWi/Redaktion/PDF/E/sanierungsbedarf-im-gebaeudebestand,property=pdf,bereich=bmwi2012,sprache=de,rwb=true.pdf>.
- building Science Group (2016). *Virtual PULSE*. <http://virtualpulse.buildsci.us/#/>.
- Building Technologies Office (2016). *Building Technologies Office: EnergyPlus Energy Simulation Software*. <http://apps1.eere.energy.gov/buildings/energyplus/>.
- Irmeler, R. und Eller, C. (2016). *A Next Generation Platform for Building Energy Modeling Based on Smart Metering Data, Point Cloud Scans, GIS and Building Typology*. In Krakau.
- IWU (2005). „Gebäudetypologie Deutschland“. In:
- NREL (2014). *BEopt*. <https://beopt.nrel.gov/downloadBEopt2>.
- Statistisches Bundesamt (2013). *Zeitverwendungserhebung Deutschland*. https://www.destatis.de/DE/Publikationen/Thematisch/EinkommenKonsumLebensbedingungen/Zeitbudgeterhebung/Zeitverwendung5639102139004.pdf?__blob=publicationFile.
- Tuschinski, M. (2016). *EnEV ab 2016: Erhöhter Neubau-Standard für Wohn- und Nichtwohngebäude*. https://www.destatis.de/DE/Publikationen/Thematisch/EinkommenKonsumLebensbedingungen/Zeitbudgeterhebung/Zeitverwendung5639102139004.pdf?__blob=publicationFile.

Agenten-basierte Echtzeitdetektion von gefährlichen Bauprozessen

Michael Kreger

*Institut für Numerische Methoden und Informatik im Bauwesen, Technische Universität
Darmstadt, Franziska-Braun-Str.7, 64287 Darmstadt, Deutschland
E-Mail: kreger@iib.tu-darmstadt.de*

Abstract Unfälle auf der Baustelle enden im Vergleich zu anderen Wirtschaftszweigen öfter mit schwereren Folgen. Bauarbeiter sind einer Reihe an Gefahren ausgesetzt, wie herunterfallenden Gegenständen, mechanischen Gefährdungen wie Absturz von großen Höhen, Stolpern, Rutschen, beim Umgang mit schweren Maschinen sowie giftigen Gasen und so weiter. Des Weiteren stellt das hohe Unfallrisiko auch ein hohen Kostenfaktor für Bauunternehmen dar. Aus diesen Gründen sind Bauunternehmen besonders interessiert an Maßnahmen zur Gefahrenprävention, die mit einer frühzeitigen Lokalisierung der Gefahrenfaktoren verbunden ist. Die besonderen Herausforderungen für den Arbeitsschutz auf der Baustelle, aufgrund der wechselnden Bedingungen durch Klimaeinflüsse oder nicht stationärer Arbeiten, fordern ein System, dass die Prozessüberwachung in Echtzeit ausführen und anhand dessen Zukunftsprognosen zu erstellen um die Entstehung von Gefahrensituation verhindern zu können.

Die Prozessüberwachung kann mit dem Einsatz geeigneter vernetzter Sensoren erfolgen, die ständig Informationen zum Erfassungsbereich des Sensors an eine Netzwerk weitergeben. Die gesammelten Daten stehen jedoch in keiner Beziehung zueinander, so dass eine Auswertung der Gesamtsituation hier schwierig wird. Zudem bedarf es Expertenwissen zur Interpretation der bereitgestellten Informationen, deren manuelle Bearbeitung viel Zeit in Anspruch nimmt.

Im Rahmen des Forschungsprojekts integratives-Safety-Awareness-Model (iSAM) wurden beispielhaft einige Sensoren mit Methodiken aus dem Bereich der Agentenmodellierung für den Einsatz der Gefährdungsdetektion auf Baustellen erweitert und in einem Multiagentensystem vernetzt. Der konzeptionelle Ansatz zur Entwicklung von Agenten zur Bedienung, Kontrolle und Kalibrierung eines Sensors und zur Detektion von Gefahrenfaktoren auf Baustellen soll im vorliegenden Beitrag präsentiert werden. Die Echtzeitfähigkeit wurde mit Hilfe eines 3D Sensors validiert.

Keywords: Sensornetzwerk, Agentensystem, Gefährdungen auf Baustellen

1 Einleitung

Arbeiter sind auf Baustellen vielen verschiedenen Gefahren ausgesetzt. Dies belegen auch Unfallzahlen, die jährlich von der Bundesanstalt für Arbeitsschutz und Arbeitsmedizin (BAuA)

sowie für das Baugewerbe von der Berufsgenossenschaft für das Baugewerbe (BG Bau) herausgegeben werden. Insgesamt wurden im Jahr 2014 955.919 Arbeitsunfälle gemeldet (BAuA, 2016). Mit 103.731 Unfällen vermeldet das Baugewerbe (BG Bau, 2015) somit ca 10% aller Arbeitsunfälle. Lediglich Wirtschaftszweige mit sehr viel mehr Beschäftigten melden noch höhere Unfallzahlen. Das Baugewerbe ist jedoch mit einer Unfallquote von 62 je 1000 Vollarbeiter trauriger Spitzenreiter in der Statistik der Bundesanstalt für Arbeitsschutz und Arbeitsmedizin (BAuA, 2016).

Das häufigste Unfallmuster auf Baustellen ist laut DGVU, 2015 mit 26,3% aller meldepflichtigen und 45,7% aller tödlichen Unfälle der Aufprall auf bzw. gegen ortsfeste Gegenstände aus der Bewegung des Verunfallten heraus, vor allem durch Stolpern oder Absturz. An zweiter Stelle der nicht tödlichen Unfälle (32,7%) ist der Kontakt mit scharfen, spitzen, harten und rauen Gegenständen zu nennen. Hierunter zählen insbesondere der Umgang mit baustellentypischen Handwerkzeugen, Maschinen sowie Baumaterialien. Die dritte Gruppe bilden Gegenstände, die in Bewegung sind. Dieses Unfallmuster ist das dritthäufigste (17,1%) aller meldepflichtigen und mit 30,4% das zweithäufigste aller tödlichen Unfälle.

Diese Zahlen sind Grund genug neue Ansätze zur Vermeidung von Unfällen auf Baustellen zu erarbeiten. Durch die äußerst komplexen und nebenläufigen Bauprozesse wird es voraussichtlich auch in naher Zukunft jedoch nicht möglich sein alle gefahrenträchtigen Tätigkeiten auf einer Baustelle in Echtzeit zu erfassen und rechtzeitig in den Prozessablauf einzugreifen, um Unfälle wirkungsvoll verhindern zu können. Im vorliegenden Ansatz, der im Rahmen des Forschungsprojekts integrative Safety-Awareness-Modell (iSAM) entwickelt wurde, liegt das Augenmerk vielmehr auf einer Steigerung des Bewusstseins vor gefährlichen Situationen. Hierbei wurden vor allem zwei Ziele angestrebt: Zum einen sollen Sicherheits- und Gesundheitsschutzkoordinatoren (SiGeKo) bei ihrer Arbeit unterstützt werden, indem Informationen, die bereits aus der Planung heraus auf gefährliche Situationen hindeuten besonders hervorgehoben werden. Zudem sollen während des laufenden Betriebs einzelne Tätigkeiten und Aktionsräume nach deren Unfallrisiko bewertet und Akteure entsprechend der Gefährdungslage informiert werden.

2 Integratives Safety-Awareness-Modell (iSAM)

Das iSAM-System basiert auf der Auswertung von Wissen über Bautätigkeiten sowie Gefährdungen und besteht im wesentlichen aus vier Komponenten. Abbildung 1 zeigt den Aufbau des Systems, dessen Kernkomponente das Bewusstsein (Awareness) ist. Hierbei handelt es sich um die Verknüpfung von Wissen in Form von semantischen Netzen (Ontologien). Dieses Wissen (Knowledge) stammt aus den unterschiedlichsten Quellen und muss vor der Zusammenführung "homogenisiert" werden (Kreger und Rüppel, 2016), das heißt alle Informationen werden bei iSAM in Ontologien überführt um sie Verknüpfen zu können. Zum Wissen, das für die Sicherheitsbeurteilung einer Baustelle relevant ist, gehören in erster Linie allgemeine Zusammenhänge und Informationen wie Herstellungsverfahren, Bauwerksstrukturen, Betriebsanleitungen, Gesetze und so weiter.

Für die Einschätzung spezieller Vorgänge auf einer bestimmten Baustelle sind zudem Informationen aus der Planung wie Bauwerksmodell und Bauablauf zum jeweiligen Bauvorhaben notwendig. Eine weitere Komponente ist für die Umwelterkennung (Sensing) mittels Sensornetzwerk verantwortlich. Hierbei kommen sowohl bereits vorhandene Sensoren, wie Kameras und Abstandsensoren an Baufahrzeugen aber auch weitere neue Sensoren, die zur Sicherheitsbeurteilung genutzt werden können, zum Einsatz. Für diese Sensoren müssen geeignete Schnittstellen geschaffen werden, so dass sie in ein Netzwerk integriert werden können. Der Vorteil

der Vernetzung der Sensoren liegt auf der Hand. Informationen die zur Zeit nur autonom bzw. systemintern, z.B. in einen Baufahrzeug, genutzt wurden, können nach der Vernetzung auch für die Beurteilung der Gesamtsituation auf der Baustelle herangezogen werden. Eine weitere wichtige Komponente dient der Verarbeitung (Processing) aller vorhandenen Informationen, dem Wissen über die Baustelle. Den Aufbau dieser Komponente und das Konzept der Umsetzung mittels Software-Agententechnologie, sind Gegenstand der vorliegenden Arbeit und werden im Folgenden näher erläutert.

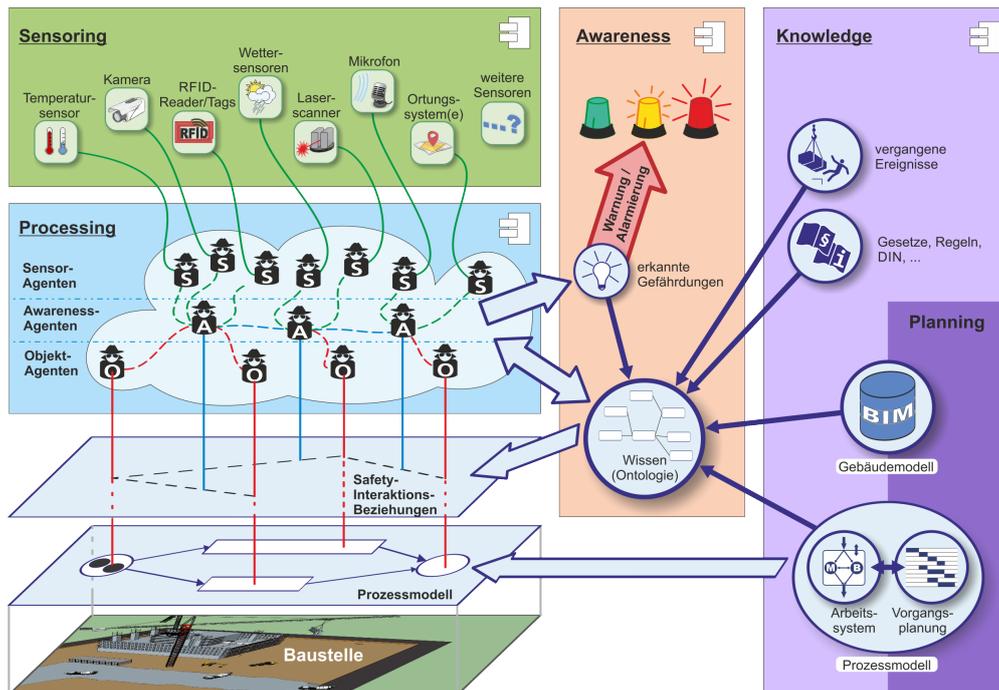


Abb. 1: iSAM System

3 Sensoren zur Erfassung von Gefährdungsdaten

Die Erfassung des Geschehens auf der Baustelle soll beim zugrunde liegenden Forschungsprojekt wie bereits erwähnt mittels Sensornetzwerk erfolgen. Hierbei sollen nicht nur Sensoren eingesetzt werden, die primär auf Unfallprävention, wie bspw. Entfernungsmesser zur Kollisionsvermeidung, spezialisiert sind. Viele weitere messbare Umweltbedingungen können zur Einschätzung der Gefährdungslage auf Baustellen dienlich sein. Hierzu zählen unter anderem Sensoren zur Beurteilung der Wettersituation (Feuchte, UV, usw.) sowie Mikrofone zur Detektion bestimmter Prozesse anhand typischer Geräuschmuster (Bohnhoff, 2015) oder zur Bestimmung des Schallpegels. Dies ist nur eine kleine Auswahl und soll verdeutlichen, das beinahe jeder Sensortyp nützliche Informationen liefern kann. Zudem kann angenommen werden, dass eine Erhöhung der Anzahl und eine geschickte Lokalisierung der Sensoren auf dem Baugelände die Erfassungsmöglichkeiten verbessert.

Je mehr Sensoren zur Umgebungsüberwachung bereitstehen, desto mehr Daten werden allerdings auch generiert. Je nach Sensortyp und Abtaststrategie (zeitliche Auflösung) variiert die anfallende Datenmenge erheblich. Für einen digitalen Temperatursensor, der sekundlich einen

Messwert als Gleitkommazahl mit einfacher Genauigkeit ermittelt, bedeutet das $32 \frac{bit}{s}$. Ein visueller Sensor wiederum, der 25 Bilder in der Sekunde in TrueColor (24 Bit je Bildpunkt) und einer VGA Auflösung von 640x480 Pixel (307.200 Pixel) ausgibt, produziert in jeder Sekunde $25 \cdot 307200 Pixel \cdot 24 \frac{bit}{Pixel} \approx 184 Mio. \frac{bit}{s} \approx 23 \frac{MB}{s}$ Daten.

Einfluss auf die anfallende Datenmenge kann durch die Wahl angepasster Abstraten je Sensor genommen werden. Sensoren für die Ermittlung der Wettersituation müssen bspw. nicht sekundlich Daten senden, da sich das Wetter eher langsam ändert. Die zu verarbeitende Datenmenge zur Erkennung der aktuellen Baustellensituation hängt demnach stark von der Definition von „Echtzeit“ ab, also der Systemvorgabe was noch rechtzeitig ist.

Eine weitere Möglichkeit ist die Vorverarbeitung (Preprocessing) der Sensordaten. Das heißt der Sensor übermittelt nicht alle Rohdaten, sondern filtert oder ermittelt relevante Daten und stellt nur diese Informationen dem Netzwerk zur Verfügung. Man spricht in diesem Zusammenhang auch von einem intelligenten Sensor oder Smart-Sensor (siehe Abbildung 2).

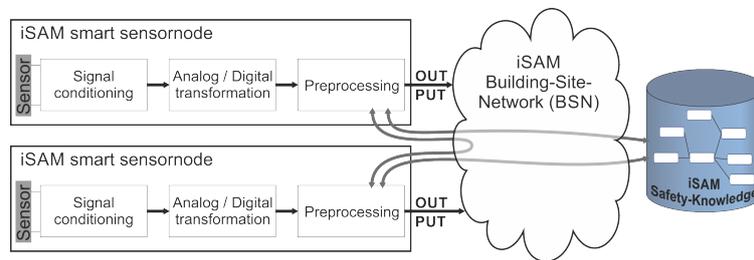


Abb. 2: iSAM intelligenter Sensorknoten

4 Agentensystem zur Analyse von Gefährungsdaten

Oft verfügen intelligente Sensoren zudem über weitere kognitive Fähigkeiten. Hierzu zählen bspw. die Fähigkeit ein Ziel zu erreichen oder auch kontextabhängige Entscheidungen zu treffen. Dazu benötigen die Sensorknoten Zugang zu „Hintergrundwissen“ und die Möglichkeit mit anderen Sensoren kommunizieren zu können. Dadurch können die durch den Sensor ermittelten Rohdaten kontextabhängig interpretiert werden. Abbildung 2 zeigt schematisch die Vernetzung solcher intelligenter Sensorknoten. Die Knoten nutzen ein spezielles Baustellen-Netzwerk (BSN) als Plattform um Zugang zum bereitgestellten Wissen zu erhalten und um Informationen mit anderen Knoten im BSN auszutauschen. Um Sensorknoten mit den beschriebenen Fähigkeiten auszustatten wird zum einen zusätzliche Hardware benötigt. Da sich die zu verarbeitenden Informationen im wesentlichen auf die Daten des Sensors und die Kommunikation beschränken kann auf eine performante Recheneinheit verzichtet werden. Gängige Einplatinen-PCs wie ODROIDS von Hardkernel oder Raspberry PIs von der Raspberry PI Foundation u.ä. sind geeignete Hardware für diesen Einsatzzweck. Zudem wird eine spezielle Software auf diesen Recheneinheiten installiert, die das Preprocessing mit den kognitiven Fähigkeiten übernimmt.

Eine solche autonom „handelnde“ und interaktive Software wird in der Informatik als Software-Agenten bezeichnet. Das iSAM-System kennt neben den „Sensoragenten“ weitere spezialisierte intelligente Softwareknoten, die alle im BSN miteinander vernetzt sind. Abbildung 3 zeigt drei Grundtypen für Software-Agenten, die bei iSAM definiert wurden. Dies sind neben bereits beschriebenen Sensoragenten noch Objekt- und Awareness-Agenten. Die iSAM Agenten teilen sich mit dem BSN eine gemeinsame Umgebung (Umwelt). Somit entsteht ein Multi-Agenten-System (MAS).

Ergänzend zu den Sensoragenten, die in Verbindung mit Sensorhardware implementiert sind, existieren auch Sensoragenten, die ohne Hardware auskommen. Hierbei handelt es sich um virtuelle Sensoren, also Softwareeinheiten, die auf die Fusion von Sensordaten spezialisiert sind. Das heißt sie lauschen im Netzwerk nach unterschiedlichen Sensordaten um daraus neue Informationen zu generieren. Als Beispiel kann hier ein Windwarn-Sensor genannt werden. Ein solcher Sensor beobachtet die Informationen von Hygrometer- und Wind-Sensoren. Sollte bei geringem oder keinem Wind der Luftdruck in kurzer Zeit stark abfallen, ist dies ein sicheres Zeichen für einen drohenden Sturm.

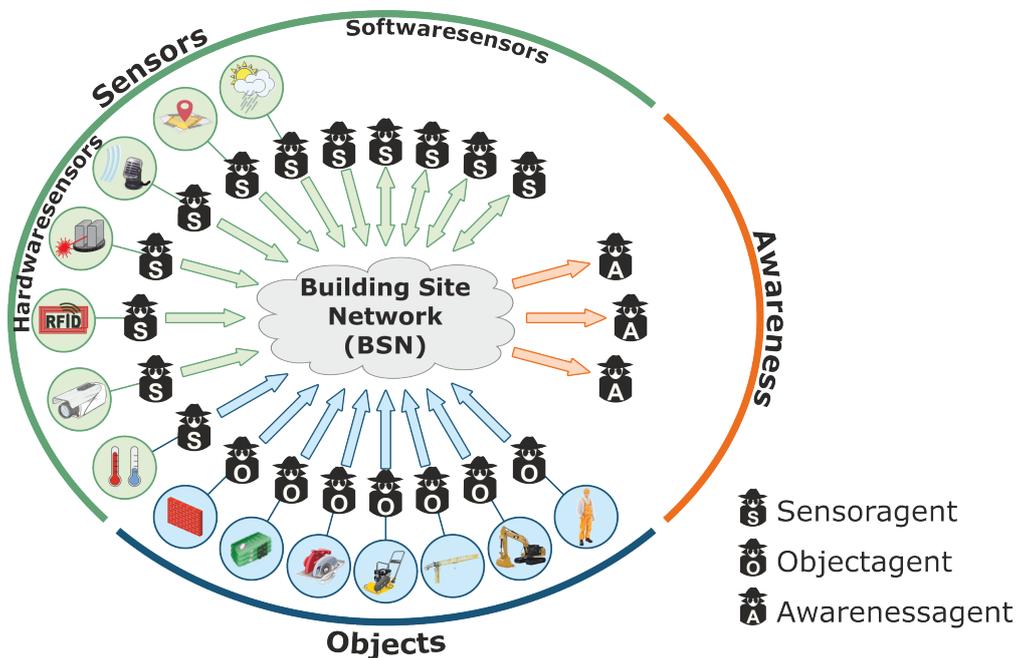


Abb. 3: iSAM Agententypen.

Objektagenten dienen dazu allen „Dingen“ auf der Baustelle Zugang zum BSN zu ermöglichen. Letztendlich handelt es sich bei den Objektagenten um Repräsentationen von Gegenständen wie Maschinen, Baumaterialien oder ganzen Bauteilen im Safety-Awareness-Modell. In Kreger und Rüppel, 2015 wurde bereits eine mögliche Erweiterung des Building Information Model zum Building Awareness Model (BAM) für solche Einsatzszenarios empfohlen. Modellerte Bauteile, die im Sinne von BIM mit zusätzlichen bauteil- bzw. objektbezogenen Informationen auf Attributebene angereichert sind, erhalten mit BAM zudem Funktionalitäten für ein gewisses „Bauteilbewusstsein“. Die Bauteilklassen agieren und reagieren im Modell in Form von Bauteil-Agenten auf Veränderungen und Umwelteinflüsse. Die Wissensbasis der Bauteilagenten könnte neben der Topologie des Bauwerks auch alle gängigen Konstruktionsregeln aus den einschlägigen Normen umfassen. Die Repräsentation von Werkzeugen, Maschinen und Baugeräten besitzen im Modell ebenfalls die Fähigkeiten sich mitzuteilen (z.B. der eigene Status) und auf Informationen anderer Netzwerkteilnehmer zu reagieren. Die hardwaremäßige Umsetzung könnte mit zusätzlicher Hardware realisiert werden, die mit der Steuereinheit der Maschine verbunden ist. Bei Baufahrzeugen kann dies bspw. mit einem speziellen Knoten

im CAN-Bus realisiert werden, der als Schnittstelle zum BSN fungiert. Das Baugerät wird dadurch zum selbstständigen Individuum innerhalb der Modellumgebung. Auch Menschen werden Zweck halber als Objektagent mit deren speziellen Rolle auf der Baustelle repräsentiert. Objekte bzw. die Objekt-Agenten werden im Modell instanziiert (ingecheckt), wenn die realen Objekte im Baustellenbereich eintreffen.

Während Sensor- und Objekt-Agenten vorwiegend Daten an das Netzwerk senden, empfangen Awareness-Agenten, die dritte Gruppe von Agententypen, vor allem Daten. Die Awareness-Agenten sind auf die Risikobeurteilung von Gefährdungssituationen spezialisiert. Sie beurteilen die Daten, die andere Teilnehmer des Netzwerks zur Verfügung stellen und geben eine Einschätzung zur Gefährdungslage für den Bereich für den sie vorgesehen sind. Dieser Überwachungsbereich kann entweder ein Baustellenabschnitt (ein räumlich begrenztes Gebiet) sein oder auch bestimmte Gefährdungen (z.B. „Absturz von Gerüst“).

5 Building Site Network (BSN)

Als Agentenumgebung (Plattform) wird ein spezielles Netzwerk, das Building Site Network (BSN), genutzt, das auf der Baustelle bereitgestellt werden muss. Innerhalb des BSN agieren alle Agenten auf einer Ebene. Das heißt es gibt keine hierarchischen Strukturen. Dies hat den Vorteil, dass neue Agenten, egal welchen Typs, jederzeit in das BSN integriert werden können und aus dem Netzwerk ausscheiden können.

Das Netzwerk folgt den Spezifikationen der Foundation of Physical Intelligent Agents (FIPA). Die Festlegung auf den FIPA-Standard soll eine universelle Entwicklung von iSAM-Softwareknoten garantieren, die auf allen Baustellen, also in unterschiedlichen Umgebungen verwendbar sind. Hierzu gehören der Transport von Nachrichten in der Agent-Communication-Language (ACL) sowie verschiedene Management-Funktionalitäten, die der Organisation bzw. der Administration der Software-Agenten dienen.

Jeder neue Agent, der in die Baustellenumgebung eingecheckt wird muss beim Agent Management System (AMS) registriert werden und erhält einen eindeutigen Agent Identifier (AI). Zudem überwacht das AMS den Lebenszyklus aller Agenten im BSN. Das AMS selbst wird ebenfalls durch einen Softwareknoten, einen spezialisierten Agenten im BSN, realisiert (FIPA, 2004).

Ein weiterer Management-Agent sorgt für die korrekte Weiterleitung von Nachrichten. Ähnlich wie ein SMTP Server im Internet kümmert sich innerhalb des BSN ein Message Transport System (MTS) für die Verteilung von ACL Nachrichten. Als Transportmedium dient FIPA-konform das Hypertext Transfer Protokoll (HTTP). Wegen der dezentralen hierarchielosen Struktur erfolgt die Weitergabe von Nachrichten generell als Broad- oder Multicast, also einem Rundruf an in der Regel alle oder selten an typengleiche Agenten. Jeder Agent entscheidet selbst, ob die empfangene Nachricht für ihn relevant ist oder nicht.

Auf ein Directory Facilitator (DF) wie er bei Agentenplattformen oft zum Einsatz kommt, wird im vorliegenden Anwendungsfall verzichtet. Beim DF können Agenten bestimmte Dienste, die sie anbieten registrieren lassen, dies ist vergleichbar mit den „Gelben Seiten“. Die Komponente ist nach FIPA optional (FIPA, 2004) und bringt für iSAM keinen Mehrwert, da alle Nachrichten ungeachtet der angebotenen Dienste an alle Teilnehmer gesendet werden.

In Abbildung 4 ist, in Anlehnung an das FIPA Agent Reference Model (FIPA, 2004), das Agenten Referenzmodell für die drei beschriebenen iSAM Agententypen im Building Site Network abgebildet.

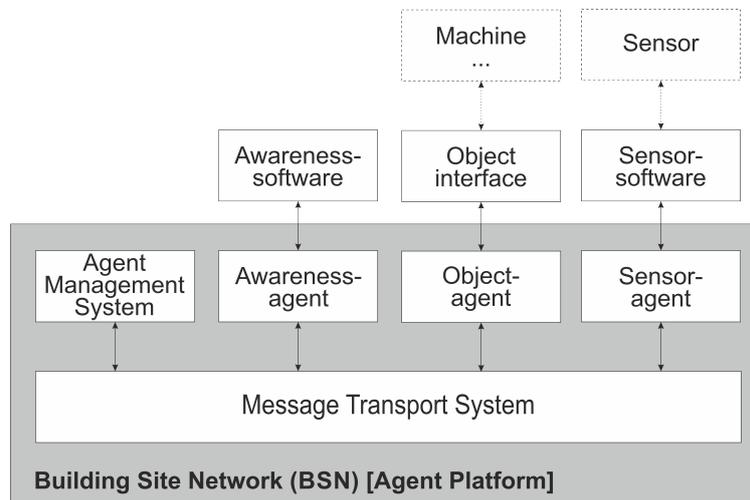


Abb. 4: iSAM Agenten Referenzmodell in Anlehnung an FIPA, 2004.

6 Zusammenfassung und Ausblick

Die hohen Unfallzahlen im Baugewerbe sind oft die Folge unzureichender Organisation. Oftmals führen auch das Missachten geltender Sicherheitsbestimmungen zu schweren Unfällen, die teilweise sogar tödlich verlaufen. Eine ganzheitliche Überwachung aller Vorgänge auf Baustellen in der Art, dass bei zu hohem Gefährdungsrisiko in den Prozess rechtzeitig eingegriffen und ein Unfall vermieden werden kann, ist in naher Zukunft nicht realisierbar. Daher wird im Rahmen des Forschungsprojekts integratives Safety-Awareness-Modell der Ansatz verfolgt das Bewusstsein für gefährliche Situationen auf Baustellen zu verbessern. Dadurch können zum einen Sicherheitsbestimmungen belegt und dadurch Arbeiter zur Einhaltung animiert werden. Außerdem werden Sicherheitsbeauftragte Personen und Arbeiter vor generell unsicheren Tätigkeiten gewarnt, was in der jeweiligen Situation zu einer erhöhten Aufmerksamkeit führen kann. Ein Teilaspekt des Forschungsprojekts ist die Verarbeitung vorhandener Informationen, die die Baustellensituation beschreiben, in Echtzeit. Im vorliegenden Paper wird der Aufbau eines Multi-Agenten-Systems (MAS) für diese Aufgaben im iSAM-System beschrieben. Für das präsentierte MAS wurden drei Software-Agententypen vorgestellt. Die drei Typen sind jeweils auf den Gebieten Sensoren, Objekte und Awareness spezialisiert und agieren vernetzt im Building Site Network (BSN), der Plattform der iSAM-Software-Agenten.

Mit den im Kontext von iSAM geschaffenen Grundlagen können in der Zukunft Standardisierungen im Bereich der Baustellensicherheit geschaffen werden. Hierzu zählen vor allem Schnittstellen zu vorhandenen Bussystemen von Baugeräten (z.B. CAN-Bus) und der Gebäudeautomation (z.B. EIB/KNX, Profibus usw.). Ziel der Standardisierung ist, dass Baugeräte, Werkzeuge usw. problemlos auf jeder Baustelle, die über ein Building Site Network zur Gefährdungsbeurteilung (iSAM-BSN) verfügt, integriert werden können. Der Bauherr muss dafür nur die Netzwerk Infrastruktur bereitstellen.

Literatur

- BG Bau (2015). *Berufsgenossenschaft der Bauwirtschaft: Jahresbericht 2014*. Berlin.
- BG Bau (2015). *Deutsche Gesetzliche Unfallversicherung: Arbeitsunfallgeschehen 2014 - Statistik*. Berlin.
- BG Bau (2016). *Bundesanstalt für Arbeitsschutz und Arbeitsmedizin: Sicherheit und Gesundheit bei der Arbeit 2014 - Unfallverhütungsbericht Arbeit*. Berlin.
- Bohnhoff, C. (2015). „Bauprozesserkenennung mittels akustischer Signalmuster“. Bachelor-Thesis. Institut für Numerische Methoden und Informatik im Bauwesen, Technische Universität Darmstadt.
- Foundation for Intelligent Physical Agents FIPA (2004). *Agent management specification SC00023K*. <http://www.fipa.org/specs/fipa00023/>.
- Kreger, M. und Rüppel, U. (2015). „Integrierte Ortung zur Vermeidung von Unfällen auf Baustellen“. In: *145. DVW-Seminar: Interdisziplinäre Messaufgaben im Bauwesen*. (Darmstadt, 26.–27. März 2015). Hrsg. von DVW - Gesellschaft für Geodäsie Geoinformation und Landmanagement e.V. Bd. 79. Augsburg: Wißner-Verlag.
- Kreger, M. und Rüppel, U. (2016). „Towards Hazards Detection on Building Sites Using Safety Interaction Matrices (SIM)“. In: *16th International Conference on Computing in Civil and Building Engineering (ICCCBE)*. (6.–8. Juli 2016). Hrsg. von Y. Nobuyoshi und M. Koji. Osaka, Japan.

An Embedded Physics-Based Modeling Approach for Wireless Structural Health Monitoring Systems

Kosmas Dragos

*Chair of Computing in Civil Engineering, Bauhaus University Weimar, Coudraystr. 7, 99423
Weimar, Germany
E-Mail: kosmas.dragos@uni-weimar.de*

Abstract Embedded computing is often applied in wireless structural health monitoring (SHM) systems to avoid the power-consuming transmission of raw time series from wireless sensor nodes to a centralized server. In this context, the embedment of algorithms performing a variety of monitoring tasks is a subject of ongoing research. Most embedded algorithms conduct on-board data processing in an attempt to extract meaningful monitoring information from the raw time series. However, little attention has been paid to using embedded computing for enhancing the ability of wireless sensor nodes to perceive the physical properties of the monitored structure. In this paper, an embedded physics-based modeling approach for wireless SHM systems is presented. First, a finite element (FE) model of the monitored structure is created and distributedly embedded into wireless sensor nodes. Then, acceleration response data is collected by the sensor nodes and processed locally by the sensor nodes to extract information on the dynamic characteristics of the structure. Based on the comparison between the extracted dynamic characteristics and the dynamic characteristics of the FE model, the parameters of the FE model are optimized to describe the structural behavior. Validation tests of the proposed approach are conducted on a laboratory test structure, showcasing the ability of the embedded model to describe the structural behavior.

Keywords: Structural health monitoring, wireless sensor networks, embedded computing, physics-based models, finite element analysis

1 Introduction

In recent years, wireless structural health monitoring (SHM) has been increasingly employed to address the problems related to the deployment of conventional cable-based systems in terms of costs and installation efforts. Embedded computing is an integral part of wireless SHM systems; exploiting the processing power integrated into wireless sensor nodes to perform monitoring tasks has been a topic of ongoing research. In this direction, research endeavors are focused on performing on-board processing of collected data, rather than sending the collected data to a centralized server. Minimizing the wireless communication is of particular merit to the power autonomy of wireless SHM systems (Lei et al., 2010).

Embedded algorithms in wireless sensor nodes are designed to execute a broad wealth of monitoring tasks, covering part (or the whole) of the monitoring process. While the type of

tasks allocated to embedded algorithms depends on the monitoring objective, the common goal of embedded algorithms is to extract the meaningful information locally and to eliminate the need for wirelessly transmitting large amounts of data to a centralized server. In wireless SHM systems, parts of monitoring are performed by “embedded models”, which employ embedded algorithms that execute the tasks according to the monitoring objective. Embedded models encompass a variety of monitoring processes, such as system identification, damage detection, and fault diagnosis.

Lots of research has been done on embedded models in wireless SHM systems. A review of research approaches implementing embedded models to execute various monitoring tasks can be found in Dragos and Smarsly (2015a). For instance, embedded autoregressive models with exogenous inputs for damage detection have been proposed by Lynch et al. (2004). Swartz et al. (2007) presented embedded system identification into a prototype wireless sensor node, while embedded models for the on-board extraction of structural dynamic characteristics, such as eigenfrequencies and mode shapes has been presented by Zimmerman et al. (2008). Zimmerman and Lynch (2007) proposed an embedded model with simulated annealing algorithms for model updating, while Dragos and Smarsly (2015b) have proposed an embedded damage detection model based on the dynamic equation of motion.

While most research approaches on embedded models address the issue of preserving the resources of wireless sensor nodes, little attention has been paid to enabling the sensor nodes to physically perceive the monitored structure. In this paper, a physics-based embedded modeling approach for wireless SHM systems is presented. The proposed approach is based on finite element (FE) modeling and the principles of dynamic substructuring. In structural engineering, considering the entire structure results in centralized analysis, which is cumbersome and inefficient for wireless sensor nodes. To utilize the embedded computing capabilities and, thus, enhance the decentralization of wireless SHM systems, the monitored structure is analyzed in segments (substructures), and at least one sensor node is responsible for each segment. Hence, the structure is first divided into substructures and an FE model for each substructure is embedded into the sensor nodes corresponding to the substructure under consideration. Second, local dynamic characteristics (mode shapes) are derived at a substructure level from collected acceleration response data. Finally, the parameters of the FE model are optimized to fit the extracted dynamic characteristics.

In the first section of this paper, the background of the proposed approach on structural dynamics and dynamic substructuring is presented. In the second section, the implementation of the embedded modeling approach into a wireless SHM system is illuminated. Finally, in the third section, laboratory validation tests of the proposed approach are presented demonstrating the ability of the approach to describe the behavior of the monitored structure.

2 Mathematical background

The principles of structural dynamics are typically employed in structural health monitoring to obtain information on the dynamic characteristics of monitored structures, such as eigenfrequencies and mode shapes. The dynamic characteristics of structures are obtained by collecting and processing acceleration response data from the structure. The analysis for obtaining the dynamic characteristics of structures is either performed on the entire structure or in a stepwise manner, by segmenting the structure into substructures (i.e. using dynamic substructuring). In this section, the basic principles of structural dynamics are presented. Furthermore, the methods

for extracting the dynamic characteristics from processed acceleration response data are briefly discussed. Finally, the dynamic substructuring method used in this study is explained.

2.1 Structural dynamics

In structural dynamics, the behavior of structures is described by the equation of dynamic equilibrium (Eq. (1)).

$$m(w) \cdot \frac{\partial^2 x(w, t)}{\partial t^2} + c(w) \frac{\partial x(w, t)}{\partial t} + k(w) \cdot x(w, t) = F(w, t) \quad (1)$$

In Eq. (1), m is the mass, c represents the damping coefficient, and k is the stiffness of the structural system, respectively. The displacement at coordinate w is denoted by x , time is represented by t , while F is the force. Eq. (1) describes the behavior of a continuous structural system; however, in structural analysis the direct use of such a formulation is cumbersome. Hence, discrete structural systems are typically considered with lumped dynamic properties at m selected degrees of freedom (DOFs) of the structure. By discretizing Eq. (1), Eq. (2) is obtained.

$$\mathbf{M}_{mm} \cdot \ddot{\mathbf{x}}_m + \mathbf{C}_{mm} \cdot \dot{\mathbf{x}}_m + \mathbf{K}_{mm} \cdot \mathbf{x}_m = \mathbf{F}_m \quad (2)$$

where, \mathbf{M} is the mass matrix, \mathbf{C} is the damping matrix, and \mathbf{K} the stiffness matrix, while $\ddot{\mathbf{x}}$ is the acceleration vector, $\dot{\mathbf{x}}$ is the velocity vector, \mathbf{x} is the displacement vector, and \mathbf{F} is the force vector.

2.2 Extraction of dynamic characteristics from processed acceleration response data

For the extraction of dynamic characteristics, acceleration response data collected from selected locations of the structure is used either in its raw form (i.e. in the time domain), or transformed into the frequency domain, i.e. in terms of frequencies and amplitudes of a multitude of harmonic functions.

In the time domain, the direct inverse solution of Eq. (2) yields the mass, damping, and stiffness parameters of the monitored structure. However, in practice, while deriving the inverse solution of Eq. (2), three major problems arise: 1) at the right-hand side of Eq. (2) the force applied to the structural system is typically unknown, 2) the number of DOFs in typical FE models is larger than the number of locations of the structure where acceleration response data is collected, thus leading to an ill-posed problem, and 3) the velocity vector and the displacement vector are calculated from the acceleration vector via numerical integration algorithms, the performance of which is unstable due to measurement factors, such as noise and quantization.

In the frequency domain, one of the most widely used methods for extracting dynamic characteristics of structures is the “frequency domain decomposition” (FDD) (Brincker et al., 2000). In FDD, acceleration response data is first transformed into the frequency domain via Fourier

transform algorithms, such as the fast Fourier transform (FFT) (Cooley and Tukey, 1965). Subsequently, the Fourier values F from r sets of acceleration response data corresponding to peaks of the frequency spectrum, indicative of resonant response (and therefore to modes of vibration), are used to formulate the spectral density matrix \mathbf{G} (Eq. (3)).

$$\mathbf{G}_{ij} = \sum_{i=1}^r \sum_{j=1}^r F_i \cdot \bar{F}_j^* \quad (3)$$

In Eq. (3), the overbar and the star denote complex conjugate. The mode shapes at the frequency corresponding to the selected peaks are extracted by applying singular value decomposition to the spectral density matrix, as shown in Eq. (4).

$$\mathbf{G}_y = \mathbf{U}\mathbf{S}\mathbf{U}^T \quad \mathbf{U} = [\mathbf{u}_{j1} \quad \mathbf{u}_{j2} \quad \mathbf{u}_{j3} \quad \mathbf{u}_{j4} \quad \dots \quad \mathbf{u}_{rj}] \quad \omega = \omega_p \Leftrightarrow \mathbf{u}_{j1} \cong \varphi_p \quad (4)$$

In Eq. (4), \mathbf{U} is the matrix containing r singular vectors \mathbf{u} (equal to the number of sets of acceleration response data), \mathbf{S} is the $r \times r$ diagonal matrix containing the singular values, ω_p is the natural frequency at the selected peak, and φ_p is the corresponding mode shape.

In this paper, the FDD is applied for extracting the natural frequencies and the mode shapes (“extracted mode shapes”), due to the aforementioned problems regarding the application of time domain methods. For updating the initially created FE model of the monitored structure, the natural frequencies and the mode shapes of the FE model (“FEM mode shapes”) must be compared to the extracted mode shapes. To perform the comparison in a decentralized manner, the FEM mode shapes must correspond to the substructures, into which the monitored structure has been divided. Deriving the extracted mode shapes at a substructure level (extracted local mode shapes) is straightforward; the r sets used in Eq. (3) and Eq. (4) correspond only to the sensor nodes of the substructure under consideration. However, for deriving FEM local mode shapes, dynamic substructuring methods in the modal domain are necessary. In this context, the Craig-Bampton method on component mode synthesis is employed, which is explained in the next subsection.

2.3 The Craig-Bampton method on component mode synthesis

Dynamic substructuring facilitates computationally intensive analysis of large structural systems. In dynamic substructuring the structure is divided into substructures and interfaces, which are analyzed separately. The global analysis results are, subsequently, obtained by synthesizing the results of each substructure and by fulfilling the interface compatibility criteria. Dynamic substructuring can be applied in the physical domain, i.e. considering the structural parameters (mass, damping, stiffness), in the frequency domain, i.e. representing the structure by frequency response functions between the input and the output, or in the modal domain, i.e. using mode shapes to represent the structure (de Klerk et al., 2008).

A modal domain method, common in dynamic substructuring, is the Craig-Bampton method on component mode synthesis (CMS) (Craig and Bampton, 1968). The Craig-Bampton method assumes fixed interfaces, while the structure is segmented into N_s substructures each having m DOFs. The global solution is derived from combining vibration modes of the q internal

DOFs of each substructure with static constraint modes of the b interface DOFs. The equation of dynamic equilibrium of substructure s is given in Eq. (5), where, similar to conventional modal analysis, damping is neglected.

$$\mathbf{M}_{mm}^{(s)} \ddot{\mathbf{u}}_m^{(s)} + \mathbf{K}_{mm}^{(s)} \mathbf{u}_m^{(s)} = \mathbf{F}_m \Leftrightarrow \begin{bmatrix} \mathbf{M}_{qq} & \mathbf{M}_{qb} \\ \mathbf{M}_{bq} & \mathbf{M}_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_q \\ \ddot{\mathbf{u}}_b \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{qq} & \mathbf{K}_{qb} \\ \mathbf{K}_{bq} & \mathbf{K}_{bb} \end{bmatrix} \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_b \end{bmatrix} = \begin{bmatrix} \mathbf{F}_q \\ \mathbf{F}_b \end{bmatrix} \quad (5)$$

In Eq. (5), $\mathbf{M}^{(s)}$ and $\mathbf{K}^{(s)}$ are the mass matrix and the stiffness matrix of substructure s , respectively. Upon calculating the vibration modes Φ_{qq} and the static constraint modes Φ_{qb} , the modal matrix $\Psi^{(s)}$ is created and the mass matrix $\mu^{(s)}$ and stiffness matrix $\kappa^{(s)}$ of substructure s in generalized coordinates are computed, as shown in Eq. (6) and Eq. (7).

$$\begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_b \end{bmatrix} = \begin{bmatrix} \Phi_{qq} & \Phi_{qb} \\ \mathbf{0}_{bq} & \mathbf{I}_{qb} \end{bmatrix} \Phi_{bq} = -\mathbf{K}_{qq}^{-1} \mathbf{K}_{qb} \quad (6)$$

$$\Psi^{(s)} = \begin{bmatrix} \Phi_{qq} & \Phi_{qb} \\ \mathbf{0}_{bq} & \mathbf{I}_{bb} \end{bmatrix} \quad \mu^{(s)} = (\Psi^{(s)})^T \mathbf{M}_{mm}^{(s)} \Psi^{(s)} \quad \kappa^{(s)} = (\Psi^{(s)})^T \mathbf{K}_{mm}^{(s)} \Psi^{(s)} \quad (7)$$

The synthesis for deriving the global solution follows the calculation of a transformation matrix \mathbf{S} mapping the internal DOFs and interface DOFs to global DOFs. Matrices $\mu^{(s)}$ and $\kappa^{(s)}$ formulate the synthesized block diagonal μ and κ matrices, which are transformed to matrices \mathbf{M} and \mathbf{K} . The natural frequencies ω_p and mode shapes in generalized coordinates Φ_g are derived by solving the eigenvalue problem with matrices \mathbf{M} and \mathbf{K} , as shown in Eq. (8) and Eq. (9).

$$\begin{bmatrix} \mathbf{u}_{q1} \\ \mathbf{u}_{b1} \\ \mathbf{u}_{q2} \\ \mathbf{u}_{b2} \\ \vdots \\ \mathbf{u}_{qN_s} \\ \mathbf{u}_{bN_s} \end{bmatrix} = \mathbf{S} \begin{bmatrix} \mathbf{u}_{q1} \\ \mathbf{u}_{q2} \\ \vdots \\ \mathbf{u}_{qN_s} \\ \mathbf{u}_b \end{bmatrix} \quad \mu = \begin{bmatrix} \mu^{(s1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mu^{(s2)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mu^{(sN_s)} \end{bmatrix} \quad \kappa = \begin{bmatrix} \kappa^{(s1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \kappa^{(s2)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \kappa^{(sN_s)} \end{bmatrix} \quad (8)$$

$$\mathbf{M} = \mathbf{S}^T \mu \mathbf{S} \quad \mathbf{K} = \mathbf{S}^T \kappa \mathbf{S} \quad [\mathbf{K} - \lambda_p \mathbf{M}][\Phi_g] = \mathbf{0} \quad \lambda_p = \omega_p^2 \quad (9)$$

The mode shapes Φ_g are transformed into physical coordinates, as shown in Eq. (10). Using the modal matrix Ψ of each substructure, the corresponding local mode shape is calculated.

$$\begin{bmatrix} \Phi_s^1 \\ \Phi_s^2 \\ \vdots \\ \Phi_s^{N_s} \end{bmatrix} = \begin{bmatrix} \Psi^1 \\ \Psi^2 \\ \vdots \\ \Psi^{N_s} \end{bmatrix} \begin{bmatrix} \Phi_{g1}^1 & \Phi_{g2}^1 & \cdots & \Phi_{gN_s}^1 \\ \Phi_{g1}^2 & \Phi_{g2}^2 & \cdots & \Phi_{gN_s}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{g1}^{N_s} & \Phi_{g2}^{N_s} & \cdots & \Phi_{gN_s}^{N_s} \end{bmatrix} \quad (10)$$

The implementation of the theoretical framework, discussed in this section, into a wireless SHM system is presented in the next section.

3 Implementation of the physics-based modeling approach into a wireless SHM system

The physics-based embedded modeling approach is implemented into a prototype wireless SHM system. In this section, the SHM system as well as the embedded software, which implements the proposed approach, is presented.

3.1 Design of wireless SHM system

The prototype wireless SHM system employed in this study comprises wireless sensor nodes, a server, and a base station serving as an interface between wireless sensor nodes and the server. The SHM system is designed such that several of the tasks described in the previous sections are executed on-board the sensor nodes and wireless communication is minimal. A flowchart of the tasks assigned to the SHM system is illustrated in Figure 1.

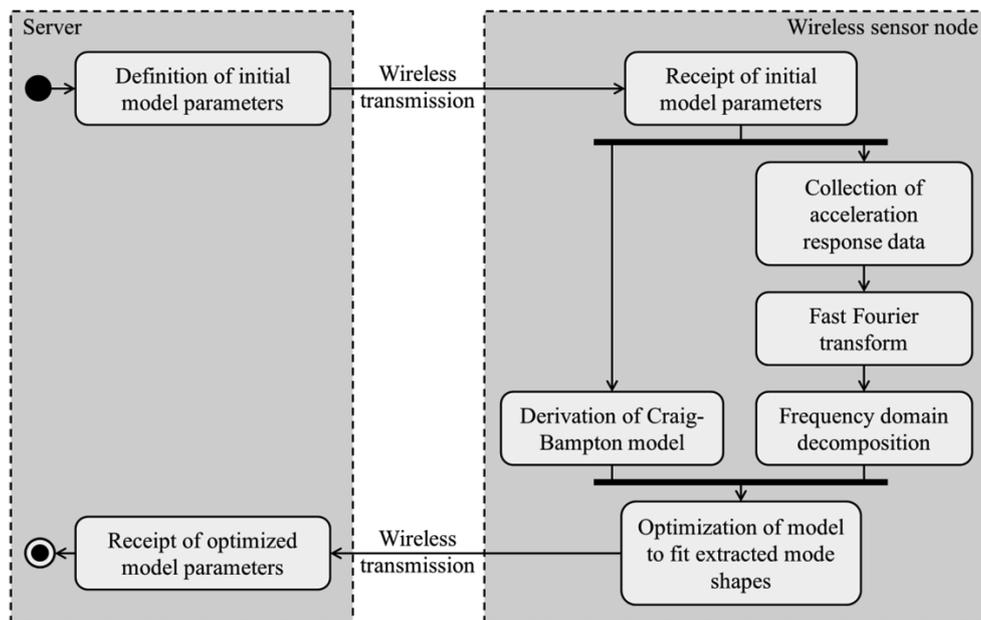


Fig. 1: Monitoring tasks of the wireless SHM system.

The FE model is created on the server, and the model parameters of each substructure are sent to the sensor nodes responsible for the substructure under consideration. Then, the sensor nodes of each substructure compute the respective Craig-Bampton parameters and the corresponding FEM local mode shapes (“Craig-Bampton model”). In parallel, each sensor node collects and transforms acceleration response data into the frequency domain, via an embedded FFT algorithm, and an embedded FDD algorithm is used to extract the local mode shapes from the collected acceleration data. Finally, based on the comparison between the extracted local

mode shapes and the FEM local mode shapes, the Craig-Bampton model (and, by extension, the FE model) is optimized.

3.2 Embedded software

The following discussion concerns the embedded software for the implementation of the proposed embedded modeling approach written in the object-oriented Java programming language. The embedded software consists of two packages, one package for the server tasks, termed “remoteServer”, and one package for the tasks of the sensor nodes, termed “sensorNode”. The information for executing the tasks described in the previous section is included in Java classes, which are used to construct the respective “objects”.

In package remoteServer, the Java classes are responsible for reading the model parameters from a text file and for communicating the model parameters wirelessly to the sensor nodes. To this end, reliable peer-to-peer wireless communication links are established between the base station and each sensor node. In addition, remoteServer package is responsible for receiving the results of the analysis. Package “sensorNode”, includes Java classes that implement the tasks assigned to the algorithms embedded into the sensor nodes (i.e. to the embedded model). The “MainNode” class handles the communication with the base station and the other sensor nodes as well as the collection of acceleration response data. The “FrequencySpectrum” class and the “FFT” class are responsible for transforming the collected acceleration response data into the frequency domain. Upon transforming the data into the frequency domain, the “FrequencyDomainDecomposition” class yields the local extracted mode shapes. The “CraigBampton” class calculates the parameters of the Craig-Bampton model. The optimization of the Craig-Bampton model is handled by the MainNode class. Finally, for matrix operations, classes from the open-source library JAMA (JAMA, 2016) are used. The sensorNode package is illustrated in Figure 2.

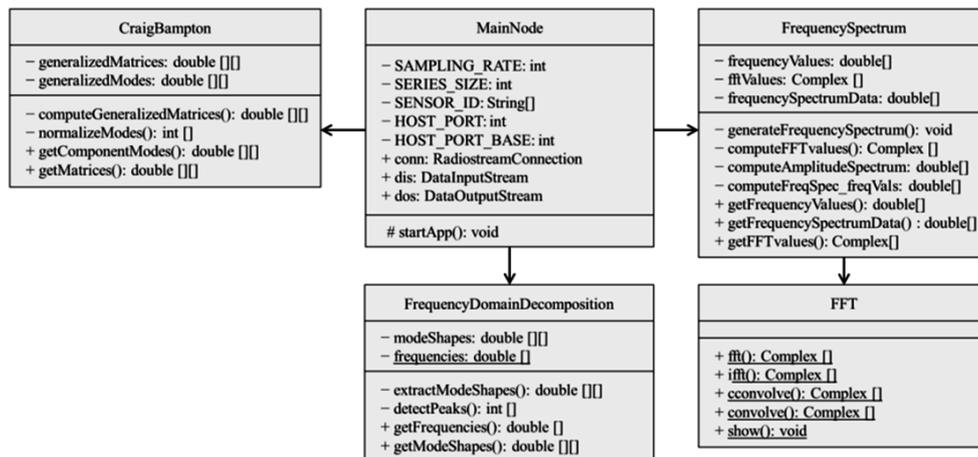


Fig. 2: The “sensorNode” package.

4 Validation test

A laboratory test for the validation of the embedded modeling approach is devised. First, the experimental setup is briefly presented. Then, the tests are described and the results are discussed.

4.1 Experimental setup

A laboratory frame structure made of steel plates of 250 mm × 500 mm × 0.8 mm (length × width × thickness), supported by M5 steel-threaded columns of circular cross sections (5 mm diameter) is used for the validation tests. The story height is 230 mm and the columns are fixed into a solid block of 400 mm × 600 mm × 300 mm. Following the dynamic substructuring method, the structure is segmented into two substructures; substructure α , including the first and the second story, and substructure β , consisting of the second, the third and the fourth story, as illustrated in Figure 3. The second story serves as the interface between the two substructures.

The wireless sensor nodes and the base station used for the experimental setup are manufactured by Oracle (Oracle Corp., 2009, 2010). Each sensor node features a Java-programmable 400 MHz ARM microprocessor, 512 kB volatile memory (RAM) used for data storage and runtime, 4 MB flash memory used for storing applications, and an IEEE 802.15.4 radio transceiver. Moreover, an MMA7455L digital output accelerometer able to sample at either 125 Hz or 250 Hz, with selectable measurement ranges of ± 2 g, ± 6 g, or ± 8 g, is embedded into each sensor node. As shown in Figure 3, one sensor node is placed at the center of each story; sensor nodes A and B are responsible for substructure α , while sensor nodes B, C, and D are responsible for substructure β .

4.2 Description of the tests

First, a FE model of the structure illustrated in Figure 3 is created. To follow the experimental setup, a 4-DOF oscillator model is selected, as shown in Figure 4.

The mass matrix of the structure is derived, by making assumptions on the material properties of the structure, while accounting for the mass of the sensor nodes. Assuming that there is no transfer of forces between adjacent stories (“shear frame” assumption), the initial stiffness matrix \mathbf{K}_0 and the mass matrix \mathbf{M} of the structure are given in Eq. (11).

$$\mathbf{K}_0 = \begin{bmatrix} 50.8 & -25.4 & 0 & 0 \\ -25.4 & 50.8 & -25.4 & 0 \\ 0 & -25.4 & 50.8 & -25.4 \\ 0 & 0 & -25.4 & 25.4 \end{bmatrix} \quad \mathbf{M} = 10^{-4} \cdot \begin{bmatrix} 8.6 & 0 & 0 & 0 \\ 0 & 8.6 & 0 & 0 \\ 0 & 0 & 8.6 & 0 \\ 0 & 0 & 0 & 8.2 \end{bmatrix} \quad (11)$$

From finite element analysis the fundamental (first) eigenfrequency of the structure is $f_{1,0} = 9.30$ Hz. The stiffness matrix and the mass matrix from Eq. (11) is divided into stiffness matrices \mathbf{K}_0^α and \mathbf{K}_0^β and mass matrices \mathbf{M}^α and \mathbf{M}^β , according to the substructures considered, as shown in Eq. (12) and Eq. (13).

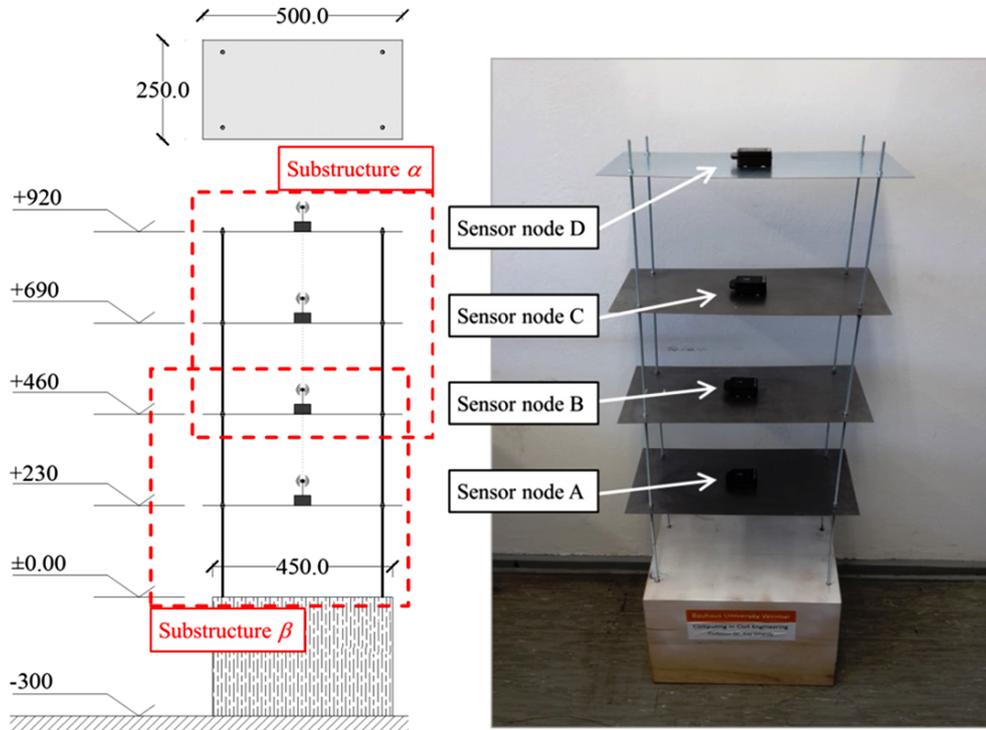


Fig. 3: Experimental setup for the validation tests.

$$\mathbf{K}_0^\alpha = \begin{bmatrix} 50.8 & -25.4 \\ -25.4 & 25.4 \end{bmatrix} \quad \mathbf{K}_0^\beta = \begin{bmatrix} 25.4 & -25.4 & 0 \\ -25.4 & 50.8 & -25.4 \\ 0 & -25.4 & 25.4 \end{bmatrix} \quad (12)$$

$$\mathbf{M}^\alpha = 10^{-4} \cdot \begin{bmatrix} 8.6 & 0 \\ 0 & 4.3 \end{bmatrix} \quad \mathbf{M}^\beta = 10^{-4} \cdot \begin{bmatrix} 8.2 & 0 & 0 \\ 0 & 8.6 & 0 \\ 0 & 0 & 4.3 \end{bmatrix} \quad (13)$$

Following the communication of the model parameters to the sensor nodes, the Craig-Bampton model is created and the FEM local mode shape is derived. Next, the structure is excited by deflecting the top story, and acceleration response data is collected by the sensor nodes, under free vibration, and transformed into the frequency domain. Then, the extracted local mode shapes are derived by applying the FDD method and compared to the FEM local mode shape. The extracted fundamental eigenfrequency is $f_{1,exp} = 2.38$ Hz. Finally, the FEM model parameters are optimized based on the comparison between the extracted local mode shape and the FEM local mode shape. The optimization is performed by multiplying the values of the stiffness matrices by a scalar, resembling variations in global model parameters (such as the modulus of elasticity), and by minimizing the residuals between the FEM local mode shapes and the extracted local mode shapes. The optimized stiffness matrices \mathbf{K}_{opt}^α and \mathbf{K}_{opt}^β for both substructures are given in Eq. (14).

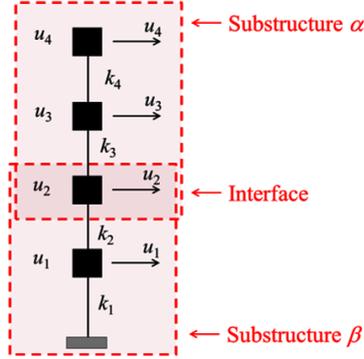


Fig. 4: FE model of the structure.

$$\mathbf{K}_{opt}^{\alpha} = \begin{bmatrix} 1.88 & -0.94 \\ -0.94 & 0.94 \end{bmatrix} \quad \mathbf{K}_{opt}^{\beta} = \begin{bmatrix} 0.95 & -0.95 & 0 \\ -0.95 & 1.90 & -0.95 \\ 0 & -0.95 & 0.95 \end{bmatrix} \quad (14)$$

The considerable deviations between the initial stiffness matrix (Eq. (11)) and the optimized stiffness matrix (Eq. (14)) are attributed to the shear frame assumption. More specifically, due to the small thickness of the plates, non-negligible rotations of plate-to-column connections are possible, thus reducing the actual stiffness of the structure. For comparison purposes, in Eq. (15) the synthesized optimized stiffness matrix is compared to the stiffness matrix \mathbf{K}_{prev} derived from a system identification methodology applied to the same laboratory structure (Dragos and Smarsly, 2015b).

$$\mathbf{K}_{opt} = \begin{bmatrix} 1.88 & -0.94 & 0 & 0 \\ -0.94 & 1.89 & -0.95 & 0 \\ 0 & -0.95 & 1.90 & -0.95 \\ 0 & 0 & -0.95 & 0.95 \end{bmatrix} \quad \mathbf{K}_{prev} = \begin{bmatrix} 2.61 & -0.99 & 0 & 0 \\ -0.99 & 2.58 & -1.30 & 0 \\ 0 & -1.30 & 2.42 & -1.11 \\ 0 & 0 & -1.30 & 1.18 \end{bmatrix} \quad (15)$$

It is clear from Eq. (15), that the physics-based modeling approach is able to accurately describe the behavior of the monitored structure. Furthermore, embedding the model into wireless sensor nodes enhances the ability of the sensor nodes to perceive the physical characteristics of the structure (in this study, the stiffness parameters).

5 Summary and conclusions

Embedded computing is an integral part of wireless structural health monitoring (SHM) systems, facilitating the assignment of monitoring tasks to wireless sensor nodes and, thus, reducing the power-consuming transmission of raw time series from wireless sensor nodes to a centralized server. While there is a broad wealth of embedded computing approaches for wireless SHM

systems in the literature, physics-based embedded models to enable wireless sensor node to perceive the physical characteristics of the monitored structure have not been adequately addressed. In this paper, a physics-based embedded modeling approach has been presented. The proposed approach is based on finite element modeling and dynamic substructuring. First, to perform the analysis in a decentralized manner by exploiting the on-board processing capabilities of wireless sensor nodes, the monitored structure is divided into substructures. Each substructure is instrumented with at least one sensor node. Then, a FE model of each substructure is created and sent to the responsible sensor nodes. Following the Craig-Bampton method on component mode synthesis, local mode shapes (FEM local mode shapes) are computed on each substructure. Next, acceleration response data is collected by the sensor nodes, and, by applying the frequency domain decomposition method, extracted local mode shapes are derived. Upon comparing the FEM local mode shapes and the extracted local mode shapes directly on the sensor nodes, the parameters of the FE model are optimized.

The proposed approach has been validated through a laboratory test on a steel frame structure. The structure has been segmented into two substructures, and an FE model of each substructure has been created. The model parameters of each substructure have been sent to the responsible sensor nodes, and the local FEM mode shapes have been derived using the Craig-Bampton method. Next, the sensor nodes have collected and transformed acceleration response data into the frequency domain. An embedded frequency domain decomposition algorithm has been employed to obtain the extracted local mode shapes. Finally, based on the comparison between the FEM local mode shapes and the extracted local mode shapes, the stiffness parameters of the FE model have been optimized. To illustrate the ability of the proposed approach to describe the behavior of the monitored structure, the optimized stiffness parameters have been compared to the stiffness parameters of the same structure estimated using a system identification method, showing a close match between the results. Therefore, it can be concluded that the physics-based embedded modeling yields an embedded model of the structure that can accurately describe the structural behavior. Moreover, embedding the model into wireless sensor nodes enhances the ability of the sensor nodes to perceive the physical characteristics (in this study, the stiffness parameters) of the monitored structure, thus facilitating the decentralized execution of monitoring tasks.

Future work will focus on extending the proposed approach to more dynamic substructuring methods. Furthermore, more elaborate FE models will be employed and the issues of damage detection and damage localization will be addressed.

Acknowledgements

Financial support of the German Research Foundation (DFG) through the Research Training Group GRK 1462 (“Evaluation of Coupled Numerical and Experimental Partial Models in Structural Engineering”) is gratefully acknowledged. Any opinions, findings, conclusions or recommendations expressed in this paper are solely those of the authors and do not necessarily reflect the views of DFG.

References

- Brincker, R., Zhang, L., and Andersen, P. (2000). „Modal identification from ambient responses using frequency domain decomposition“. In: *Proc. of the 18th International Modal Analysis Conference (IMAC)*. (San Antonio, Texas, USA, Feb. 7, 2000).
- Cooley, J. W. and Tukey, J. W. (1965). „An algorithm for the machine calculation of complex Fourier series“. In: *Mathematics of Computation* 19.90, pp. 297–301.
- Craig Jr., R. R. and Bampton, M. C. (1968). „Coupling of substructures for dynamic analyses.“ In: *AIAA Journal* 6.7, pp. 1313–1319.
- de Klerk, D., Rixen, D. J., and Voormeeren, S. (2008). „General framework for dynamic substructuring: history, review and classification of techniques“. In: *AIAA journal* 46.5, pp. 1169–1181.
- Dragos, K. and Smarsly, K. (2015a). „A comparative review of wireless sensor nodes for structural health monitoring“. In: *Proceedings of the 7th International Conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII)*. (Turin, Italy, July 1, 2015).
- Dragos, K. and Smarsly, K. (2015b). „Embedding numerical models into wireless sensor nodes for structural health monitoring“. In: *Proceedings of the 10th International Workshop on Structural Health Monitoring (IWSHM)*. (Stanford, CA, USA, Sept. 1, 2015).
- JAMA (2016). *JAMA—A Java Matrix Package*. <http://math.nist.gov/javanumerics/jama> (visited on 02/29/2016).
- Lei, Y., Shen, W. A., Song, Y., and Wang, Y. (2010). „Intelligent wireless sensors with application to the identification of structural modal parameters and steel cable forces: from the lab to the field“. In: *Advances in Civil Engineering* 2010.
- Lynch, J. P., Sundararajan, A., Law, K. H., Sohn, H., and Farrar, C. R. (2004). „Design of a wireless active sensing unit for structural health monitoring“. In: *Proceedings of SPIE’s 11th Annual International Symposium on Smart Structures and Materials*. (San Diego, CA, USA, Mar. 14, 2004).
- Oracle Corp. (2009). *Sun SPOT Theory of Operation, 1.5.0*. Santa Clara, CA, USA: Sun Labs.
- Oracle Corp. (2010). *Sun SPOT eDEMO Technical Datasheet*. 8th ed. Santa Clara, CA, USA: Sun Labs.
- Swartz, R. A., Zimmerman, A., and Lynch, J. P. (2007). „Structural health monitoring system with the latest information technologies“. In: *Proceedings of 5th Infrastructure & Environmental Management Symposium, Yamaguchi, Japan*. (Yamaguchi, Japan, Sept. 28, 2007).
- Zimmerman, A. and Lynch, J. P. (2007). „Parallelized Simulated Annealing for Model Updating in Ad-Hoc Wireless Sensing Networks“. In: *Proceedings of the International Workshop on Data Intensive Sensor Networks (DISN’07)*. (Mannheim, Germany, May 1, 2007).
- Zimmerman, A., Shiraishi, M., Swartz, R. A., and Lynch, J. P. (2008). „Automated modal parameter estimation by parallel processing within wireless monitoring systems“. In: *Journal of Infrastructure Systems* 14.1, pp. 102–113.

Mechanismen agiler Methoden

Menno van Rahden¹ und Jan Linxweiler²

*Institut für Rechnergestützte Modellierung im Bauingenieurwesen (iRMB),
Technische Universität Carolo-Wilhelmina zu Braunschweig,
Pockelsstr. 3, 38106 Braunschweig, Deutschland*

¹E-Mail: m.v-rahden@tu-braunschweig.de, ²E-Mail: j.linxweiler@tu-braunschweig.de

Abstract Dieser Artikel präsentiert die Mechanismen der agilen Softwareentwicklung. Durch eine Gegenüberstellung ausgewählter, agiler Vorgehensmodelle wie dem Extreme Programming oder auch SCRUM versucht diese Arbeit ein tiefergehendes Verständnis über die Mechanismen der agilen Praktiken zu liefern. Darüber hinaus wird die Umsetzung agiler Methoden im akademischen Forschungs- und Lehrbetrieb diskutiert.

Keywords: Vorgehensmodelle, agile Softwareentwicklung, Lean Development, Extreme Programming, SCRUM

1 Einleitung

Bauwerke und Software sind unter dem Aspekt vergleichbar, dass beide in der Regel individualisierte Produkte sind. Individualisiert meint, dass der Kunde als Besteller oder auch Nutzer die Anforderungen wie z.B. das Umfeld, die Funktionalität, Optik oder die Qualität an das Produkt definiert. Der Entwicklungsprozess ist dementsprechend von den Anforderungen des Kunden abhängig. Ändern sich die Kundenanforderungen, sollte auf diese Änderungen flexibel reagiert werden können. Ähnlich wie im Bauhauptgewerbe der Prozess der baubegleitenden Planung Einzug fand, hat die Softwareentwicklungsbranche auf diese Problematik mit agilen Vorgehensmodellen reagiert. Eine späte Änderung in den Anforderungen wird durch die der Softwareentwicklung inhärenten Möglichkeit, vergleichsweise kostengünstig Änderungen zu implementieren, in Verbindung mit der Fähigkeit agiler Vorgehensmodelle, flexibel auf Änderungen reagieren zu können, zu einem signifikanten Wettbewerbsvorteil. Agile Vorgehensmodelle berücksichtigen den Ansatz des Lean Developments. Dieser ermöglicht es Entwicklern mit möglichst wenig Ressourceneinsatz schnell, flexible und qualitativ gute Ergebnisse zu liefern. Gerade bei Unternehmensgründungen aber eben auch im akademischen Forschungs- und Lehrbetrieb, wo Entwickler mit engen Budgets und geringem Personaleinsatz konfrontiert werden, können agile Methoden unterstützend Anwendung finden.

Diese Arbeit ist das Ergebnis einer Literaturrecherche zum Thema *agile Methoden*. Aufgrund der durch die initialen Unterzeichner des *agile Manifesto*³ zunehmend bemängelten *semantischen Diffusion* und Verkommerzialisierung des „agile“-Begriffs in der letzten Dekade (Fowler,

³ <http://www.agilemanifesto.org>

2015; Thomas, 2015) sind die folgenden Abschnitte vorrangig an die jeweilige Primärliteratur angelehnt.

2 Agiles Vorgehen in der Projektdurchführung

In der Natur ist das Konzept der *Selbstorganisation* allgegenwärtig. Auch agile Projekte basieren auf Selbstorganisation (Appelo, 2011). Die Idee der prozessorientierten *bewussten Steuerung und Kontrolle* entstammen der frühen Managementlehre, ausgehend vom *Scientific Management* durch Frederick W. TAYLOR. Die selbstorganisierte Unternehmung ist indes die Grundlage der umstrittenen und jungen Lehre des *evolutionären Managements*.⁴ Selbstorganisation ist häufig negativ konnotiert, da sie chaotisch und unkontrollierbar scheint. Tatsächlich funktioniert Selbstorganisation über Verhaltensregeln, die einer kontinuierlichen, reaktiven Anpassung unterliegen – und zwar an sich verändernde Rahmenbedingungen. Damit wird die durch Veränderung entstehende Komplexität zu einer festen Konstante im Entwicklungsprozess.

Auf einer Skala von *geführter Ordnung* über *semi-kontrollierte Komplexität* zu *anarchistischem Chaos*,⁵ lassen sich die agilen Vorgehensmodelle auf einem Spektrum zwischen dem Chaos und der Komplexität einordnen (Appelo, 2011). Agile Modelle verstehen sich als eine Kombination aus Sammlungen von in der Praxis erprobten *Werten, Prinzipien und Praktiken* und kurzzyklisch gestalteten *Prozessrahmenwerken*. Auf der einen Seite versuchen agile Vorgehensmodelle den Entwicklern die Freiheit zur Selbstentfaltung nicht zu nehmen, verlangen auf der anderen Seite aber eine disziplinierte Umsetzung der vorgegebenen Praktiken. Je weniger Vorgaben ein agiles Modell einer selbstorganisierten Einheit diktiert, desto agiler ist das Modell.

Das agile Manifest stellt die Wertevorgabe für das Verhalten in agilen, selbstorganisierten Teams dar. Die Werteprämisse für agile Prozesse ist wie folgt definiert:⁶

1. *Individuen und Interaktionen* über die *verwendeten Prozesse und Werkzeuge*.
2. Ein *funktionierendes Produkt* über eine *ausführliche Dokumentation*.
3. Die *Zusammenarbeit mit dem Kunden* über *Vertragsverhandlungen*.
4. Das *Reagieren auf Veränderungen* über das *Befolgen eines konkreten Plans*.

Im Folgenden werden die in den Modellen wirkenden Mechanismen des Lean Development, Extreme Programming und Scrum diskutiert.

2.1 Lean Development

Lean Development ist der Ursprung der agilen Methoden. Es steht für einen Teil einer Ordnung von Lean-Konzepten, die jeweils auf unterschiedliche unternehmerische Bereiche wie z.B. der Produktion oder eben im Fall von Lean Development auf die Entwicklung abzielen. Die Taxonomie der Lean-Konzepte ist in Abbildung 1 dargestellt. Organisationen, die eine Lean-Philosophie verfolgen, können auch als „integrierte sozio-technische Systeme, deren Hauptziel

⁴ Vgl. Malik, 1982; Malik und Probst, 1981; Radatz, 2003

⁵ Abgeleitet aus altgriechisch *anarchia* „ohne Herrschaft“

⁶ In Anlehnung an <http://www.agilemanifesto.org> und den beteiligten Autoren.

die *Beseitigung von Verschwendungen*, bei paralleler Reduktion der lieferantenbedingten, kundenbedingten und internen Schwankungen, ist“ (Shah und Ward, 2007) beschrieben werden. Die Lean-Philosophie wurde von Toyota als flexible Antwort auf die starre amerikanische, arbeitsteilige (tayloristische, fordistische) aber dennoch sehr erfolgreiche Fließbandproduktion in der Nachkriegszeit entwickelt. Im Jahr 1962 – ca. 15 Jahre nach Entwicklungsbeginn – wurde das Konzept unter dem Namen *Toyota Production System* konzernweit umgesetzt. Die beiden Hauptpraktiken *Just-in-Time Flow* und *Autonomation*⁷ sollten die Lösung für eine ressourcenschonende und flexible Produktion sein, die im Vergleich zu dem tayloristisch-fordistischen⁸ Modell auf eine punktuelle und sich verändernde Nachfrage reagieren konnte.

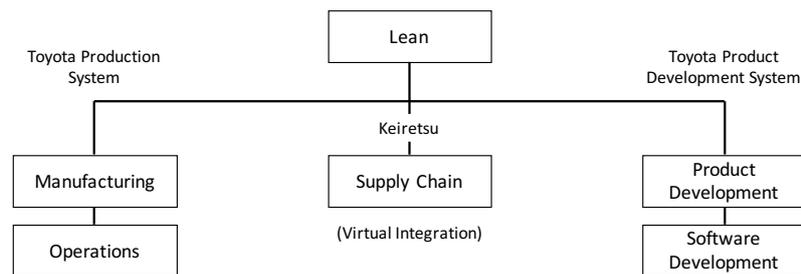


Abb. 1: Taxonomie der Lean-Konzepte (M. Poppendieck und T. Poppendieck, 2007).

Die sich verändernde Nachfrage nach Vielfalt am Automobil brachte eine neue Dimension der Komplexität im Produktionsprozess mit sich. Die Idee der Just-in-Time Produktion war seiner Zeit weit voraus und um der Komplexität Herr zu werden für Toyota der Schlüssel zum Erfolg. Erst bei Entstehen einer Nachfrage wurde der Produktionsprozess angeschoben. Daraus resultierte die Möglichkeit einer *Nichtlagerproduktion*. Zwischenprodukte mussten nicht mehr auf Vorrat produziert werden. Stattdessen liegt der Fokus auf der Produktion kleiner Inkremente. Dadurch können Engpässe schnell identifiziert und beseitigt werden. Dies setzt voraus, dass die Maschinen so konfigurierbar sein mussten, dass ein schneller Wechsel zwischen den Produktvarianten problemlos und schnell möglich war.

Die Praktik der Autonomation ist ein unverzichtbarer Zusatz für den Just-in-Time Flow. Das Konzept geht davon aus, dass die Maschinen so entworfen sind, dass sie durch verschiedene Tests Unregelmäßigkeiten am Material, am Produkt oder ihrer Arbeitsweise *autonom* und *automatisiert* erkennen können. In einem solchen Fehlerfall wird die Produktion durch die Maschine automatisch pausiert und *Feedback* zur Fehlerherkunft an einen Maschinenaufseher gegeben. So kann ein Problem schnell bis zu seiner *Wurzel* verfolgt und nachhaltig behoben werden. Im Anschluss kann die Arbeit von der Maschine wieder fehlerfrei aufgenommen werden. Dieses Konzept ist auch als *Stop-the-Line* bekannt und führt zu einer *Nullinspektionsumgebung*, da abgesehen von regulären Wartungsarbeiten erst bei einem Fehlerfall inspiziert werden muss. Die automatisierten Tests sind eine Absicherung gegen Fehler im Produktionsprozess. Sie verringern den Ausschuss, die Personalkosten und erhöhen signifikant und nachhaltig die *Qualität* des Produktionsprozesses, des Produkts und seiner Weiterentwicklung.

Wie aus dem obigen Zitat von Shah und Ward, 2007 deutlich wird, geht es um die *Beseitigung von Verschwendungen*. Dies entspricht auch dem ersten Lean-Prinzip. Dabei sind Verschwendungen alle Zustände und Tätigkeiten, die die Wertschöpfung eines Produktes (oder

⁷ Der Begriff der Autonomation ist ein Neologismus, bestehend aus *autonom* und *Automatisierung* (en: Automation).

⁸ Taylorismus ist ein Modell der Arbeitsteilung, das eine klare Trennung von Arbeitsprozessen und Managementprozessen und eine starke Stückelung der Arbeitsprozesse in sich vorsieht. Fordismus knüpft daran an und beschreibt die arbeitsteilige Fließbandproduktion.

Unternehmens) negativ beeinflussen. Auf Vorrat produzierte Zwischenprodukte sind – ebenso wie unnötiger Ausschuss, überhöhte Personalkosten oder schwerfällige Weiterentwicklungsmöglichkeiten des Produktionsprozesses – Verschwendungen. Im Folgenden sind die sieben Lean-Prinzipien aufgelistet: (M. Poppendieck und T. Poppendieck, 2007)

1. Eliminiere Verschwendung
2. Baue Qualität (von Beginn an) ein
3. Entwickle Wissen
4. Liefere schnell aus
5. Verschiebe (irreversible) Entscheidungsbindungen
6. Respektiere Menschen
7. Optimierte das Ganze

Der Toyotismus dreht im Vergleich zum Taylorismus die Verantwortungen um. Anstatt wie im tayloristischen Modell die Verantwortungen nur auf das Management zu reduzieren, bekommen alle Beteiligten wieder mehr Gestaltungsfreiheit am Entwicklungs- und Produktionsprozess (6). Die oberste Priorität ist es dabei Ineffizienzen zu lösen (1) um eine hohe Qualität zu erreichen (2). Eine hohe Qualität bedeutet schnellere Auslieferungszyklen (4). Im Optimierungsgeschehen ist es notwendig, dass nicht nur einzelne Prozesse als Blackbox betrachtet werden, sondern auch die Auswirkungen auf das organisatorische und interorganisatorische Gesamtsystem berücksichtigt werden (7). Während der Eliminierung von Verschwendungen oder Durchführung experimenteller Projekte, ist das daraus Erlernte durch ein entsprechendes Wissensmanagementsystem in Kurzform zu dokumentieren, sodass organisationsweit auf dieses Wissen zugegriffen werden kann (3). Die Bindung an Entscheidungen sollte auf den kritischen Moment verlagert werden, sodass ein Maximum an Informationen zu diesem Zeitpunkt vorliegt (5). (M. Poppendieck und T. Poppendieck, 2007)

Für den Lean-Ansatz spielen *Feedback* und *Reflexion* eine wichtige Schlüsselrolle. Erst durch das *empirische Feedback* aus den automatisierten Tests der Automation können Fehler-Ursachen-Analysen angestoßen werden. Das aus einer Fehler-Ursachen-Analyse stammende Feedback kann idealerweise zur Definition neuer Spezifikationen für neue Tests genutzt werden. So kann ein solcher oder auch verwandter Fehler in Zukunft vermieden werden. Das erlangte Wissen über den Mangel kann über ein geeignetes Wissensmanagementsystem der Organisation zur Verfügung gestellt werden. So kann die *technische Schuld* (Cohn, 2010) nachhaltig gering gehalten werden.

2.2 Extreme Programming

Die von Kent BECK 1996 entwickelte und damals sehr revolutionäre Entwicklungsmethodik Extreme Programming (XP) ist bis heute eine der drei populärsten Entwicklungspraktiken aus der Familie der agilen Entwicklungsansätze im Bereich der Softwareentwicklung. (Komus und GMP, 2014) Während allerdings andere agile Entwicklungsmethodiken relativ allgemein gehalten sind und sich somit auch auf Projekte anderer Branchen anwenden lassen, ist XP rein für den Einsatz in der Softwareentwicklung ausgelegt.

XP ist eine Sammlung einfacher, konkreter Praktiken, die kombiniert in einem agilen Softwareentwicklungsprozess aufgehen. Die *Praktiken* die unter dem Dach von XP zusammengeführt werden, behandeln sowohl die technische als auch die Managementsicht. Sie sind das Resultat einer von BECK definiert Reihe von *Werten* und *Prinzipien* und haben sich sowohl einzeln als auch kombiniert in der Praxis bewährt.

Der reine Entwicklungsprozess ist in XP nach dem Pull-Modell vorgesehen. Anders als im Push-Modell determiniert nicht die Spezifikation das Design, das Design den Code und der Code die Tests. Nach dem Pull-Modell werden die User Stories unmittelbar vor ihrer Implementierung im Detail spezifiziert. Aus dieser Spezifikation werden die Tests abgeleitet. Das Programmierschnittstellendesign wird passend zu den Anforderungen der Tests entworfen. Nach den Tests wird der Code passend zum Design der Programmierschnittstelle und den Tests geschrieben. Die sich aus dem Code ergebenden Anforderungen und Erkenntnisse müssen in das Schnittstellendesign zurückgeführt werden. Hieraus lässt sich die nächste zu spezifizierende Story ermitteln (Beck, 2005). In Abbildung 2 sind Push- und Pull-Modell gegenüber gestellt.

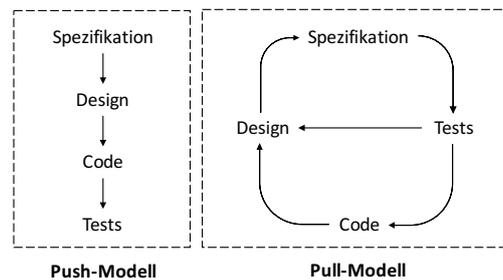


Abb. 2: Push- vs. Pull-Modell. Eigene Darstellung in Anlehnung an Beck, 2005.

Das Rahmenwerk von XP ist dahingehend vorteilhaft, dass die Praktiken schrittweise etabliert werden können. So kann vom aktuellen Tagesgeschäft stetig zu XP migriert werden. Der Erfolg von XP liegt in

- einer gesteigerten Reaktionsfreudigkeit, sichergestellt durch kurze Zyklen und schnelles Feedback,
- dem kontinuierlichen Lernverhalten durch entsprechende Reflexion der getanen Arbeit,
- der sich fortlaufend steigernden Leistungsfähigkeit des Teams mittels Routinisierung vieler Tätigkeiten und der einhergehenden Habituation an verschiedenste vorher für komplex empfundene Abläufe und der damit verbundenen Reduktion von Angst gegenüber eben diesen Prozessen
- sowie einer konkurrenzfreien, gemeinschaftlichen Zusammenarbeit zwischen dem Kunden und den Entwicklern.

Wie in Lean Development sind auch in Extreme Programming *Feedback* und *Reflexion* die treibenden Mechanismen zur Qualitätsverbesserung. Durch kurze Entwicklungszyklen, implementationsnahes Spezifikationsdesign, Stand-Up-Besprechungen, die testgetriebene Entwicklung und der paarweisen Programmierung in Verbindung mit dem Refactoring des Codes entstehen Feedbackzyklen. Aus diesen Feedbacks ergeben sich fortlaufend neue Impulse zur Verbesserung der internen und externen Qualität der Anwendung und zur Verbesserung des tagtäglichen Entwicklungsgeschehens. In Abbildung 3 sind die Feedbackzyklen von XP dargestellt.

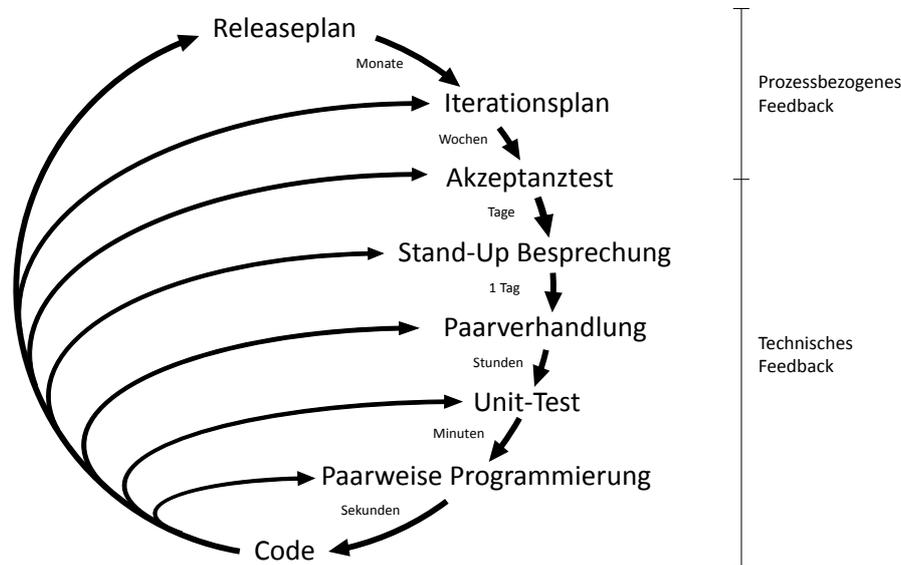


Abb. 3: Planungs- und Feedbackschleifen von XP. Eigene Darstellung in Anlehnung an Wells, 2001.

2.3 Scrum

Scrum ist ein Vorgehensmodell, das aus den Prinzipien des Lean-Managements gewachsen ist. Die fundamentalen Bestandteile von Scrum sind von den Japanern Hirotaka Takeuchi und Ikujiro Nonaka in dem 1986 veröffentlichten Paper „The New New Product Development Game“ veröffentlicht worden. Takeuchi und Nonaka betrachteten dafür die weltweit innovativsten und produktivsten Unternehmen, wie z.B. Honda, Fuji-Xerox, 3M oder Hewlett-Packard. Das Papier zielte darauf ab, innovationsfördernde, leichtgewichtige Praktiken für Teams zu schaffen. Die beiden Japaner verglichen das Vorgehen der funktionierenden Teams mit dem eines Rugby Scrums. „Der Ball wird innerhalb des Teams zugepasst, während es wie eine Einheit das Spielfeld hoch läuft.“ (Takeuchi und Nonaka, 1986)

Ken Schwaber und Jeff Sutherland sammelten unabhängig voneinander ab 1990 Erfahrungen mit diesen leichtgewichtigen Entwicklungsmethoden. Sie entschieden sich 1993 die Inhalte des Artikels, die sich hauptsächlich mit Entwicklungsprozessen aus Teams des industriellen Maschinenbaus befassten, auf Softwareprojekte anzuwenden. 1995 stellten sie dann in dem Papier „The SCRUM Development Process“ die erste formalisierte Version von Scrum vor. Schwaber und Sutherland gehören zu den Mitbegründern des *agile Manifesto*. Im Gegensatz zu XP ist Scrum ein Vorgehensmodell, das ausschließlich den Fokus auf Managementprozesse richtet. Das bedeutet, Scrum kann nicht nur in der Softwareentwicklung zur Anwendung kommen, sondern prinzipiell überall (wo selbstorganisierte Teams gewünscht sind) (Sutherland, 2014). Die zentralen Elemente von Scrum umfassen die Rollen im Team, die prozessbezogenen Werkzeuge und den iterativen Workflow. In Abbildung 4 ist der übergeordnete Auslieferungszyklus von Scrum dargestellt.

Scrum ist ein sehr allgemein gehaltenes Managementrahmenwerk, das sein Potenzial vor allem in Projekten aus den Bereichen innovativer Tätigkeiten wie der Forschung und Entwicklung voll entfalten kann. In diesen Bereichen ist es notwendig, kurzfristig auf sich ändernde

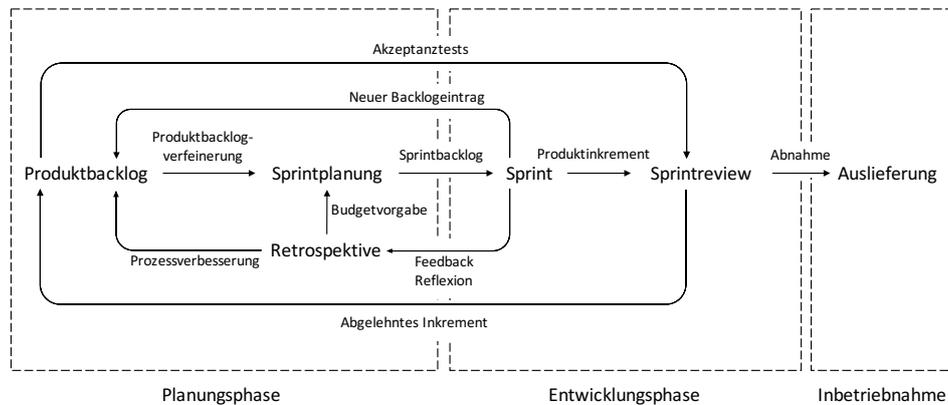


Abb. 4: Ablauf eines Scrum Auslieferungszyklus. Eigene Darstellung.

Rahmenbedingungen reagieren zu können, z.B. die sich ändernde Vision des Kunden. Die zentralen Elemente von Scrum sind

- die jeweiligen Backlogs⁹, die für ein hohes Maß an Transparenz und Disziplin sorgen,
- die täglichen Stand-Up Besprechungen, die es dem Scrum Master ermöglichen kurzfristig Hindernisse zu beseitigen
- sowie die Retrospektive, in der das Team durch *Feedback* und *Reflexion* an Leistung gewinnt und für folgende Iterationen selbst Verbesserungen vereinbart und in den Workflow einarbeitet.

Ein Blick auf Abbildung 4 verdeutlicht, dass es Feedback-Rückkopplungen aus der Entwicklungsphase in die Planungsphase gibt. Diese Rückkopplungen stellen sicher, dass anknüpfende Iterationen auf dem Stand der aktuellen Iteration aufbauen und machen es möglich auf Änderungen kurzfristig reagieren zu können.

3 Agile Praktiken im akademischen Lehrbetrieb

Die Bildungsforschung zeigt, dass der traditionelle Mix aus Frontallehre und Einzelarbeit, wie er überwiegend an akademischen Einrichtungen praktiziert wird, keine optimale Art der Wissensvermittlung darstellt. Ein kooperativer Unterricht soll effektvoller sein (Zierer, 2014). In diesem Zusammenhang wird folgend die Anwendbarkeit von agilen Methoden im akademischen Lehrbetrieb diskutiert. Neben dem Vorteil, dass der traditionelle, frontale Lehrbetrieb aus Sicht der Lehrenden sehr effizient ist, da mit möglichst geringem Ressourceneinsatz ein breites Publikum gezielt erreicht werden kann, steht dieser Ansatz aber auch in der Kritik.

Gudjons, 2006 nennt zehn Kritikpunkte der Frontallehre, die sich durch eine dominierende Rolle des Lehrenden auszeichnet. Im Folgenden sind die wichtigsten, den Lernerfolg und Lehrer-SchülerInnen-Interaktion betreffenden, Argumente aufgelistet:

1. Das Lehren des Lehrenden ist nicht gleichzusetzen mit dem Lernen der Lernenden.

⁹ Instrument zur Priorisierung von Teilaufgaben und zur Verbesserung der Selbsteinschätzung mittels Soll-Ist-Vergleichen.

2. Vernachlässigung sozialer Ziele, wie Kooperationsfähigkeit, Selbstständigkeit, Kommunikation und Verantwortung für einen gemeinsamen Arbeits- und Lernprozess.
3. Verstärkung der Bindung an die Autorität des Lehrenden führt zu undemokratischem Unterricht und Unterdrückung der SchülerInnen.
4. Vernachlässigung individueller Lernfortschritte durch Lernen im Gleichschritt.
5. Passive Informationsaufnahme der SchülerInnen. Selbstständigkeit und Eigenständigkeit werden vernachlässigt.
6. Frontalunterricht bedeutet Kontrolle für den Lehrenden. Chaos und Ineffektivität bei Gruppenarbeiten werden nicht riskiert.

Die Einzelarbeit bringt ähnliche Kritik wie die Frontallehre mit sich. Kritikpunkte sind überwiegend die Vernachlässigung sozialer Komponenten, wie z.B. der Teamfähigkeit und des sozialen Lernens. Menschen mit einer Konzentrationsschwäche werden in längeren Lernphasen überfordert. Häufig wird auch bei der Kombination aus Frontallehre und Einzelarbeit übersehen, dass nur noch diejenigen Lernenden Leistungen erbringen können, die der Vorlesung ausgiebig folgen konnten (Reich, 2015).

Einen tiefgehenden Einblick brachte eine von Hattie, 2009 veröffentlichte Studie, die für viel Aufsehen im Bildungswesen gesorgt hat. Die Studie *Visible Learning* ist eine Synthese aus über 800 Meta-Studien aus 15 Jahren Forschung und fasst die Ergebnisse von mehr als 80.000 kleineren Studien mit über 250 Mio. involvierten Studierenden zusammen. Mit der Studie sollen die Einflüsse verschiedener Faktoren auf den Lernerfolg ermittelt werden. Die Faktoren decken mehrere Bezugsdomänen ab. Darunter fallen Faktoren die Lehrperson, der Unterricht, die Lernenden, die Einrichtung, das Elternhaus und den Lebenslauf betreffend. Zu erwähnen ist, dass die Faktoren in globalen Abhängigkeiten zueinander stehen und dementsprechend keine selektive Bewertung angewandt werden sollte, um daraus „das eine Kochrezept“ für gute Lehre zu erstellen. Vielmehr können Tendenzen abgeleitet werden, wie guter Unterricht aussehen sollte. Denn wie erfolgreich der Unterricht ist, hängt am Ende von weitaus mehr als der reinen Konzeption des Unterrichts ab. In Abbildung 5 sind die ausgewählten Faktoren gemeinsam mit ihrer Effektstärke dargestellt. Dabei tragen alle Faktoren mit Effektstärken ≥ 0.00 positiv zum Lernerfolg bei. Je größer der Wert, desto besser sein Beitrag. Alle Effektstärken ≥ 0.40 sind nach HATTIE erwünschte Effekte (Zierer, 2014).

Die Faktoren *Reziprokes Lernen*, *Feedback* und *Selbsteinschätzung des eigenen Lernniveaus* haben besonders großen Einfluss auf den Lernerfolg. Wohingegen die *Fachkompetenz* der Lehrkraft einen verschwindend geringen Einfluss hat. Als wesentlich wichtiger zeigen sich die sozialen Komponenten der Lehrkraft, dessen Selbstreflexion und Optimierung des Lehrverhaltens (Micro-Teaching). Erkennbar wird auch, dass *rhythmisiertes* und *kooperatives* Lernen, eine *metakognitive Strategie* und *Lernziele* förderlich sind.

Aus der Studie von HATTIE lassen sich einige Parallelen zwischen agilen Methoden und der kooperativen Lehre ableiten. In der Lehre geht es nicht mehr nur darum, den Lehrstoff an die Masse zu bringen, sondern vielmehr legt *Visible Learning* besonderen Wert auf eine aktive Interaktion zwischen Lehrkraft und den Studenten. Ein Lehrplan sollte nicht „abgefrühstückt“ werden, sondern auch dem Tempo und den Bedürfnissen der Studenten angepasst werden. Ebenso messen die agilen Methoden einer engen Interaktion zwischen dem Kunden und den Entwicklern und den Entwicklern untereinander einen hohen Wert bei. Auch hier gibt das Entwicklerteam auf Basis seiner Selbsteinschätzung an, wieviele Features pro Sprint (siehe Abb. 4) umgesetzt werden können – und gibt damit die Geschwindigkeit vor.

Faktor	Effektstärke
Lernende	
Selbsteinschätzung des eigenen Leistungsniveaus	1,44
Vorausgehendes Leistungsniveau	0,67
Konzentration, Ausdauer und Engagement	0,48
Motivation	0,48
Schule/Einrichtung	
Klassenführung	0,52
Lernen in Kleingruppen	0,49
Zusatzangebote für Hochbegabte	0,39
Curriculum	
Kreativitätsförderung	0,65
Leseförderung	0,50
Spielförderung	0,50
Förderung der Sozialkompetenz	0,40
Lehrperson	
Micro-Teaching (Lehrverhaltenstraining)	0,88
Klarheit der Lehrperson	0,75
Lehrer-Schüler-Beziehung	0,72
Nichtetikettieren von Lernenden	0,61
Qualität der Lehrperson (aus Schülersicht)	0,44
Fachkompetenz	0,09
Unterricht	
Formative Evaluation des Unterrichts	0,90
Reziprokes Lernen (Dialog)	0,74
Feedback	0,73
Rhythmisiertes vs. Geballtes Unterrichten	0,71
Metakognitive Strategien	0,69
Lautes Denken	0,64
Problemlösen	0,61
Kooperatives vs. Individuelles Lernen	0,59
Direkte Instruktion	0,59
Mastery-Learning	0,58
Fallbeispiele	0,57
Ziele	0,56
Peer-Tutoring	0,55
Kooperatives vs. Kompetitives Lernen	0,54
Interaktive Lernvideos	0,52
Fragenstellen	0,46
Kooperatives Lernen	0,41
Ergänzende Materialien	0,37
Forschendes Lernen	0,31

Abb. 5: Auszug einiger Faktoren aus *Visible Learning* und ihre Effektstärke. Eigene Darstellung in Anlehnung an (Zierer, 2014).

In einer anwendungsorientierten Disziplin, wie der Softwareentwicklung, sollten Wissen und Qualität von Beginn an eine Rolle spielen. Das anfänglich erlernte Wissen über Qualität determiniert den späteren Lernerfolg. Denn früh umgesetzte Konventionen sorgen für mehr Klarheit im Quellcode und einer besseren Verständlichkeit. Zwei der Lean-Prinzipien lauten, Wissen entwickeln (Prinzip 3) und Qualität von Beginn an einbauen (Prinzip 2). Agile Methoden, die diese Prinzipien verkörpern und auf sozialen Mechanismen beruhen, können dazu beitragen, die Lehre zu verbessern. Diese Erkenntnis wird von verschiedenen Studien, über den Versuch agiler Methoden in der Lehre anzuwenden, geteilt.¹⁰

¹⁰ Vgl. Astrachan et al., 2002; Gondim et al., 2011; Keefe et al., 2006; Müller, 2004; Sanders, 2002

4 Agile Praktiken zur Organisation des Forschungsbetriebes

Forschungsprojekte können je nach Ziel der Projektträger und Komplexität des Forschungsgebietes einfache bis komplexe Projektorganisationen annehmen. Dabei können Projekte im Alleingang sowie in Kooperationen von verschiedenen akademischen Instituten oder privatwirtschaftlichen Entitäten organisiert sein. Eine Kooperation zeichnet sich durch einen arbeitsteiligen, partnerschaftlichen Verbund von einzelnen Personen oder Gemeinschaften aus, um eine Steigerung des eigenen Potenzials im Hinblick auf ein bestimmtes Ziel zu erreichen (Gabler, 2016). Dabei kann eine Kooperation verschiedene Formen annehmen. Abbildung 6 zeigt einen Morphologischen Kasten der verschiedenen Merkmale der organisationalen Kooperationen. Man unterscheidet im Hinblick auf die Richtung, Ausdehnung, Bindungsintensität, Verbindlichkeit, Zeitdauer, Zielidentität und die betroffenen unternehmerischen Bereiche wie z.B. Forschung und Entwicklung (F&E), Einkauf oder Vertrieb.

In der Forschungsorganisation werden Landesstellen zur quasi-freien Forschung angeboten. Diese Stellen praktizieren häufig Forschung im Alleingang. Die betreffenden Stellen unterliegen der Aufsicht der jeweiligen Institutsleitung und werden über Landesmittel finanziert. Weiterhin können wissenschaftliche Mitarbeiter an Projekten beteiligt werden. Diese Projekte haben i.d.R. eine horizontale oder vertikale Ausrichtung und betreffen somit entweder ein gemeinsames Fachgebiet oder sind interdisziplinär ausgelegt. Die Ausdehnung kann einerseits instituts- (lokal) oder universitätsintern (regional), andererseits aber auch einrichtungs- (national) und auch länderübergreifend (global) stattfinden. Bei den letzteren drei Ausdehnungen kommt es zur Problematik von *verteilten Teams*.

Merkmal	Ausprägung					
Richtung	horizontal			vertikal		diagonal
Ausdehnung	lokal	regional	national	global		
Bindungsintensität	gering		moderat		hoch	
Verbindlichkeit	Absprache		Vertrag		Kapitalbeteiligung	
Zeitdauer	temporär			unbegrenzt		
Zielidentität	redistributiv			reziprok		
Untern. Bereiche	F&E	Vertrieb	Einkauf	Marketing	Produktion	Sonstige

Abb. 6: Morphologischer Kasten von Kooperationsformen. Eigene Darstellung in Anlehnung an (Killich, 2011)

In Forschungsk Kooperationen ist es sinnvoll, einen Prozessablauf umzusetzen, der Feedbackzyklen auf unterschiedlichen Ebenen beinhaltet. Dabei kann ein Vorgehen nach dem Kanban¹¹-Ansatz – der einen kontinuierlichen Fluss erzeugt – oder eben ein iteratives Vorgehen vergleichbar dem von Scrum gewählt werden. Der Vorteil eines iterativen Vorgehens besteht vor allem darin, eine mögliche kooperationsweite Synchronisation der Auslieferungszyklen der Teilprojekte. Ein weiterer Vorzug ist eine einfachere Messbarkeit der Inputs und Outputs eines Sprints und einer damit einhergehenden Prozessverbesserung und Selbsteinschätzung. Der Vorteil eines Vorgehens nach dem Kanban-Ansatz wäre beispielsweise eine kooperationsweite, kontinuierliche, zyklunabhängige Auslieferung und Integration der Ergebnisse.

Beteiligte einzelner Teilprojekte arbeiten idealerweise raumnah, um kurze Kommunikations- und Feedbackwege zu erhalten. Einen Sonderfall stellen damit die verteilten Teams in der organisationalen Kooperation dar. Durch den Einsatz von Werkzeugen kann kooperationsweit für ein hohes Maß an Transparenz gesorgt werden. Es sollten beispielsweise sinnvolle Instrumente

¹¹ Iterationsfreies Vorgehen nach dem Pull-Prinzip.

zur Kommunikation herangezogen werden. Dabei können Tools zur formlosen Kommunikation durchaus Sinn machen. Sinnvoll ist außerdem der Einsatz von Versionskontrollsystemen wie z.B. Git für alle Dokumente. Ein geeignetes Instrument des Wissensmanagements wie ein Blog oder Wiki, wo Forschungsergebnisse und sonstige Erkenntnisse informell geteilt werden können, kann gegen personelle Risiken absichern und implizites Wissen einzelner Mitarbeiter abfangen. Werkzeuge zur Organisation von Verantwortungen, wie z.B. ein Backlog können genutzt werden, um teamübergreifende Verantwortlichkeiten je Auslieferungszyklus zu klären.

Einen weiteren Sonderfall stellt die Einzelarbeit bei der Anwendung von agilen Praktiken in der Forschung dar. Hier kommt es zu einer starken Einschränkung in der Anwendbarkeit, da die meisten Praktiken auf ein gemeinschaftliches Arbeiten ausgelegt sind. Nichts desto trotz sollten Praktiken, die die Transparenz fördern, diszipliniert umgesetzt werden. Dazu zählt z.B. die Verwendung eines Backlogs.

In einem Forschungsprojekt, das womöglich einen vertikalen Ansatz mit verteilten Teams wählt, ist es notwendig, die Prozesse so abzustimmen, dass die Entstehung von *empirischem Feedback* ermöglicht wird. Weiterhin sollte dafür gesorgt werden, dass eine einfache und bestmögliche Vernetzung der Teams möglich ist, sodass das entstehende Feedback auch vom Sender an den Empfänger übertragen werden kann. So kann einerseits für eine gute technische und andererseits für eine gute prozessbezogene Qualität gesorgt werden.

5 Fazit

Agile Vorgehensmodelle sind Rahmenwerke, um einen Entwicklungsprozess in eine disziplinierte Form der Selbstorganisation einzubetten. Selbstorganisation funktioniert nur durch stetige Verhaltensanpassungen an sich ändernde Rahmenbedingungen.

Dabei spielen Feedback und Reflexion eine wichtige Rolle um qualitative Verbesserungen sowohl aus technischer als auch aus prozessbezogener Hinsicht zu erwirken. Empirisches Feedback ist dabei der Schlüssel zum Erfolg.

Je mehr Impulse ein Entwickler oder ein Entwicklungsteam durch Feedback über ein System erhält, desto öfter kann er seine Annahmen mit der Realität abgleichen (Freeman und Pryce, 2010). Reflektierte Auswertung führt zu einem verstärkten Verständnis eines Systems. Technische Prozesse, wie auch organisatorische Prozesse, sollten entsprechend immer so gestaltet sein, dass sie kontinuierliches Feedback erzeugen.

Literatur

- Appelo, J. (2011). *Management 3.0: Leading Agile Developers, Developing Agile Leaders*. Addison-Wesley.
- Astrachan, O., Duvall, R. C. und Wallingford, E. (2002). „Bringing Extreme Programming to the Classroom“. In: *XP Perspectives*, S. 237–250.
- Beck, K. (2005). *Extreme Programming Explained: Embrace Change*. 2. Aufl. Addison-Wesley Professional.
- Beck, K. (2015). *Extreme Programming 20 Years Later*. Vortrag beim Lean IT Summit 2015. <https://www.youtube.com/watch?v=cGuTmOUdFbo> (besucht am 06/2016).
- Cohn, M. (2010). *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley.

Mechanismen agiler Methoden

- Fowler, M. (2015). *Explaining Agile - Martin Fowler and Neil Ford at USI*. Vortrag beim USI-Events 2013. <https://www.youtube.com/watch?v=GE6lbPLEAzc> (besucht am 06/2016).
- Freeman, S. und Pryce, N. (2010). *Growing-Object Oriented Software, Guided by Tests*. Addison-Wesley.
- Gabler (2016). *Kooperation*. Wirtschaftslexikon. <http://wirtschaftslexikon.gabler.de/Archiv/7992/kooperation-v11.html> (besucht am 06/2016).
- Gondim, H. W., Ambrósio, A. P. und Costa, F. M. (2011). „TaskBoard– Using XP to Implement Problem-Based Learning in an Introductory Programming Course“. In: *Agile Processes in Software Engineering and Extreme Programming*. Springer, S. 162–175.
- Gudjons, H. (2006). *Methodik zum Anfassien: Unterrichten jenseits von Routinen*. Verlag Julius Klinkhardt.
- Hattie, J. (2009). *Visible Learning: A Synthesis of Meta-Analysis relating to Achievement*. Taylor & Francis Group.
- Keefe, K., Sheard, J. und Dick, M. (2006). „Adopting XP practices for teaching object oriented programming“. In: *Proceedings of the 8th Australasian Conference on Computing Education*. Bd. 52. Australian Computer Society, Inc., S. 91–100.
- Killich, S. (2011). „Formen der Unternehmenskooperation“. In: *Netzwerkmanagement: Mit Kooperationen zum Unternehmenserfolg*. 3. Aufl., S. 13–22.
- Komus, A. und GMP (2014). *Study Status Quo Agile 2014*. BPM-Laboratory, Koblenz University of Applied Science.
- Malik, F. (1982). „Evolutionäres Management: Eine Replik zur Kritik von Karl Sandner“. In: *Die Unternehmung* 36.2, S. 91–106.
- Malik, F. und Probst, G. (1981). „Evolutionäres Management“. In: *Die Unternehmung* 35, S. 121–149.
- Müller, R. A. (2004). „Extreme Programming in a university project“. In: *Extreme Programming and Agile Processes in Software Engineering*, S. 284–293.
- Poppendieck, M. und Poppendieck, T. (2007). *Implementing Lean Software Development*. Addison-Wesley.
- Radatz, S. (2003). *Evolutionäres Management: Antworten auf die Management- und Führungsherausforderungen des 21. Jahrhunderts*. Verlag Systemisches Management.
- Reich, K. (2015). *Methodenpool*. Einzelarbeit, Beitragssammlung. <http://methodenpool.uni-koeln.de> (besucht am 07/2016).
- Sanders, D. (2002). „Extreme Programming and the Software Design Course“. In: *XP Perspectives*, S. 261–272.
- Shah, R. und Ward, P. T. (2007). „Defining and developing measures of lean production“. In: *Journal of operations management* 25.4, S. 785–805.
- Sutherland, J. (2014). *SCRUM: The Art of Doing Twice the Work in Half the Time*. Crown Publishing Group.
- Takeuchi, H. und Nonaka, I. (1986). „The new new product development game“. In: *Harvard Business Review* 64.1, S. 285–305.
- Thomas, D. (2015). *Agile is Dead • Pragmatic Dave Thomas*. Vortrag beim GOTO Amsterdam 2015. <https://www.youtube.com/watch?v=a-BOSpxYJ9M> (besucht am 06/2016).
- Wells, J. D. (2001). *Extreme Programming*. <http://www.extremeprogramming.org> (besucht am 05/2016).
- Zierer, K. (2014). *Kernbotschaften von John Hatties Visible Learning*. Konrad Adenauer Stiftung.

Entwicklung nachhaltiger Software mit Hilfe des objektorientierten Designs

Janis Zisenis¹ und Jan Linxweiler²

*Institut für rechnergestützte Modellierung im Bauingenieurwesen,
Pockelsstraße 3, 38106 Braunschweig*

¹E-Mail: janis.zisenis@tu-braunschweig.de, ²E-Mail: j.linxweiler@tu-braunschweig.de

Abstract Dieser Artikel stellt ein dynamisches Applikationsframework zur Berechnung von hydrodynamischen und hydrostatischen Problemen im Bereich des Küsteningenieurwesens vor. Die Berechnungsgrundlagen im Küsteningenieurwesen bilden komplexe empirische Formeln, die mithilfe dieser Applikation automatisiert gelöst und flexibel kombiniert werden können. Zu diesen Berechnungen gehören unter anderem die Auswertung der Linearen Wellentheorie sowie die Berechnung von Kräften aus Wellen- und Wasserströmungen auf Bauwerke und die Gewässersohle. Da es sich im Küsteningenieurwesen um einen dynamischen Bereich handelt, in dem kontinuierlich neue Berechnungsansätze verfolgt werden, ist es von besonderer Bedeutung die Applikation flexibel und erweiterbar zu gestalten. Dadurch wird sichergestellt, dass die Applikation jederzeit an den neuesten Stand der Wissenschaft angepasst werden kann. Um diese Anforderungen zu gewährleisten, wurde die Applikation nach den Methoden und Prinzipien des objektorientierten Designs entwickelt, die eine hoch kohäsive Struktur der Software mit gleichzeitig geringer Kopplung entstehen lassen (Yourdon, 1979). Anhand des Applikationsframeworks konnte gezeigt werden, dass sich durch ODD nachhaltige Software entwickeln lässt, die flexibel an neue Spezifikationen angepasst werden kann.

Keywords: Objektorientiertes Design, Software Design, SOLID, Softwarearchitektur

1 Einleitung

Durch den heutigen Stand der Technik im Bereich der Digitalisierung und Automatisierung sowie durch die gute Verfügbarkeit an hoch performanten Computern nimmt die Anzahl an Softwarelösungen rasch zu. Dabei handelt es sich häufig um Software, die an bestimmte Rahmenbedingungen gebunden ist. Diese werden durch äußere Faktoren wie zum Beispiel Marktänderung, Kundenwünsche oder wissenschaftlichen Fortschritt beeinflusst. Durch die kontinuierlich wechselnden Anforderungen, treten ebenfalls Änderungen in der Spezifikation auf, die Anpassungen an der Software nach sich ziehen. Aus diesem Grund muss nachhaltige Software nach Konzepten entwickelt werden, die es erlauben flexible Modifikationen des Verhaltens einzuarbeiten, ohne ein Versagen der Software hervorzurufen.

Vor einigen Jahren entstand im Leichtweiß Institut für Wasserbau in Braunschweig die Idee die Empfehlungen für die Ausführung von Küstenschutzwerken (EAK, 2002) als Software

umzusetzen, um fachspezifische Formeln und Problemstellungen schneller und automatisiert auswerten zu können. Dabei ist zu beachten, dass die mathematischen und physikalischen Beziehungen miteinander gekoppelt sind. Aus diesem Grund ist die Applikation so zu gestalten, dass verschiedene Einzeltheorien kombinierbar sind, sodass Ergebnisse der einen Berechnung auch als Eingangswerte für eine andere dienen können. Desweiteren soll es die Applikation zulassen bereits umgesetzte Theorien zu verändern sowie neue Theorien aufzunehmen. So wird sichergestellt, dass die Applikation jederzeit an die neusten Ergebnisse der Forschung anpassbar ist. Im Rahmen dieses Artikels ist ein Applikationsframework entstanden, das es ermöglicht die EAK (2002) schrittweise zu integrieren und dabei die Ergebnisse verschiedenster mathematischer und physikalischer Zusammenhänge zu visualisieren. Das hier entstandene Applikationsframework wurde in Zusammenarbeit mit dem iRMB¹² entwickelt. Es zeigt die Entstehung nachhaltiger Software durch Umsetzung der Schlüsselkonzepte des OOD.

Für das Verständnis dieses Artikels werden Kenntnisse über die Methoden und Prinzipien des objektorientierten Designs vorausgesetzt.

2 Spezifikation des Applikationsframeworks CECloud

Mit CECloud soll ein Anwendungsframework entstehen, das zur automatisierten Berechnung und Visualisierung von mathematischen und physikalischen Beziehungen im Fachbereich Küsteningenieurwesen (KI) verwendet werden kann. Der Name CECloud setzt sich aus „CE“ und „Cloud“ zusammen. Dabei steht „CE“ für Coastal Engineering¹³. Das Wort „Cloud“ soll verdeutlichen, dass es sich dabei um eine Sammlung verschiedener Tools in diesem Bereich handelt.

Als Beispiel dazu werden hier die Lineare Wellentheorie nach Airy & Laplace (LWT, EAK, 2002) und die Berechnung der Wellenkraft auf Pfahlbauwerke nach Morison (EAK, 2002) erläutert:

Lineare Wellentheorie Die LWT wird im KI dazu verwendet deterministische Aussagen über die Ausbreitung von Wasserwellen und deren Dynamik zu treffen. In die Berechnung gehen dafür die Parameter des vorherrschenden Wellenklimas und Informationen über den betrachteten Zustand ein. Das Wellenklima wird charakterisiert durch die Parameter Wellenlänge L , Wellenperiode T , Wellenhöhe H und Wassertiefe h . Der betrachtete Zustand (i. F. Koordinatensystem genannt) drückt die räumliche und zeitliche Distanz x und t zum Ausgangszustand sowie die betrachtete Wassertiefe z aus.

Druckkraftberechnung nach Morison Trifft eine Welle auf ein Pfahlbauwerk resultieren daraus Druckkräfte auf den Pfahl, die über die Wassertiefe veränderlich sind. Die Berechnung der Druckverteilung hängt von der Dynamik der Welle ab, welche durch die LWT beschrieben wird. Zur Ermittlung der Druckverteilung müssen deswegen die Ergebnisse der LWT bekannt sein. Desweiteren hängt die Berechnung nach Morison von dem Pfahldurchmesser ab, sowie von der Viskosität und Dichte des Wassers.

Die Erweiterung des Applikationsframeworks durch neue Theorien wird anhand der Berechnung des Sedimenttransports erläutert.

¹²Institut für rechnergestützte Modellierung, TU Braunschweig

¹³Küsteningenieurwesen

Sedimenttransport Durch die Bewegungen der Welle entstehen Strömungen, die das Sediment der Gewässersohle verlagern. Berechnungen in diesem Kontext gehören zu der Berechnungstheorie des Sedimenttransports. Diese ist abhängig von den Ergebnissen der LWT und erweitert diese um die Parameter des Wassers sowie des betroffenen Sediments.

Bei der Auswertung der Berechnungsvorschriften ist zu beachten, dass Theorien voneinander abhängen können. Die dadurch entstehenden hierarchischen Strukturen sind dann nacheinander auszuwerten. Dabei greift die Theorie, die ganz oben in der Hierarchie steht (z.B. Morison), auf die Ergebnisse untergeordneter Theorien zu (z.B. LWT) und wird dann zuletzt ausgewertet.

Das Applikationsframework soll dem Benutzer die Möglichkeit geben Berechnungstheorien in einer solchen hierarchischen Struktur zu organisieren und auszuwerten. Außerdem soll die Software durch neue Theorien erweitert werden können, ohne die Basisstruktur des Frameworks zu verändern.

3 Die Struktur der Applikation CECloud

Eine der Methoden des OOD ist die Modularisierung (Meyer, 2011). Im Zuge dessen untergliedert der Entwickler die gesamte Problemdomäne in mehrere Bereiche, die einzeln gelöst und bearbeitet werden können. Die Zusammenführung aller Einzelmodule ergibt später die Gesamtlösung. Im Groben lassen sich zwei Hauptaufgaben identifizieren:

- (1) Berechnung und Datenrepräsentation
- (2) Visualisierung und Präsentation der Berechnungsergebnisse

3.1 Die Basisarchitektur

Damit das Single Responsibility Principle (SRP, R. C. Martin und M. Martin, 2008) auf oberster Architekturebene umgesetzt wird, erfordert die Implementierung der Software die Aufteilung in Module mit den oben genannten Aufgabenbereichen.

Model Die unter 1 aufgeführten Zuständigkeiten sind Gegenstand des Modells. Hier wird die gesamte Struktur organisiert, die zur Lösung der Problemdomäne beiträgt. Dazu gehören zum einen der Aufbau der in Kapitel 2 beschriebenen hierarchischen Projektstruktur, zum anderen die Auswertung und Berechnung der Projekte.

Der unter 2 aufgeführte Aufgabenbereich ist für die Visualisierung des Modells zuständig. Da sich die Art der Präsentation häufiger ändert als die Logik der Problemdomäne, handelt es sich hierbei um einen volatilen Bereich des Programms. Um die Logik des Modells vor den Auswirkungen der Modifikation in der Präsentation zu schützen, wird dieser Aufgabenbereich erneut unterteilt.

View Die tatsächliche Visualisierung in Form von Fenstern, Grafiken, Diagrammen etc. ist Gegenstand der View. Hier werden lediglich Daten auf dem Bildschirm präsentiert ohne dabei Informationen über den semantischen Hintergrund zu kennen. Die View ist vollständig unabhängig von dem Modell.

Presenter Die Schnittstelle zwischen View und Model ist der Presenter. Der aktuelle Modelzustand wird durch den Presenter erfasst und die Daten für die View aufbereitet. Anschließend werden die Daten zur Darstellung an die View delegiert.

Es entsteht eine Softwarearchitektur nach dem *Model-View-Presenter*-Muster (Potel, 1996), durch die eine lose Kopplung zwischen Model und View entsteht. Dadurch wird sichergestellt, dass die Art der Präsentation keine Auswirkung auf die Logik der Problem-domaine hat.

In diesem Artikel wird der Schwerpunkt auf die nachhaltige Entwicklung der Modellogik gelegt.

3.2 Die Hierarchie der Projekte

Durch erneute Anwendung der Modularisierung innerhalb des Models entstehen kleine Bausteine, die dann flexibel miteinander kombiniert werden können. Abbildung 1 zeigt die Klassen und ihre Beziehungen zur Implementierung der in Kapitel 2 erläuterten Projekte LWT und Morison. Jede dieser Klassendefinitionen kapselt die Repräsentation eines bestimmten Bausteins, der dann durch Komposition in Beziehung mit anderen gebracht wird. In der Klasse *WaveDynamic* findet die Berechnung der LWT statt, die als Eingangsparameter die in Kapitel 2 erläuterten Werte des Koordinatensystems und der Welle benötigt. Diese Parameter sind in den Klassen *CoordinateSystem* und *Wave* enthalten.

Die Berechnungen der Morisonkräfte finden sich in *Morison* wieder. Dafür werden die Ergebnisse der Klasse *WaveDynamic* sowie die Parameter des Pfahls und des Wassers benötigt. Diese sind analog zum Koordinatensystem und zur Welle in den Klassen *Pile* und *Water* gekapselt.

3.3 Erweiterung der Projekthierarchie durch das Projekt Sedimenttransport

Durch den gewählten Aufbau lässt sich das Model gemäß des *Open Closed Principle* (OCP, Meyer, 2011) mit neuen Projekttypen erweitern. Dafür kann auf bereits implementierte Klassen zurückgegriffen werden, die neu kombiniert werden können. Zusätzlich können Klassendefinitionen hinzugefügt werden, die in die Projekthierarchie eingehen.

Abbildung 2 zeigt die Erweiterung von CECloud durch den Projekttypen Sedimenttransport. Dazu müssen die neuen Klassendefinitionen *SedimentTransport* und *Sediment* angelegt werden. Die Ergebnisse der LWT werden durch die Verwendung der Klasse *WaveDynamic* zur Verfügung gestellt. Die Parameter, die das Sediment charakterisieren, sind dabei in der Klassendefinition *Sediment* zusammengefasst.

Durch Anwendung der Modularisierung im Model wurde das SRP auf Klassenebene umgesetzt. Dadurch entstehen zum einen Klassen mit hoher Kohäsion (Yourdon, 1979) zum anderen kann die Wiederverwertbarkeit erhöht und Redundanzen vermieden werden, indem neue Projekttypen bereits implementierte Klassendefinitionen verwenden.

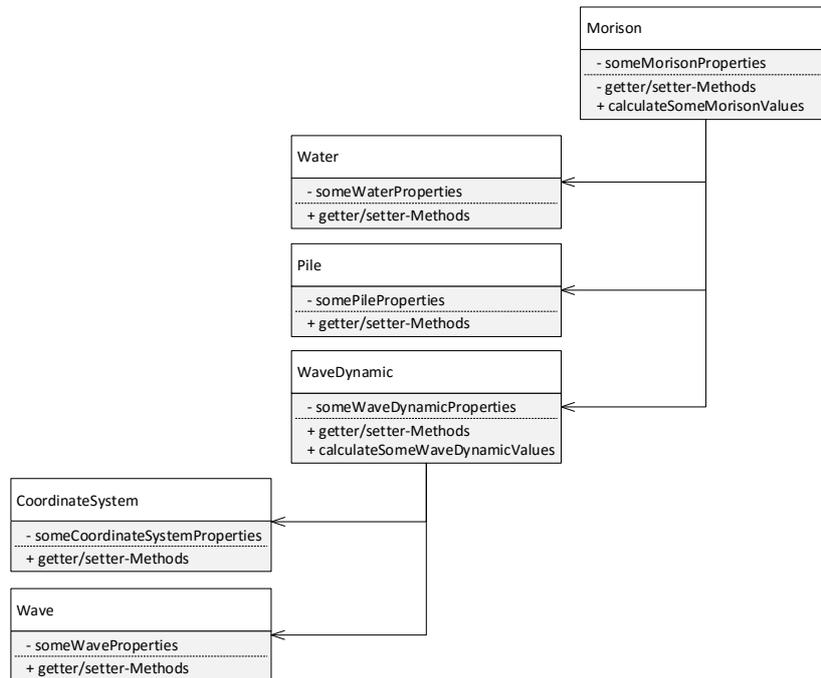


Abb. 1: Hierarchie der Projekte

3.4 Visualisierung der Projekthierarchie

Die Darstellung der Projekthierarchie geschieht durch eine Baumstruktur (*TreeView*) innerhalb einer Explorer-Ansicht, wobei jedes Projekt durch einen Knoten repräsentiert wird. Zu den im Baum angezeigten Informationen gehören der Name, der Typ sowie das Icon des darzustellenden Projekts. In einer weiteren Ansicht werden die Eigenschaften der Projekte in Form einer Tabelle (*TableView*) visualisiert. Innerhalb der Tabelle werden zu jeder Zeit die Eigenschaften des im Baum ausgewählten Projekts angezeigt (siehe Abbildung 3).

Das Zeichnen der Baumstruktur in dem *Solution Explorer* beginnt bei einem nicht sichtbaren Wurzel-Element. Ausgehend von dieser Wurzel werden die einzelnen sichtbaren Kind-Knoten gezeichnet. Dafür werden Name, Typ und Icon der Projekte erfragt und an die richtige Stelle gesetzt. Nach dem Zeichnen eines Knotens wird der gleiche Prozess mit den Kind-Knoten durchgeführt. Um korrekt dargestellt zu werden muss ein Projekt öffentliches Verhalten zur Verfügung stellen, dass zum einen die Eigenschaften, zum anderen die Positionen der Kind-Knoten publiziert.

Um die Erweiterung des Frameworks gemäß dem OCP zu ermöglichen, muss die Funktionalität für neue Projekttypen gewährleistet werden. Hier findet sich eine weitere Methode des OOD wieder. Abbildung 4 zeigt die durch Anwendung der Abstraktion entstandene Klassenhierarchie, die durch die abstrakte Basisklasse *HierarchyNode* repräsentiert wird. Um die Baumstruktur zu zeichnen, ist die Abhängigkeit der Klasse *TreeView* zur Klasse *HierarchyNode* ausreichend, hinter der die konkrete Implementierung der einzelnen Projekte verborgen ist. Zur Einführung neuer Projekttypen kann nun von der Basisklasse *HierarchyNode* abgeleitet werden. Folgen die neuen Projekte dem *Liskov's Substitution Principle* (LSP, Liskov, 1987), wird die

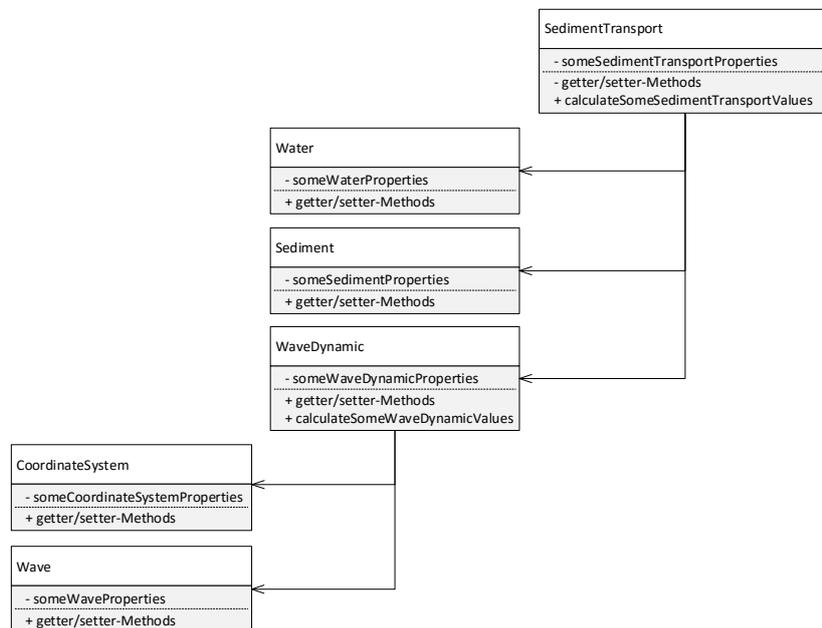


Abb. 2: Erweiterung der Projekthierarchie durch den Projekttyp Sedimenttransport

Funktionalität der Baumstruktur nicht gefährdet. Gleichzeitig wird durch Einführung der Klassenhierarchie das *Dependency Inversion Principle* (DIP, R. C. Martin und M. Martin, 2008) umgesetzt. Um die konkreten Klassen im *TreeView* zu zeichnen, müsste dieser jede Klassendefinition kennen, um auf deren Methoden zugreifen zu können. Die Verwendung der Klasse *HierarchyNode* kehrt diese Abhängigkeiten um. Der *TreeView* definiert dabei die Schnittstelle mit der er kommuniziert. Jede Klasse die in der Baumstruktur gezeichnet werden soll, muss das Verhalten dieser Schnittstelle implementieren und hängt somit von ihr ab. Dadurch entsteht eine lose Kopplung zwischen der Klasse *TreeView* und den konkreten Klassendefinitionen der Projekte.

Durch das direkte Darstellen der Klasse *HierarchyNode* in der Baumstruktur entsteht Präsentationslogik, die sich nun im Model befindet (*HierarchyNode*). Dazu gehören, die Positionen der Kind-Knoten und das Icon (siehe Abbildung 4). Die Berechnung und Auswertung der verschiedenen Projekte ist unabhängig von der Ordnung der Kinder bzw. von dem gewählten Icon. Durch Kapselung beider Verantwortungen in der Klasse *HierarchyNode* wird nun die Modellogik mit der Präsentationslogik gekoppelt. Änderungen in der einen Verantwortung können ebenfalls Modifikationen der Anderen hervorrufen. Da es sich bei der Präsentationslogik um einen sehr volatilen Bereich handelt, ist die Modellogik von der Präsentation abhängig. Um die Unabhängigkeit des Models zu wahren, wird hier auf Klassenebene das SRP angewendet. Es wird eine neue Klasse *HierarchyItem* erstellt, welche für die graphische Aufbereitung der für den *TreeView* relevanten Daten zuständig ist. In dieser Klasse wird die Reihenfolge der Kind-Knoten sowie das Icon festgelegt. Damit ist gewährleistet, dass jederzeit das Icon und Ordnung der Kinder ausgetauscht werden können, ohne die Modellogik zu gefährden. Auch unterhalb von *HierarchyItem* ist eine Klassenhierarchie notwendig. Diese muss parallel zu jener unterhalb von *HierarchyNode* verlaufen. Für jeden Projekttypen ist eine spezielle Item-Klasse

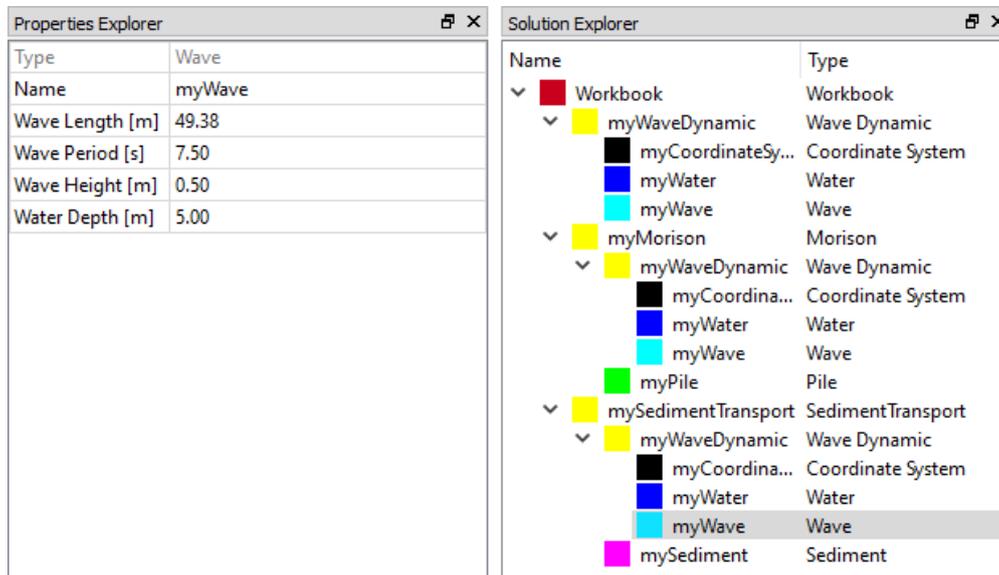


Abb. 3: Explorer-Ansichten

erforderlich, die auf die verschiedenen Ausprägungen¹⁴ eingeht.

Die Visualisierung der Eigenschaften des im *TreeView* ausgewählten Projekts geschieht über die Klasse *TableView*, die ein *PropertiesModel* zeichnet (siehe Abbildung 4). Auch hier wird unter Berücksichtigung des SRP eine weitere Klassenhierarchie erstellt. Diese enthält die Logik für die Zusammenstellung der projektspezifischen Eigenschaften in einem *PropertiesModel*. Die Eigenschaften können dabei durch die in Abbildung 3 gezeigten *getter-Methoden* erfragt werden.

Die Auswahl der entsprechenden Klassendefinitionen zwischen den verschiedenen Hierarchien erfolgt über einen *Look-Up-Mechanismus* (Fowler, 1997). Anhand des Typs eines *HierarchyNodes*, der über die *getType-Methode* erfragt wird, kann lokalisiert werden welche Klasse einer anderen Hierarchie benötigt wird. Anschließend erfolgt eine sichere Typumwandlung des *HierarchyNodes* zu der entsprechenden konkreten Klasse. Dadurch kann das spezifische Verhalten der Projekte verwendet werden.

3.5 Bearbeiten der Projekteigenschaften

Die Dateneingabe erfolgt über das Eigenschaftsfenster (Abbildung 3, links). Hier können die einzelnen Eigenschaften der Projekte bearbeitet werden. Jedes Projekt birgt unterschiedliche Eigenschaften, die durch den Benutzer verändert werden können. Um auch in diesem Kontext die Erweiterung gemäß des OCP ermöglichen zu können, wurde ein generalisierter Weg umgesetzt.

Wie in Abbildung 5 dargestellt, existiert eine weitere Klassenhierarchie, die durch die abstrakte Klasse *Editor* repräsentiert wird. Auch hier wird durch den zuvor erwähnten *Look-Up-Mechanismus* der entsprechende *Editor* ausgewählt. Diesem wird dann der Name sowie

¹⁴Icon sowie Anzahl und Reihenfolge der Unterprojekte

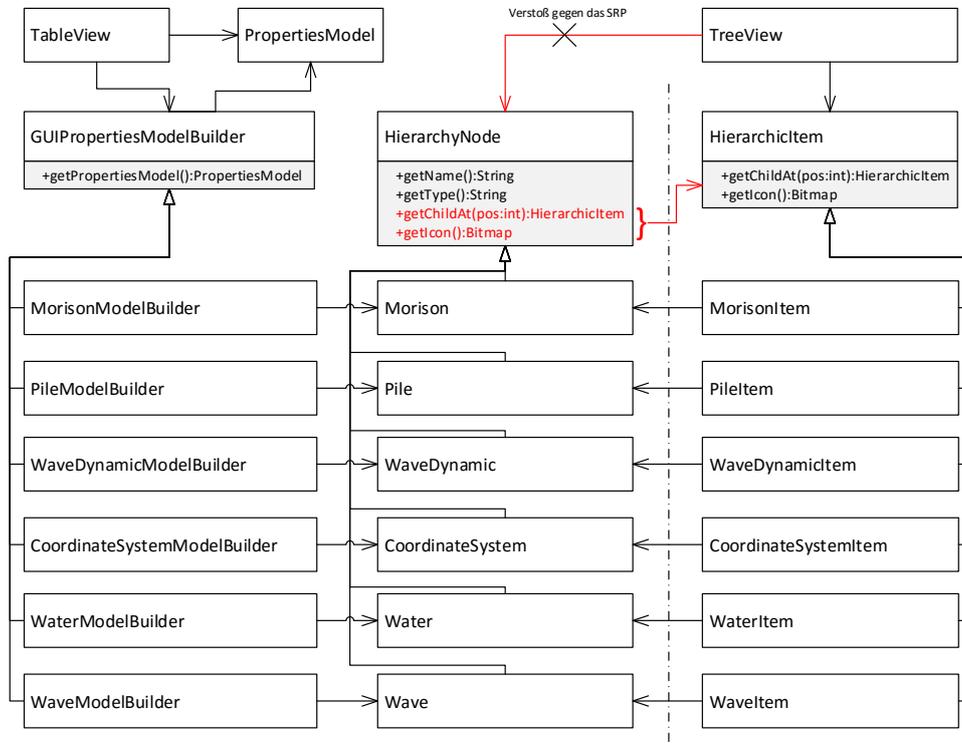


Abb. 4: Hierarchien für die Anzeige der Knoten in TreeView und der Eigenschaften im TableView

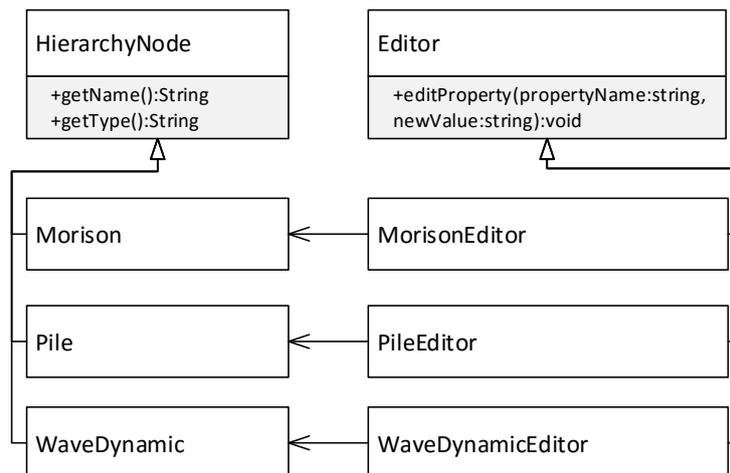


Abb. 5: Editor-Hierarchie

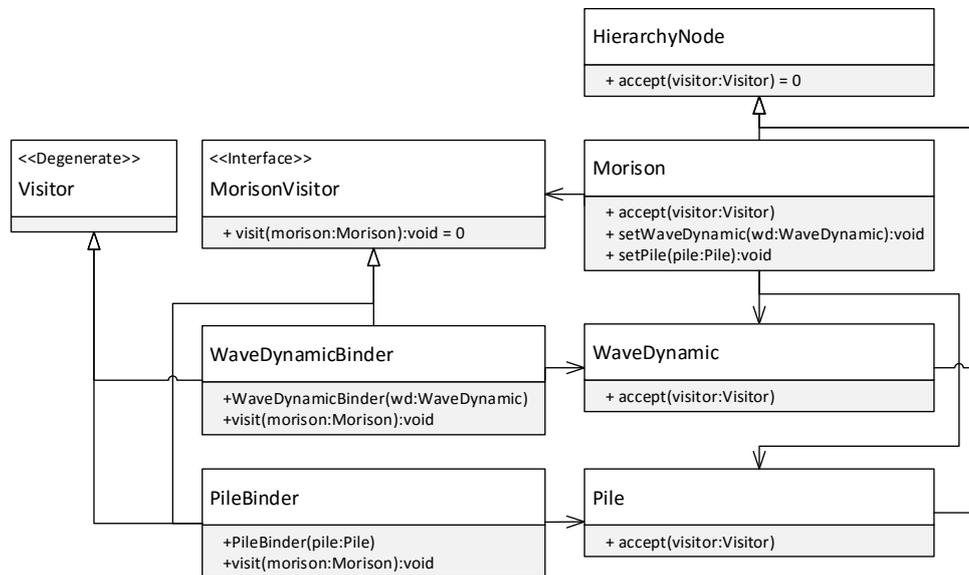


Abb. 6: Binding-Mechanismus durch Besucher Muster

der neue Wert der Eigenschaft übermittelt. Die Editor-Klasse ist dafür zuständig anhand des Namens der Eigenschaft die entsprechende *setter-Methode* zu verwenden und die Eigenschaft auf den neuen Wert zu setzen.

3.6 Verbinden der Projekte

Die Auswertung der Projekte kann nur erfolgreich durchgeführt werden, wenn die in Kapitel 2 beschriebenen Eingangsparameter vorliegen. Dafür müssen die einzelnen Projekte miteinander verknüpft werden. Um beispielsweise ein *WaveDynamic* zu einem *Morison* hinzuzufügen, wird die in Abbildung 6 gezeigte *setWaveDynamic*-Methode aufgerufen. Dafür ist ein Objekt vom Typ *WaveDynamic* notwendig. Da im Kontext des abstrakten Applikationskern nur der Typ *HierarchyNode* bekannt ist, wird zur Rückgewinnung der Typinformation das *acyclic Visitor-Pattern* (azyklische Besucher-Entwurfsmuster, R. C. Martin und M. Martin, 2008) verwendet. Die an diesem Entwurfsmuster teilnehmenden Klassen sind das degenerierte Interface *Visitor*¹⁵, die projektspezifischen Besucherschnittstellen (z.B. *MorisonVisitor*), eine konkrete Implementierung dieser Schnittstellen (*WaveDynamicBinder*) sowie die Projektklassen selbst (siehe Abbildung 6).

Die *Binder*-Klassen enthalten die Logik für das Verknüpfen der Projekte. Sie rufen nach der Identifikation der Typen die entsprechende *setter-Methode* auf. Der Mechanismus des Bindens wird anhand eines Beispiels erklärt, in welchem die Verknüpfung eines Objekt des Typs *WaveDynamic* mit einem *Morison* stattfindet. Zunächst wird dazu mit dem zu bindenden Objekt ein Binder des entsprechenden Typs erstellt (hier *WaveDynamicBinder*). Den benötigten Typ des Binders stellt die *getType*-Methode der Klasse *HierarchyNode* zur Verfügung. Anschließend wird der Binder dem Objekt des Typs *Morison* in der *accept*-Methode übergeben. Innerhalb

¹⁵DIP, Basiskern stellt Schnittstelle zur Verfügung von welcher die konkreten Implementierungen (z.B. *WaveDynamicBinder*) abhängen

dieser Methode findet eine dynamische Typumwandlung von *Visitor* zu *MorisonVisitor* statt. Danach übergibt sich das *Morison* selbst dem *MorisonVisitor*. Nun sind beide Typen der beteiligten Projekte bekannt. Es kann die *setWaveDynamic*-Methode des *Morison* aufgerufen werden.

Diese Umsetzung des Verknüpfens erlaubt die Erweiterung der Projekthierarchie durch neue Projekttypen (OCP) sowie ihre flexible Kombination ohne den Kern der Applikation zu verändern. Dafür muss zu jedem neuen Projekttyp eine spezielle Besucherschnittstelle und ein Binder erstellt werden. Die Schnittstelle ermöglicht es Bindern anderen Typs die Objekte des neuen Typs zu besuchen. Dadurch können Projekte anderen Typs an den neuen Projekttyp gebunden werden. Der Binder des neuen Typs enthält die Logik für das Binden des neuen Projekttyps an andere Objekte.

3.7 Visualisierung der Projekte

Nachdem die einzelnen Projekte vollständig mit Daten gefüllt sind, können diese ausgewertet werden. Dafür werden sowohl Klassen im Presenter, als auch im Bereich der View benötigt. Diese Klassen basieren auf der konkreten Implementierung der einzelnen Projekte. Sie können Ergebnisse und ausgewertete Daten von diesen erfragen und darstellen. Für jedes Projekt um welches das Applikationsframework erweitert wird, können neue Presenter-Klassen sowie eine neue View-Klassen erstellt werden ohne den Kern der Applikation zu modifizieren (OCP).

3.8 Erweiterung der Applikation um das Projekt SedimentTransport

Um die in 3.3 beschriebene Erweiterung der Projekthierarchie zu erreichen, muss die Struktur des Applikationsframeworks um einige Klassen vergrößert werden. Dazu leitet man von den zuvor beschriebenen abstrakten Schnittstellen im Applikationskern ab. Eine Erweiterung des Applikationsframeworks um ein Modul, welches die mathematischen Beziehungen zur Berechnung des Sedimenttransports beinhaltet, müssen die Klassen *SedimentTransport*, *SedimentTransportItem*, *SedimentTransportModelBuilder*, *SedimentTransportEditor* und *SedimentTransportBinder* hinzugefügt werden.

4 Zusammenfassung

Dieser Artikel erläutert die nachhaltige Entwicklung von Software durch die Umsetzung der Methoden und Prinzipien des objektorientierten Designs anhand des Applikationsframework *CECloud*. Hierbei handelt es sich um Software die automatisiert Berechnungen im Bereich des Küsteningenieurwesens auswertet und visualisiert.

In einem ersten Schritt werden durch Anwendung der Modularisierung und des Single Responsibility Principle die Aufgabenbereiche der Datenhaltung und der Datenpräsentation in einzelne Module unterteilt. Dadurch werden zwei übergeordnete Module erzeugt, in denen eine große Zusammengehörigkeit der Daten vorherrscht (Kohäsion). In einem weiteren Schritt der Modularisierung wird die Datenpräsentation erneut in zwei Komponenten geteilt. Es entsteht eine *Model-View-Presenter*-Architektur. Dabei ist das Model für die Datenhaltung zuständig. Die View präsentiert den aktuellen Zustand des Models, den sie vom Presenter-Modul erhält. Durch Verwendung dieses Architekturmusters wird das Model von dem volatilen Bereich der

View entkoppelt. Dies erlaubt eine Änderung der Darstellungsweise ohne das Risiko die Logik der Problemdomäne zu gefährden.

Durch feinere Untergliederung im Model entstehen logische Bausteine (Projekte), in denen zusammengehörige mathematische und physikalische Beziehungen gekapselt sind. Durch die Umsetzung des *Single Responsibility Principles* innerhalb dieser Klassen erhöht sich die Wiederverwertbarkeit. Werden bereits implementierte Berechnungsvorschriften neu verwendet, wird auf existierende Projekte zugegriffen. Die dadurch vermiedenen Redundanzen steigern die Wartbarkeit der Software. Für einige Projekte bedarf es die Ergebnisse anderer Berechnungen, wodurch eine Projekthierarchie entsteht. Diese Hierarchie wird stufenweise ausgewertet, wobei die Projekte höheren Rangs auf die Ergebnisse jener niedrigeren Rangs zugreifen. Die Visualisierung der Projekthierarchie geschieht über eine Baumstruktur, in der ein projektspezifisches Icon, ein Name und ein Projekttyp dargestellt wird. Damit die Baumstruktur jeden Projekttyp darstellen kann, muss sie jede dieser Klassendefinitionen kennen. Dadurch entsteht eine enge Kopplung zwischen den Projekten und der Baumstruktur. Durch Anwendung der Abstraktion und Umsetzung des *Dependency Inversion Principle* lässt sich diese Kopplung minimieren. Es entsteht eine Klassenhierarchie, die hinter der abstrakten Klasse *HierarchyNode* verborgen ist. Mithilfe der Polymorphie kann nun die Applikation gemäß des *Open Closed Principle* durch neue Projekttypen erweitert werden, indem diese von *HierarchyNode* abgeleitet werden. Zur Wahrung des *Single Responsibility Principle* wird eine parallele Klassenhierarchie erstellt, die durch die Schnittstelle *HierarchyItem* repräsentiert wird. In dieser Klassenhierarchie findet die graphische Aufbereitung der für den Baum relevanten Daten der Projekte statt. Nun kann die Darstellung des Projektes im Baum verändert werden, ohne Modifikation im Model durchzuführen. Neben der Projekthierarchie zeigt eine Tabellenansicht die Eigenschaften des im Baum ausgewählten Projektes. Die Tabelle wird auf Grundlage der Klasse *PropertiesModel* gezeichnet. Da jedes Projekt unterschiedliche Eigenschaften hat, wird auch in diesem Kontext eine Klassenhierarchie erzeugt, die parallel zu der Hierarchie unterhalb der Klasse *HierarchyNode* ist. Somit existiert zu jedem Projekt eine weitere Klassendefinition, die ein entsprechendes *PropertiesModel* erstellen kann. Für jede Aufgabe, die hinzugefügt werden soll, wird eine neue parallele Klassenhierarchie erzeugt. Dabei wird der Basiskern nur um eine abstrakte Klasse erweitert. Durch Polymorphie wird dann die Funktionalität für die verschiedenen Projekttypen erreicht. Die Erweiterung des Applikationskerns durch abstrakte Schnittstellen zur Umsetzung neuer Funktionalität wurde am Beispiel der Editor- und der Visitor-Klassenhierarchie gezeigt. Letztlich wird Anhand der Erweiterung der Struktur von CECloud durch das Projekt *Sediment-Transport* gezeigt, dass die gewählte Architektur leicht erweiterbar ist, in dem jede Klassenhierarchie um eine entsprechendes Derivat vergrößert wird. Dabei wird keine Logik modifiziert, sondern nur neue Logik hinzugefügt. Dadurch wird es vermieden bereits implementierte Software zu gefährden. Durch die Umsetzung der Methoden und Prinzipien des Objektorientierten Designs entsteht leicht erweiterbare Software mit hoher Wartungsfreundlichkeit.

Literatur

Ausschuss für Küstenbauwerke (EAK) (2002). *Empfehlungen für die Ausführungen von Küstenschutzwerken*.

Fowler, M. (1997). *Analysis Patterns: Reusable Object Models*.

Gamma, E., Helm, R., Johnson, R. E. und Vlissides, J. (1994). *Design Patterns. Elements of Reusable Object-Oriented Software*.

Liskov, B. (1987). „Data Abstraction and Hierarchy“. In: *ACM Sigplan Notices* 23.5, S. 17–34.

Martin, R. C. und Martin, M. (2008). *Agile Software Development. Principles, Patterns and Practices*.

Meyer, B. (2011). *Object-oriented software construction*.

Potel, M. (1996). *MVP: Model-View-Presenter. The Taligent Programming Model for C++ and Java*.

Yourdon E. and Constantine, L. (1979). *Structured Design*.

Effiziente Implementierung eines Gittergenerators auf GPGPU's

Sören Peters¹ und Jan Linxweiler²

*Institut für rechnergestützte Modellierung im Bauingenieurwesen, TU Braunschweig,
Pockelsstr. 3, 38106 Braunschweig, Deutschland*

¹E-Mail: soeren.peters@tu-braunschweig.de, ²E-Mail: j.linxweiler@tu-braunschweig.de

Abstract In dieser Arbeit wird die Implementierung eines Gittergenerators zur Verwendung auf der Grafikkarte vorgestellt. Hierbei generierte Gitter dienen als Grundlage zur numerischen Berechnung von Strömungssimulationen, die ebenfalls auf der Grafikkarte durchgeführt werden. Die entstandene Applikation demonstriert eine effiziente und robuste Implementierung der Gittergenerierung, die annähernd beliebige dreidimensionale Geometrien abbilden kann. Infolgedessen können interaktive Geometrieänderungen zur Laufzeit vorgenommen werden, ohne dabei eine signifikante Unterbrechung der Strömungssimulation zu erzeugen.

Keywords: Gittergenerierung, LBM, HPC, GPU, Strömungsmechanik

1 Einleitung

Die computergestützte numerische Strömungssimulation ist seit vielen Jahren fester Bestandteil von Wissenschaft und Forschung. Ziel ist es mithilfe von Modellgleichungen reale Systeme von Strömungen abzubilden und Aussagen über die Entwicklung des Fluids zu treffen. Dabei kann das Verhalten eines Fluids in Raum und Zeit mittels der bekannten Navier-Stokes-Gleichungen beschrieben werden. Da sich die Gleichungen jedoch nur in Sonderfällen analytisch lösen lassen, müssen die Strömungsprobleme i.d.R. unter der Annahme weiterer Vereinfachungen näherungsweise gelöst werden. Eine Herangehensweise zur näherungsweisen Lösung der Navier-Stokes-Gleichungen ist die Lattice-Boltzmann-Methode (LBM). Sie beschreibt die Bewegung und Interaktion der Fluide auf Teilchenebene. Weil das Verfahren sehr rechen- und speicherintensiv ist, werden für praxisrelevante Probleme typischerweise Hochleistungsrechner herangezogen. Allerdings hat sich gezeigt, dass die LBM gut parallelisierbar ist und das Verfahren insbesondere auf Grafikkarten (GPU¹⁶) effizient implementiert werden kann (Krafczyk, 2001; Linxweiler, 2011).

Das Simulationsverfahren lässt sich generell in drei Schritte unterteilen. Im ersten Schritt wird ein Gitter als Berechnungsgrundlage generiert. Danach folgt die Anwendung des Strömungslösers. Der letzte Schritt beinhaltet die Auswertung und Visualisierung der Ergebnisse. Im Vordergrund steht im Allgemeinen die Simulation. Eine Parallelisierung der Gittergenerierung ist häufig zweitrangig, da sie zumeist nur einmalig vor der eigentlichen Simulation

¹⁶engl.: Graphics Processing Unit

durchgeführt wird. Anders gestaltet sich die Situation, wenn während der Simulation Änderungen an der Geometrie erfolgen und somit der Simulationsvorgang unterbrochen und das Gitter neu berechnet werden muss. In diesem Fall hat der Prozess der Gittergenerierung einen erheblichen Anteil an der Gesamtlaufzeit, die sich durch eine zusätzliche Parallelisierung reduzieren lässt. Im Falle der Verwendung von Grafikkarten ist es zudem anzuraten, zu der Simulation auch die Gittergenerierung parallel auf der GPU durchzuführen, um zeitintensive Datentransfers zwischen dem Hauptspeicher des Prozessors (CPU)¹⁷ und der GPU zu vermeiden.

Dieser Artikel beginnt in Kapitel 2 mit einer Einführung in das Hochleistungsrechnen auf Grafikkarten. Dabei wird im Speziellen auf die Unterschiede zwischen der CPU und GPU eingegangen. In Kapitel 3 wird der Algorithmus des implementierten Gittergenerators vorgestellt. Dabei werden die geometrischen Berechnungen sowie der Parallelisierungsansatz erläutert. Zuletzt werden in Kapitel 4 die Datenstrukturen und die Implementierung des Kernels beschrieben.

2 GPGPU – general purpose graphics processing unit

Wie bereits erwähnt ist die Lattice-Boltzmann-Methode gut parallelisierbar und eignet sich daher insbesondere zur Berechnung auf Grafikkarten. Um umfangreiche Kopiervorgänge auf die Grafikkarte zu vermeiden, wird der Gittergenerierungsprozess auf diese verlagert. Die Gemeinsamkeiten und Unterschiede zwischen der CPU und GPU sowie die Funktionsweise der GPU werden im folgendem Kapitel beschrieben.

2.1 Von der CPU zur GPU

Traditionell werden in der elektronischen Datenverarbeitung Berechnungen auf der CPU durchgeführt. Gordon Moore formulierte 1956 ein Richtmaß, nachdem sich die Anzahl der Transistoren bei integrierten Schaltungen in einem Prozessor alle 12 bis 24 Monate bei gleicher Flächennutzung verdoppelt. Eine höhere Anzahl von Transistoren führt zu schnelleren Berechnungen und Speicherzugriffsraten. In den ersten 35 Jahren nach der Formulierung durch Moore erfüllte sich seine Prognose.

Die CPU ist darauf optimiert auf wenigen Rechenkernen Daten seriell und effizient zu verarbeiten. Eine hohe Taktung des Prozessors verspricht dabei viele Fließkommaoperationen je Sekunde (FLOPS¹⁸). Andererseits kann auch die Taktung eines Rechenkerns durch Wärmeentwicklung und Stromverbrauch nicht beliebig gesteigert werden. Die Gesetzmäßigkeit nach Moore stößt daher heute aus physikalischen Gründen an ihre Grenzen. Um dessen ungeachtet mehr Leistung zu erzielen werden heute *Mehrkernprozessoren* eingesetzt. Im Gegensatz zur CPU sind in der GPU heute hunderte bis tausende von Rechenkernen miteinander verbunden. Als Beispiel sei der Grafikchip GP100 von NVIDIA genannt, der mit 15 Milliarden Transistoren auf fast 4000 Rechenkernen ausgestattet ist. Moores Vorhersage wird damit von der GPU nach wie vor übertroffen. Die Verantwortung der Grafikkarte lag in der Vergangenheit hauptsächlich in der Berechnung und Darstellung von Grafiken. Durch die große Anzahl von Recheneinheiten können gleiche Programmabläufe massiv parallel berechnet werden, zum Beispiel die Rasterung von Grafiken.

¹⁷engl.: Central Processing Unit

¹⁸engl.: floating point operations per second

2.2 Beschleunigung mit GPU's von NVIDIA

Im Allgemeinen erstellt ein Entwickler eine Computersoftware in einer ihm verständlichen Hochsprache, zum Beispiel C oder C++. Ein entsprechender Compiler übersetzt den Quellcode in eine Form, die von dem jeweiligen Prozessor ausgeführt werden kann. Rechenintensive Abschnitte innerhalb des Programms können auf die Grafikkarte ausgelagert werden. Das Unternehmen NVIDIA Corporation entwickelt neben HPC¹⁹ geeigneten Grafikkarten, ein Framework, zur Portierung und Ausführung von C-Code auf NVIDIA-GPU's – das CUDA²⁰-Framework. Zusätzlich stellt NVIDIA den NVCC-Compiler zur Verfügung. Code, der auf der GPU ausgeführt werden soll, muss sich in einer Datei mit der CUDA Dateinamenserweiterung (.cu) befinden. Der Quellcode, der auf der CPU läuft, wird von dem NVCC-Compiler an den entsprechenden CPU-Compiler gesendet (z.B.: gcc oder mvc). Die anderen Programmteile werden direkt vom NVCC-Compiler kompiliert und bei Programmausführung auf der GPU ausgeführt (vgl. Abb. 1) (NVIDIA, 2016a).

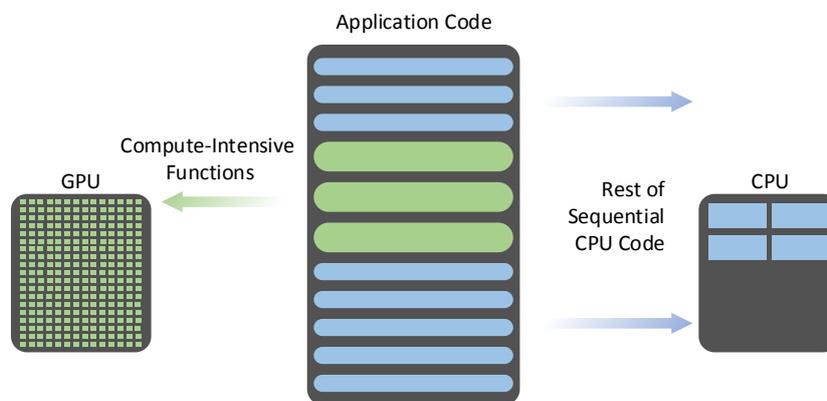


Abb. 1: Beschleunigung mit der GPU basierend auf NVIDIA, 2016c

2.3 Funktionsweise der Grafikkarte

Eine C-Funktion, die auf der GPU ausgeführt werden soll, wird als **Kernel** bezeichnet. Zu jedem Aufruf eines Kernels muss ein *Grid* definiert werden, das aus *Blöcken* besteht, welche wiederum aus *Threads* aufgebaut sind (vgl. Abb. 2). Ein Thread bezeichnet in der Informationsverarbeitung einen Ausführungsstrang. Ein Thread auf der GPU ist in der *Grid-Block-Thread*-Hierarchie die ausführende Einheit. Ein serieller Programmablauf entspricht dem Aufruf von einem Thread. In einem parallelen Programm wird die Anzahl der Threads entsprechend erhöht. Die GPU ist aus einer skalierbaren Anzahl von Streaming Multiprozessoren (SM) aufgebaut. Ein Block innerhalb des Grids wird von einem SM ausgeführt. Je nach Anzahl der vorhandenen SM auf der GPU werden die Blöcke auf diese verteilt. Wenn alle Threads innerhalb eines Blocks abgeschlossen sind, wird die Bearbeitung des nächsten Blocks gestartet. Es gehört zur Aufgabe des Entwicklers die Anzahl an Threads und Blöcken zu bestimmen, um eine effiziente Berechnung zu gewährleisten (NVIDIA, 2016a,b).

¹⁹engl.: high performance computing

²⁰engl.: Compute Unified Device Architecture

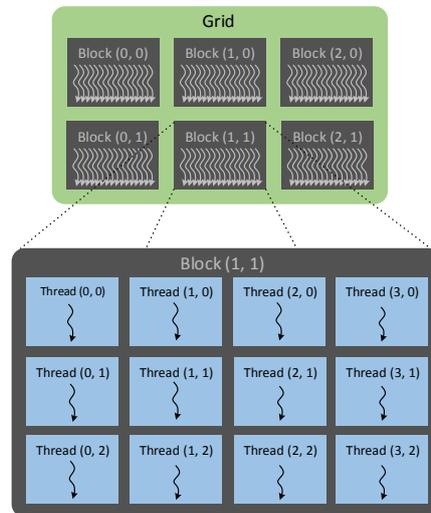


Abb. 2: 2D Grid mit 6 Blöcken und 12 Threads je Block basierend auf NVIDIA, 2016b

2.4 Software Architektur auf der Grafikkarte

Größere und komplexere Programme stellen Entwickler vor immer größere Hürden, die im Allgemeinen mithilfe der Objektorientierung erfolgreich gelöst werden sollen. Gleichzeitig steigen auch die Ansprüche an die Qualität der Software. Neben den korrekten Ergebnissen erwartet der Nutzer eine effiziente Verwendung der vorhandenen Ressourcen. Das Programm sollte sich intuitiv bedienen lassen und gegenüber Fehlverhalten des Nutzers robust reagieren. In den letzten 10 bis 30 Jahren erschien eine Vielzahl an Literatur über die Softwareentwicklung. Designprinzipien, Metriken und Entwurfsmuster sowie die korrekte Nutzung der objektorientierten Sprache geben Ansätze um diesen steigenden Ansprüchen an qualitativ hochwertiger Software gerecht zu werden (Gamma, 1995; R. C. Martin, 2009; R. C. Martin und M. Martin, 2008).

Eine weitere moderne Herangehensweise ist die agile Softwareentwicklung. Sie ist ein Sammelbegriff für eine Menge von meist iterativ agierenden Methoden der Programmierung. Eine agile Praktik ist die testgetriebene Entwicklung (TDD²¹). Dieses Schema folgt einem iterativen Zyklus von drei Schritten. Zuerst wird ein Test geschrieben, der fehlschlägt. Daraufhin wird der benötigte Produktionscode entwickelt, der diesen Test bestehen lässt. Im dritten und letzten Schritt werden sowohl die Tests als auch der Produktionscode restrukturiert²². Durch die Nutzung von TDD schafft der Entwickler sich eine saubere, automatisierte Kontrollbasis für die Software, die ihm jederzeit und kurzfristig Feedback über die Funktionalität des Programms liefern kann (Beck, 2003).

Die meisten der bekannten Methoden unterstützen den Programmierer in der Entwicklung von seriellem Code. Die vorhandene Literatur wird unpräziser, wenn parallele Prozesse hinzugezogen werden. Die ungleich höhere Komplexität durch die Parallelisierung der Programmabläufe geht zu Lasten einer adäquaten Softwarestruktur.

²¹engl.: test-driven-development

²²Refactoring - Strukturverbesserung der Software

Ein Programm, das nur auf der CPU läuft, kommt ohne die Grafikkarte aus. Umgekehrt jedoch enthält jede GPU-parallelisierte Software Bestandteile, die auf der CPU ausgeführt werden. Diese Teile sind allerdings grundlegend verschieden. Der CPU-Code ist in der Regel C++-Code und dementsprechend objektorientiert programmiert. Der GPU-Code ist im Fall von CUDA C-Code prozedural aufgebaut.

Nach Yourdon und Constantine (1979) soll in einem guten Design der Grad der Zusammengehörigkeit maximiert und gleichzeitig der Grad der Kopplung minimiert werden. Dementsprechend sollte der CPU- und GPU-Code räumlich von einander getrennt vorliegen. Die Kapselung kann in C++ mithilfe von statischen oder dynamischen Bibliotheken realisiert werden. Die Abhängigkeiten sollten dabei, wie in Abb. 3 illustriert, immer von der CPU in Richtung des Kernels zeigen.

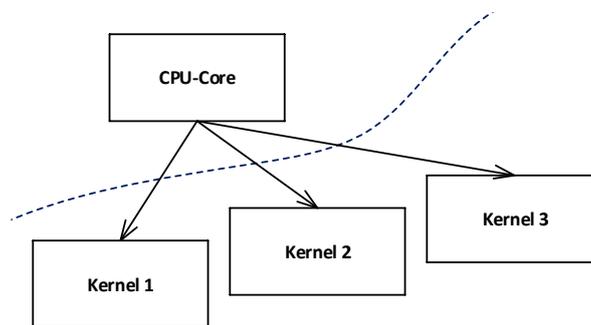


Abb. 3: Schematische Abhängigkeit des CPU-Cores von mehreren Kernel-Klassen

Dadurch entsteht der Vorteil, dass der GPU-Code von dem CPU-Code entkoppelt ist. Der Programmkernel wird damit zu einer Erweiterung für die GPU-Kernel. In der Regel ist der NVCC-Compiler langsamer als z.B. der gcc-Compiler. Durch die Entkopplung muss bei Änderungen des CPU-Codes kein Kernel neu kompiliert werden. Außerdem steigt die Wiederverwendbarkeit der Kernel.

Zusätzlich ist es von Vorteil ein Programm in möglichst viele kleine Funktionen aufzuteilen. Diese Funktionen sollten losgelöst von jeder parallelen Logik sein und möglichst genau eine Aufgabe erfüllen. Damit erfüllen die GPU-Funktionen das von Robert C. Martin formulierte "Single Responsibility Principle" (R. C. Martin und M. Martin, 2008). Es besagt, dass eine Funktion sich nur aus genau einem Grund ändern darf. Sie hat folglich nur eine Verantwortung, bzw. nur eine Aufgabe.

Kleine Funktionen haben mehrere positive Effekte zur Folge. Sie sind übersichtlicher und haben einen prägnanten Namen, der die Aufgabe der Funktion beschreibt. Zudem lassen sie sich einfach wiederverwenden und reduzieren damit Redundanzen. Die parallele Logik sollte gebündelt in der Kernelfunktion vorliegen. Indem alle weiteren Funktionen keinen parallelen Code enthalten, werden sie unabhängig von der GPU. Auf diese Weise werden sie auf der CPU testbar (NVIDIA, 2016a). Um die Wiederverwendbarkeit des GPU-Codes zu gewährleisten, ist eine separate Kompilierung der Programmteile notwendig. Was in C- und C++-Programmen bereits lange ein essentieller Bestandteil ist, wird in CUDA erst seit Version 5.0 ermöglicht. Um den Device-Code im weiteren Schritt wieder gegeneinander zu linken, ist außerdem eine Grafikkarte mit einer Compute Capability 2.0 erforderlich (Scudiero, 2014).

3 Gittergenerierung

Als Berechnungsgrundlage einer Strömungssimulation mittels LBM dient ein kartesisches Gitter. Zu Beginn gilt es dieses zu initialisieren. Jeder Knoten auf dem Gitter muss dafür in einen gültigen Zustand gebracht werden. Üblicherweise werden dafür in einem ersten Schritt alle Knoten innerhalb des Gitters auf den Zustand *fluid* gesetzt. Der LBM-Solver errechnet während der Simulation in diskreten Zeitschritten jeweils die Geschwindigkeiten und den Druck an jedem *fluid* Knoten. Des weiteren müssen Randbedingungen an den Außenknoten des Gitters gesetzt werden. Die Knoten können unter anderem mit *Geschwindigkeits-*, *Wand-* oder *Periodischen-RB* initialisiert werden (Linxweiler, 2011). Abbildung 4 illustriert einen Strömungskanal in 2D mit einer möglichen Initialisierung. Im Inneren des Kanals liegen die als *fluid* deklarierten Knoten. An den Rändern wurden *Wand-*, bzw. *Geschwindigkeits-*Knoten gesetzt.

Zusätzlich soll das Gitter beliebige Geometrien abbilden, wobei die Ränder der Geometrie ebenso mit einer Randbedingung initialisiert werden. Der in diesem Paper beschriebene Algorithmus befasst sich mit der Abbildung von Oberflächenvermaschungen aus Dreiecken (STL²³) auf ein kartesisches Gitter. Das Verfahren detektiert jeweils eine Reihe von Knoten innerhalb der STL und setzt die Knoten daraufhin auf die gewünschte Randbedingung.

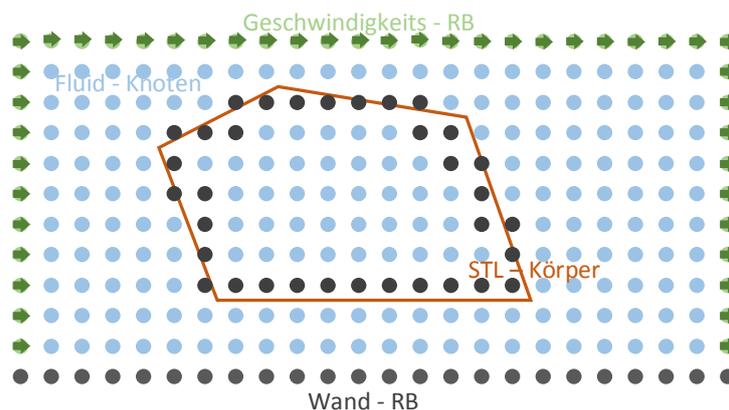


Abb. 4: Abbildung eines STL - Körper auf ein 2D Gitter

3.1 Bounding Box

Die Implementierung auf der Grafikkarte wurde so realisiert, dass jeweils ein Thread alle für ein Dreieck relevanten Knoten prüft. Eingangs wird eine umhüllende Box um das betrachtete Dreieck errechnet (vgl. Abb. 5). Der Thread iteriert im Anschluss über alle Knoten innerhalb dieser Bounding Box. Dieses Verfahren lässt sich sehr effizient parallelisieren. Die Effizienz der Berechnung ist allerdings stark abhängig von der Qualität und Form der Eingangsdaten.

3.2 Unterhalb der Dreiecksebene

Die Ebene des Dreiecks teilt die Bounding Box in einen inneren und äußeren Bereich. Für jeden Knoten in der Bounding Box wird nun im zweiten Schritt geprüft, ob er im inneren Volumen

²³ engl.: STereoLithography oder Standard Tessellation Language

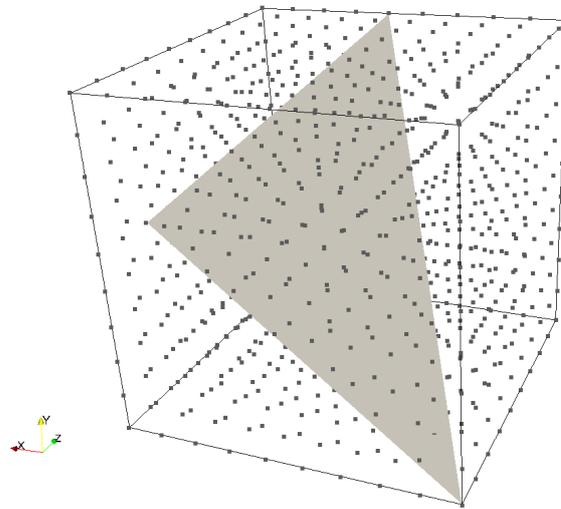


Abb. 5: Bounding Box mit einem Dreieck

liegt. Zu diesem Zweck wird der senkrechte Abstand des Knotens zu der Ebene berechnet. Ist der Abstand größer oder gleich Null, so liegt der Punkt innerhalb der Ebene. In Abbildung 7 sind unter anderem die erforderlichen Größen zur Ermittlung des senkrechten Abstandes dargestellt.

Der aktuell untersuchte Knoten ist als P_4 definiert. Die Richtungsvektoren von den Eckpunkten des Dreiecks zu P_4 ergeben sich dementsprechend zu

$$v_{3+i} = P_4 - P_i \quad i = 1, 2, 3. \quad (1)$$

Der gesuchte senkrechte Abstand s lässt sich ableiten aus dem Vektor v_4 und der Normalen n zu

$$s = - \langle v_4, n \rangle . \quad (2)$$

3.3 Auflösung von spitzen Winkeln

Die Gittergenerierung erfordert eine geschlossene Dreiecksvermaschung als Input. Von einer geschlossenen STL kann gesprochen werden, wenn zu jedem Dreieck drei benachbarte Dreiecke vorhanden sind. Das betrachtete Dreieck teilt sich jeweils mit jedem Nachbarn eine Kante. Es ist zu prüfen, ob sich ein Knoten innerhalb von zwei benachbarten Dreiecken befindet.

Abbildung 6 zeigt in 2D die relevanten Größen zur Quantifizierung eines Punktes bei einem spitzen und stumpfen Winkel (α) zwischen zwei Dreiecken. Der Winkel α charakterisiert dabei den halben Winkel zwischen den Kanten von zwei benachbarten Dreiecken. Zur Ermittlung von α wird anfangs der Vektor s_{12} ermittelt, der die Schwerpunkte beider Dreiecke verbindet.

$$s_{12} = s_1 - s_2 \quad (3)$$

Danach wird das Skalarprodukt des eben ermittelten Vektors mit der Normalen n_2 errechnet. Es wird die Ergebnisvariable χ eingeführt. Sie nimmt einen positiven Wert an, falls die Normalen

der beiden betrachteten Dreiecken zueinander gerichtet sind. Zeigen sie in entgegengesetzte Richtungen nimmt χ einen negativen Wert an (vgl. Abb. 6).

$$\chi = \langle \mathbf{n}_2, \mathbf{s}_{12} \rangle \quad (4)$$

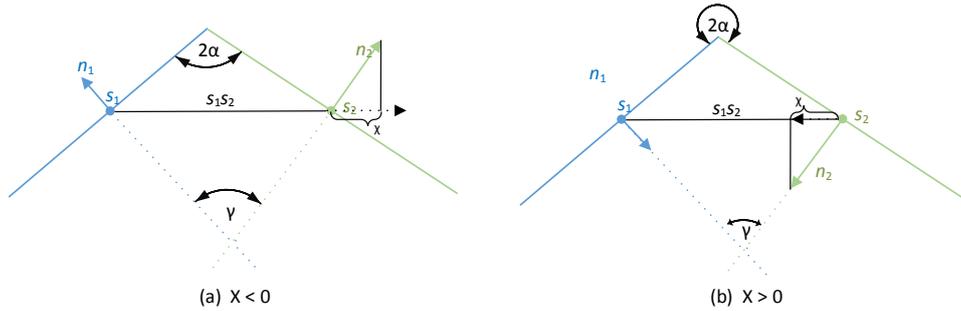


Abb. 6: Ermittlung des Faktors χ mit in (a) voneinander weg zeigenden Normalen und in (b) zueinander zeigenden Normalen basierend auf Koliha (2013).

Der aufgespannte Winkel γ zwischen den Vektoren \mathbf{n}_1 und \mathbf{n}_2 ergibt sich zu

$$\gamma = \cos^{-1} \langle \mathbf{n}_1, \mathbf{n}_2 \rangle . \quad (5)$$

Mithilfe der Variable χ kann α errechnet werden zu

$$2\alpha = \begin{cases} 180^\circ - \gamma & : \chi < 0 \\ 180^\circ & : \chi = 0 \\ 180^\circ + \gamma & : \chi > 0 \end{cases} . \quad (6)$$

Der Winkel β , illustriert in Abbildung 7, liegt zwischen den beiden Vektoren \mathbf{q}_i und \mathbf{r}_i . Ein Knoten liegt nun innerhalb der geschlossenen STL, wenn die Winkel β_i jeweils kleiner sind als die im vorherigen Schritt ermittelten Winkel α_i . Zur Ermittlung des Vektors \mathbf{r}_i ist es zunächst

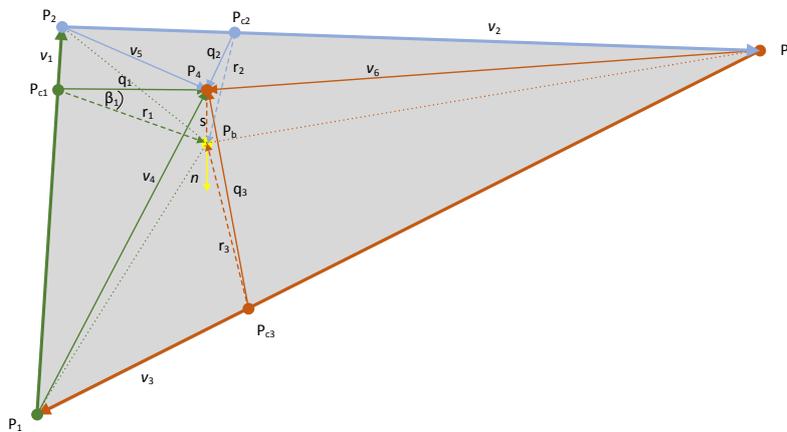


Abb. 7: relevante Größen zur Quantifizierung eines Knotens basierend auf Koliha, 2013

erforderlich den Punkt P_b zu errechnen. Dieser ergibt sich zu

$$P_b = P_4 + s * b. \quad (7)$$

Zusätzlich sind die Punkt P_{ci} zu ermitteln, sodass sich der kürzesten Abstand zum Punkt P_b ergibt. Dafür ist die orthogonale Projektion des Vektors v_{3+i} auf den Vektor v_i zu errechnen:

$$t_i = \frac{\langle v_{3+i}, v_i \rangle}{\langle v_i, v_i \rangle} v_i \quad i = 1, 2, 3 \quad (8)$$

Folgend kann der Punkt mit dem kürzesten Abstand zu P_b bestimmt werden zu

$$P_{ci} = P_i + t_i \quad i = 1, 2, 3. \quad (9)$$

Daraus ergeben sich die Vektoren zwischen P_{ci} und P_b zu

$$r_i = P_b - P_{ci} \quad i = 1, 2, 3. \quad (10)$$

Der Vektor von P_{ci} zu P_4 ergibt sich zu

$$q_i = P_4 - P_{ci} \quad i = 1, 2, 3. \quad (11)$$

Der Winkel β zwischen den Vektoren q_i und r_i ergibt sich anschließend zu

$$\beta = \cos^{-1} \left(\frac{\langle q_i, r_i \rangle}{\|q_i\| \|r_i\|} \right) \quad i = 1, 2, 3. \quad (12)$$

3.4 Außerhalb des Dreiecks

In Kapitel 3.1 wurden Knoten innerhalb der Bounding Box einem äußeren und inneren Volumen zugeordnet. Im letzten Schritt müssen die Knoten ausgeschlossen werden, die zwar im inneren Volumen der Bounding Box liegen, jedoch fälschlicherweise im vorherigen Schritt nicht als außerhalb eines spitzen Winkels deklariert wurden. Diese Knoten werden durch den Vergleich der Winkel nicht als außerhalb detektiert, da ein Knoten gespiegelt an der Kante des Dreiecks zu dem gleichen Winkel β resultiert (vgl. Abb. 8).

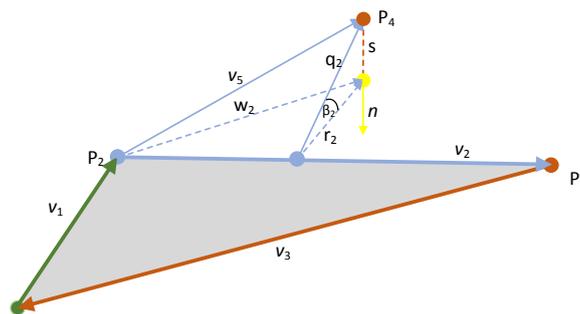


Abb. 8: Bestimmung von Punkten die nicht über halb der Dreiecksfläche liegen basierend auf Koliha (2013)

Dafür wird der Vektor w_i definiert zu

$$w_i = P_b - P_i \quad i = 1, 2, 3. \quad (13)$$

Das Kreuzprodukt des Vektors w_i ergibt mit dem Kantenvektor des Dreiecks

$$t_i = w_i \times v_i \quad i = 1, 2, 3. \quad (14)$$

Der Vektor t_i bildet mit w_i und v_i ein Rechtssystem. Liegt der betrachtete Knoten nun außerhalb der Dreiecksfläche, so sind der Normalenvektor und der Vektor t_i gleichgerichtet. Andererseits sind in dem Fall, dass der Knoten innerhalb der Dreiecksfläche liegt, die Normale und t_i entgegengerichtet. Das führt zu dem Skalar g_i :

$$g_i = \langle t_i, n \rangle \quad i = 1, 2, 3 \quad (15)$$

Dieses nimmt einen positiven Wert an, sofern der Knoten außerhalb des Dreiecks liegt. Befindet er sich innerhalb nimmt das Skalar einen negativen Wert an. Danach ist zu prüfen, ob für jeden Knoten alle g_i kleiner als Null sind und er somit innerhalb der Fläche liegt.

4 Implementierung

Die Informationen aus dem STL-File müssen möglichst effizient auf die Grafikkarte transferiert werden. Der Kopiervorgang kann bei großen Datenmengen einen erheblichen Anteil an der Gesamtlaufzeit einnehmen. Daher sollten so wenig Daten wie möglich auf die GPU übertragen werden. Zur Abbildung der STL-Daten werden dafür C-struct's erstellt (illustriert in Abb. 9). Die Struktur `Vertex` repräsentiert einen Eckpunkt des Dreiecks und beinhaltet die drei Koordinaten. Die Struktur `Triangle` hingegen besteht aus vier Attributen vom Typ `Vertex`, welche die drei Eckpunkte des Dreiecks sowie die Normale beschreiben. Bei einer einfachen

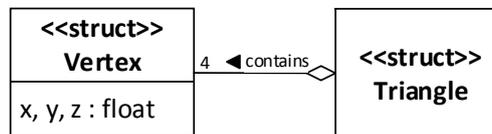


Abb. 9: UML-Diagramm der STL-Datenstruktur

Genauigkeit der Dreieckskoordinaten hat ein `Triangle struct` somit die Größe von 48 Bytes. Zu Beginn der Gittergenerierung werden die Strukturen erstellt und mit den Daten aus dem STL-File initialisiert.

Des Weiteren wird zur Gittergenerierung die Information benötigt, welche Dreiecke sich jeweils eine Kante teilen. Dies muss ermittelt und auf die GPU kopiert werden. Insgesamt ergibt das demzufolge einen Speicherverbrauch auf der Grafikkarte von 60 Bytes pro Dreieck. Zur Repräsentation des Gitters wird ein Array vom Datentyp `char` angelegt. Der Speicher für das Array muss im Vorfeld auf der GPU allokiert werden. Da das Gitter auf der GPU errechnet wird, sind keine weiteren Kopiervorgänge notwendig.

Im Anschluss an die Allokier- und Kopiervorgänge auf die GPU werden zwei Kernel gestartet. Ein erster Kernel ist für die Initialisierung des Gitters mit `Fluid`-Knoten verantwortlich. Der zweite startet den eigentlichen Gittergenerierungsprozess. Der Algorithmus aus Kapitel 3 kann dann auf der GPU ausgeführt werden. Eine Implementierung wurde folgendermaßen realisiert:

Listing 1:

```
foreach triangle in Triangles
  create BoundingBox;
  foreach node in BoundingBox
    if (getPerpedicularDistanceFrom ( triangle , node ) >= 0)
      calculate g1, g2, g3;
      if (g1 <= 0 && g2 <= 0 && g3 <=0)
        if (areAllBetaSmallerThanAlpha ())
          setNodeToBoundaryCondition ();
```

Folgend soll die Gittergenerierung eines Golf GTIs, bestehend aus knapp 10.000 Dreiecken, betrachtet werden. Der GTI wird auf ein Gitter mit ca. 25 Millionen Knoten abgebildet. In Abbildung 10 sind auf der rechten Seite die Oberfläche der STL und auf der linken Seite sind die ermittelten `Solid`-Knoten der Gittergenerierung abgebildet.

Ingesamt werden für das Gitter, die Dreiecke, sowie die Nachbarinformationen 25 MB an Speicherplatz auf der GPU reserviert.

Die Anzahl an Threads wird zu Beginn mit 512 Threads je Block festgelegt. Bei einer Parallelisierung von einem Thread je Dreieck muss der Gittergenerierungs-Kernel mit 20 Blöcken und 512 Threads gestartet werden. Die Berechnung des Gitters dauert auf einer Tesla C2070 0,35 Sekunden, bzw. mit allen Allokier- und Kopiervorgängen sowie der Initialisierung des Gitters 0,618 Sekunden.

Die gleiche Gittergenerierung auf der CPU, parallelisiert mit OpenMP auf 8 Kernen, dauert hingegen 3,7 Sekunden. Das ergibt für dieses Beispiel eine sechsfache Beschleunigung der GPU gegenüber der CPU.

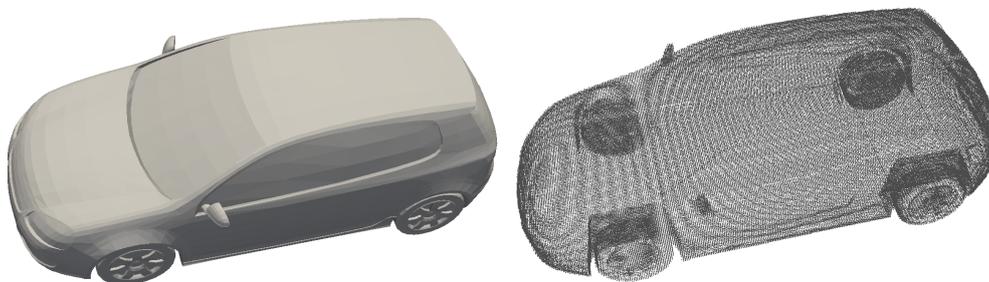


Abb. 10: Gittergenerierung des Golf GTIs, links die Input STL-File, rechts die detektierten `Solid`-Knoten innerhalb der Geometrie

5 Fazit & Ausblick

In diesem Paper wurde eine effiziente parallele Implementierung eines Gittergenerators auf der Grafikkarte gezeigt.

Zu Beginn wurde kurz auf die Entwicklung und die Funktionsweise von GPGPUs eingegangen. Dabei wurde erläutert wie man auch auf der Grafikkarte elegante und trotzdem effiziente Architekturen schaffen kann. Im darauf folgendem Kapitel wurde der Algorithmus zur Gittergenerierung detailliert erläutert. Als Parallelisierungsansatz wird immer jeweils eine Bounding

Box um ein Dreieck gelegt. Ein Thread iteriert anschließend über alle Knoten in dieser Box und prüft, ob der Knoten sich innerhalb der Oberflächenvermaschung befindet.

Durch den gewählten Algorithmus können Gittergenerierungen effizient vorgenommen werden. Bislang wurde der Gittergenerator in das Simulationsframework **DC-Steering** (Linxweiler, 2011) eingebunden. In diesem ist es zur Simulationslaufzeit möglich Veränderungen an der umströmten Geometrie vorzunehmen. Jede Änderung erfordert jeweils eine neue Generierung des Gitters, die durch die parallele Implementierung keine wesentlichen Wartezeit entstehen lässt.

In der aktuellen Implementierung der Randbedingung liegt eine Wand genau zwischen zwei Knoten. Es ist geplant als weiteren Entwicklungsschritt die exakten Abstände der äußeren Gitterknoten zu der Dreiecksfläche zu ermitteln. Zusätzlich soll an vorgegeben Stellen die Gitterweite verkleinert werden (Gitterverfeinerung), um Gebiete höherer Informationsdichte feiner aufzulösen zu können.

Literatur

- Beck, K. (2003). *Test-driven development: By example*. The Addison-Wesley signature series.
- Carter, J. (2015). *Points and Figures. Moore's Law*. <http://pointsandfigures.com/2015/04/18/moores-law/> (besucht am 27.06.2016).
- Gamma, E. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley professional computing series.
- Koliha, N. (2013). *Computation and Real-Time Rendering of Complex Multiphase Flows on GPGPU Clusters*. Hamburg University of Technology, Institute for Fluid Dynamics und Ship Theory.
- Krafczyk, M. (2001). „Gitter-Boltzmann-Methoden – von der Theorie zur Anwendung“. Habil. Technische Universität München.
- Linxweiler, J. (2011). „Ein integrierter Softwareansatz zur interaktiven Exploration und Steuerung von Strömungssimulationen auf Many-Core-Architekturen“. Diss. Braunschweig: Technische Universität Carolo-Wilhelmina.
- Martin, R. C. (2009). *Clean code: A handbook of agile software craftsmanship*. London: Pearson Education.
- Martin, R. C. und Martin, M. (2008). *Agile Software Development: Principles, Patterns and Practices*. Robert C. Martin series. Prentice Hall PTR.
- NVIDIA (2016a). *Cuda C Best Practices Guide*. <http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/%5C#axzz41qhkLuP3> (besucht am 27.06.2016).
- NVIDIA (2016b). *CUDA C Programming Guide v7.5*. <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html%5C#axzz41qhkLuP3> (besucht am 27.06.2016).
- NVIDIA (2016c). *GPU Computing*. <http://www.nvidia.de/object/gpu-computing-de.html> (besucht am 27.06.2016).
- Scudiero, T. (2014). *Separate Compilation and Linking of CUDA C++ Device Code*. <https://devblogs.nvidia.com/parallelforall/separate-compilation-linking-cuda-device-code/> (besucht am 27.06.2016).
- Yourdon, E. und Constantine, L. L. (1979). *Structured design: Fundamentals of a discipline of computer program and systems design*. Yourdon Press computing series.

Effiziente Simulation turbulenter Windströmungen im urbanen Raum

Stephan Lenz

*Institut für rechnergestützte Modellierung im Bauingenieurwesen, TU Braunschweig,
Pockelsstr. 3, 38106 Braunschweig, Deutschland
E-Mail: stephan.lenz@tu-bs.de*

Abstract Im Rahmen dieser Arbeit wird untersucht, inwiefern die Cumulante Gitter-Boltzmann Methode (Geier et al., 2015) in der Lage ist, turbulente Strömungen im urbanen Raum zu simulieren. Dazu wurden unterschiedliche Windrichtungen für ein Stadtviertel in Basel simuliert. Die Ergebnisse werden mit einer Referenzsimulation (Giometto et al., 2016) und realen Messungen der BUBBLE Kampagne (Rotach et al., 2005) verglichen. Es wird gezeigt, dass die Cumulante Gitter-Boltzmann Methode ohne komplexe Turbulenzmodelle sehr genaue Ergebnisse für Geschwindigkeitsprofile und turbulente Größen generiert. Weiterhin können die Simulationen, dank einer effizienten GPU Implementierung, auf handelsüblichen Desktop-Computern durchgeführt werden. Somit werden hoch aufgelöste Windsimulationen auch für kleine und mittlere Unternehmen zugänglich.

Keywords: Computational Fluid Dynamics, Gitter-Boltzmann Methode, GPU-Computing, Turbulente Strömungen, Urban Wind Simulation

1 Einleitung

Nach Angaben der UN lebten im Jahr 2014 mehr als die Hälfte der Weltbevölkerung in Städten (Nations, 2014). In Europa (73%) und Nordamerika (82%) ist dieser Anteil noch wesentlich höher. Die steigende Bevölkerungszahl im urbanen Raum geht mit einer steigenden Schadstoffemission einher. Die Bebauungsdichte hat hierbei einen großen Einfluss auf die Luftzirkulation und den Schadstofftransport. Windsimulationen im urbanen Raum bieten das Potential bereits in der Planungsphase einzelner Gebäude, Bezirke und ganzer Städte den Schadstofftransport mit zu berücksichtigen. Hierfür ist eine feine Diskretisierung ($\Delta x \leq 1m$) zwischen den Gebäuden notwendig.

Die meisten bisher veröffentlichten Arbeiten auf diesem Gebiet nutzen abstrakte Formen als Stadtmodelle. Z. Xie und Castro (2006) haben eine Windströmung über eine reguläre Anordnung von Quadern zufälliger Höhe simuliert und herausgefunden, dass RANS²⁴-Simulationen (gegenüber LES²⁵) eher ungeeignet für die Berechnung urbaner Windströmungen sind. Michioka et al. (2011) haben verschiedene Anordnungen von Sattel- und Flachdächern vor und

²⁴RANS: Reynolds-Averaged-Navier-Stokes, siehe Kap. 2.1

²⁵LES: Large-Eddy-Simulation, siehe Kap. 2.1

hinter Straßenschluchten untersucht. LES hat auch hier bessere Ergebnisse gezeigt. Einen weiteren Schritt gehen Kukačka et al. (2014). Sie haben die Windströmung über abstrakte dreidimensionale Wohnblöcke, wie sie sich in vielen europäischen Städten finden, untersucht. Die Simulationsergebnisse werden hier auch mit Windkanalmessungen verglichen.

Simulationen für reale urbane Räume existieren bisher nur bedingt. Die Windströmung über einem Londoner Viertel wurde von Z.-T. Xie und Castro (2009) untersucht. Die Ergebnisse werden allerdings nur anhand von Windkanalmessungen validiert. Eine Validierung einer Windsimulation anhand von Messungen in einer realen Stadt findet sich bei Nozu et al. (2008). Das Berechnungsgebiet ist dabei das Umfeld mehrerer Hochhäuser in Tokio. Die Windgeschwindigkeitsmessungen beschränken sich auf zwei Punkte auf diesen Gebäuden. Wesentlich detailliertere Messergebnisse liefert die BUBBLE (Basel UrBan Boundary Layer Experiment) Kampagne (Rotach et al., 2005). Ziel dieser groß angelegten Messreihe war ein besseres Verständnis der Prozesse in der urbanen Grenzschicht. Ein Bestandteil der Kampagne war die Messung von Windgeschwindigkeiten und turbulenten Größen in der Sperrstrasse in Basel. Für diese Messungen wurde ein 32 m hoher Turm mit sechs Anemometern errichtet. Da viele turbulente Transportgrößen nicht direkt gemessen werden können haben Giometto et al. (2016) LES Simulationen durchgeführt um diese Größen zu quantifizieren. Das Berechnungsgebiet umfasst die Sperrstrasse, sowie die umliegenden Straßen und Gebäude. Die Daten der BUBBLE-Messungen wurden genutzt um diese Simulationen zu validieren.

Klassische Strömungslöser, wie sie für die oben genannten Simulationen genutzt wurden, basieren auf der Diskretisierung der Navier-Stokes Gleichungen, welche kontinuierliche Fluide beschreiben. Der Berechnungsaufwand von Simulationen, basierend auf diesen Lösern, wächst mit steigender Diskretisierungsgenauigkeit stark an. Daher sind in der Regel Computer-Cluster notwendig um entsprechende Simulationen zu realisieren. Diese sind teuer in Anschaffung und Betrieb und somit häufig für planende Ingenieurbüros und Behörden nicht zugänglich.

Die Boltzmann-Gleichung hingegen beschreibt statistisch das kinetische Verhalten von Gasen. Die daraus abgeleitete Gitter-Boltzmann Methode (eng.: Lattice-Boltzmann-Method, LBM) ist ebenfalls ein Lösungsverfahren für Navier-Stokes Strömungen. Die LBM eignet sich sehr gut für die Implementierung auf Grafikprozessoren (GPUs) und profitiert somit von der rasanten Entwicklung auf dem Gebiet des GPU Computings. GPUs erlauben hohe Rechenleistung bei relativ geringen Kosten. Eine modifizierte LBM (CumLBM, (Geier et al., 2015)) ist in der Lage stabile und genaue Ergebnisse für turbulente Strömungen zu generieren.

In dieser Arbeit wird am Beispiel der Sperrstrasse in Basel untersucht, inwiefern CumLBM in der Lage ist, turbulente Strömungen im urbanen Raum zu simulieren. Kapitel 2 gibt eine kurze Einführung in die genutzten numerischen Methoden. In Kapitel 3 wird der Aufbau der Simulation inklusive der eingestellten Parameter aufgezeigt. Die Ergebnisse werden dann in Kapitel 4 mit Giomettos Simulationsergebnissen und den Messungen der BUBBLE Kampagne verglichen. Abschließend wird die Arbeit in Kapitel 5 zusammengefasst.

2 Numerische Simulation

2.1 Simulation turbulenter Strömungen

Für die Simulation von Strömungen gibt es eine große Bandbreite an numerischen Methoden. Der klassische Ansatz ist dabei die Diskretisierung der Navier-Stokes-Gleichungen (NSG). Für

die Simulation von Windströmungen im urbanen Raum kann die inkompressible Form verwendet werden, da die auftretenden Windgeschwindigkeiten weit unter der Schallgeschwindigkeit liegen und thermische Ausdehnung zunächst nicht von Interesse ist. In der folgenden Gleichung ist \underline{U} der Geschwindigkeitsvektor, p der Druck und ν die kinematische Viskosität.

$$\begin{aligned} \nabla \underline{U} &= 0 \\ \frac{\partial}{\partial t} \underline{U} + (\underline{U} \cdot \nabla) \underline{U} &= -\frac{1}{\rho} \nabla p + \nu \Delta \underline{U} \end{aligned} \quad (1)$$

Die inkompressiblen NSG (Gl. 1) sind eine mathematische Formulierung der Massen- und Impulserhaltung. Sie bilden ein System partieller, nichtlinearer, differential-algebraischer Gleichungen. Der differential-algebraische Charakter zeigt sich im impliziten Vorkommen des Drucks in der Impulsgleichung. Zur Lösung der inkompressiblen NSG muss daher eine explizite Druckgleichung hergeleitet werden. Diese hat den numerischen Nachteil, dass sie elliptischer Natur ist, und somit in jedem Zeitschritt ein Gleichungssystem für das gesamte Berechnungsgebiet gelöst werden muss. Die Nichtlinearität liegt in dem sogenannten konvektiven Term $(\underline{U} \cdot \nabla) \underline{U}$. Konvektiv bedeutet in diesem Zusammenhang, dass der Impuls durch die Bewegung des Fluides transportiert wird. Durch das Fließen breiten sich Informationen (Störungen) anisotrop aus.

Zur Diskretisierung der NSG werden häufig Finite-Volumen-Methoden genutzt, da diese den Erhaltungscharakter der NSG sehr gut darstellen. Möglich sind aber auch Finite-Differenzen, Finite-Elemente und Spektral Methoden. Die Diskretisierung hat zur Folge, dass die Lösung nicht als kontinuierliche Funktion, sondern lediglich an bestimmten Punkten in Raum (Knoten) und Zeit (Zeitschritt) berechnet wird. Der Abstand dieser Punkte ist die Schrittweite (Δx im Raum und Δt in der Zeit).

Die Simulation turbulenter Strömungen bringt eine weitere Herausforderung mit sich. Überwiegt der konvektive den diffusiven (viskosen) Impulstransport deutlich, entsteht eine turbulente Strömung.

Für einen äußeren Betrachter erscheint Turbulenz als chaotische Bewegung, da eine turbulente Strömung eine sehr breite Spanne an Größenskalen enthält. Direkte numerische Simulation (DNS) turbulenter Strömungen ist möglich. Der numerische Ressourcenaufwand ist für reale Strömungen allerdings viel zu groß, da die räumliche Schrittweite Δx kleiner als die kleinsten Wirbel der Strömung sein muss. Auf der anderen Seite sind die kleinen Skalen in der Regel nicht besonders interessant und können modelliert werden. Für die Modellierung der Turbulenz gibt es zwei unterschiedliche Herangehensweisen.

Die turbulente Strömung kann in eine laminare über die Zeit gemittelte Strömung $\overline{(\cdot)}$ und turbulente, zeitabhängige Fluktuationen $(\cdot)'$ zerlegt werden. Dieser Ansatz resultiert in den Reynolds-Averaged-Navier-Stokes (RANS) Gleichungen. Diese Gleichungen beschreiben die zeitlich gemittelte laminare Strömung. Die Turbulenz geht als modellierte turbulente Viskosität in die Gleichungen ein. Für die Berechnung turbulenter Windströmungen ist RANS nicht geeignet (Michioka et al., 2011; Z. Xie und Castro, 2006). Trotzdem bietet diese Zerlegung eine gute Grundlage um turbulente Strömungen zu analysieren.

Einen anderen Ansatz verfolgen sogenannte Large-Eddy-Simulations (LES). Die Idee hinter LES entstammt der Komolgorov'schen Kaskade der turbulenten kinetischen Energie. Der Energieeintrag in die turbulenten Moden findet auf den größten Skalen statt. Die Energie wird dann sukzessive zu kleineren Skalen hin transportiert, um schlussendlich auf den kleinsten Skalen

zu dissipieren. In LES wird die Strömung in eine zeitabhängige, energiereiche Strömung und in eine rein dissipative Strömung zerlegt. Nur die erste der beiden wird von der numerischen Methode aufgelöst. Letztere wird dann als turbulente Viskosität modelliert. Die Trennung der beiden Strömungen erfolgt durch einen Filter. Die Filterweite ist in der Regel größer als das numerische Gitter. Es ist in bestimmten Fällen aber auch möglich das Gitter selbst als Filter zu verwenden. Auch kann auf ein explizites Modell für die turbulente Viskosität verzichtet werden, wenn die numerische Viskosität des Lösungsverfahrens bestimmte Voraussetzungen erfüllt (Fröhlich, 2006, Kapitel 5.4 und 5.6). Diese Verfahren werden als implizite LES bezeichnet. Die in dieser Arbeit genutzte Simulation ist eine solche implizite LES.

2.2 Gitter-Boltzmann-Methoden

Die im vorherigen Abschnitt eingeführten Navier-Stokes-Gleichungen leiten sich aus der Kontinuität des Fluides und der Erhaltung von Masse und Impuls her. Eine alternative Beschreibung des Verhaltens eines Fluides ist die aus der Gas-Kinetik hergeleitete Boltzmann-Gleichung. Die Gas-Kinetik betrachtet das Fluid als Summe vieler Teilchen, die miteinander kollidieren. Der Strömungszustand wird in der Gas-Kinetik als Verteilungsfunktion $f(x, t, u)$ dargestellt. Diese gibt an, mit welcher Wahrscheinlichkeit sich zum Zeitpunkt t ein Teilchen mit der Geschwindigkeit u am Ort x befindet. Die Boltzmann-Gleichung ist eine Transportgleichung für diese Verteilungsfunktion:

$$\underbrace{\frac{\partial f}{\partial t} + u \cdot \nabla f}_{\text{Transport}} = \underbrace{\Omega(f)}_{\text{Kollision}} \approx \underbrace{\frac{f^{eq} - f}{\tau}}_{\text{BGK-Näherung}} \quad (2)$$

Die rechte Seite der Boltzmann-Gleichung ist das Kollisionsintegral $\Omega(f)$. Dieses ist relativ komplex und wird in der Praxis daher durch das Bhatnagar-Gross-Krook (BGK) Modell ersetzt (Bhatnagar et al., 1954). Das Modell basiert auf der Annahme, dass viele Kollisionen das Gas in einen Gleichgewichtszustand f^{eq} führen. Die Geschwindigkeit, mit der sich das Gas diesem Gleichgewichtszustand nähert, wird durch die Relaxationszeit τ beschrieben.

Die Verbindung zwischen den mikroskopischen Größen f und u und den makroskopischen Größen ρ und U ²⁶ erfolgt über die Bildung von Momenten der Verteilungsfunktion:

$$\rho = \int_{-\infty}^{\infty} f(x, t, u) du \quad , \quad \rho U = \int_{-\infty}^{\infty} u f(x, t, u) du \quad (3)$$

Die makroskopischen Navier-Stokes-Gleichungen können mittels der Chapman-Enskog-Entwicklung sowohl von der Boltzmann-Gleichung, als auch von der BGK-Boltzmann-Gleichung abgeleitet werden.

Zur numerischen Lösung mittels Gitter-Boltzmann-Methoden (eng.: Lattice-Boltzmann-Method, LBM) werden nicht nur Raum und Zeit diskretisiert. Auch der kontinuierliche Geschwindigkeitsraum u wird auf eine finite Anzahl von Geschwindigkeitsvektoren u_i reduziert. Für die Simulation in dieser Arbeit wird ein dreidimensionales Modell mit 27 diskreten Geschwindigkeiten u_i verwendet. In der LBM-Literatur werden solche Modelle mit D3Q27 bezeichnet. Die Geschwindigkeitsvektoren sind so gewählt, dass sie einen Knoten in einem

²⁶ u ist die Geschwindigkeit eines Gasmoleküls. U ist die makroskopisch messbare Geschwindigkeit des Fluids.

gleichförmigen Gitter mit allen 26 Raumnachbarn und dem Knoten selbst verbinden (Moore Nachbarschaft). Analog zu den 27 Geschwindigkeiten ergeben sich 27 diskrete Verteilungsfunktionen f_i . Die diskretisierte BGK-Boltzmann-Gleichung (Gitter-Boltzmann-Gleichung) hat die folgende Form:

$$f_i(x + u_i \Delta t, t + \Delta t) = f_i(x, t) + \frac{\Delta t}{\tau_{num}} (f_i^{eq}(x, t) - f_i(x, t)) \quad (4)$$

Eine bemerkenswerte Eigenschaft der Gitter-Boltzmann-Gleichung ist, dass der Wert im neuen Zeitschritt $t + \Delta t$ nicht für den Ort des betrachteten Knotens x (wie in klassischen numerischen Methoden) berechnet wird. Stattdessen zeigt sich hier direkt der Transportcharakter der Boltzmann-Gleichung. Die berechnete Verteilung bewegt sich mit der zugehörigen Geschwindigkeit u_i zu dem Knotens $x + u_i \Delta t$. Algorithmisch zerlegt dies die Berechnung in die Relaxation der Verteilungen und den Transport der Verteilungen zu den benachbarten Knoten. Die räumliche Abhängigkeit der Gitter-Boltzmann-Gleichung beschränkt sich auf den Transport der Verteilungen zu den nächstliegenden Nachbarn. Eine elliptische Druckgleichung muss nicht gelöst werden, da die Gitter-Boltzmann-Gleichung schwach kompressibel ist und sich Druckstörungen über die Dichteänderungen ausbreiten. Die oben dargestellte Gitter-Boltzmann-Gleichung ist nur stabil für niedrige Reynoldszahlen. Wesentlich bessere Stabilität bei hohen Reynoldszahlen bietet die Cumulante-LBM (CumLBM) von Geier et al. (2015). Die Kollision wird dabei nicht direkt auf die Verteilungen, sondern die Cumulanten der Verteilungen angewendet. Numerische Tests zeigen Stabilität auch noch bei hohen Reynoldszahlen. Außerdem zeichnet sich diese Methode durch wenig numerische Viskosität aus.

2.3 GPU-Computing

Eine besondere Eigenschaft der Gitter-Boltzmann-Methode ist die hohe Datenlokalität (sehr begrenzte räumlich Abhängigkeit). Diese Eigenschaft begünstigt Implementierungen auf Grafikprozessoren (Graphics Processing Unit, GPU). Die Architektur von Grafikkarten ist darauf ausgelegt, viele einfache Aufgaben möglichst gleichzeitig zu erledigen. Der Unterschied zwischen regulären Computerprozessoren (Central Processing Unit, CPU) und GPUs ist in Abbildung 1 dargestellt. CPUs sind für die möglichst effiziente Bearbeitung komplexer (meist sequentieller) Aufgaben ausgelegt. Der Platz auf dem Chip wird daher für große Caches (Zwischenspeicher) und komplexe Kontrollstrukturen genutzt. Die eigentlichen Recheneinheiten (Arithmetic Logic Unit, ALU) nehmen nur einen überschaubaren Bereich des Prozessors ein. GPUs hingegen haben, wegen der Limitierung auf einfache Aufgaben, kleinere Kontrollstrukturen und Caches. Der dadurch frei werdende Platz auf dem Chip wird mit ALUs gefüllt. Moderne GPUs sind daher in der Lage tausende Rechnungen gleichzeitig auszuführen. Ursprünglich diente das dazu, die Pixelwerte (Farben) des Bildschirms zu berechnen. Heutzutage sind Grafikkarten mittels CUDA (Compute Unified Device Architecture, nur NVIDIA GPUs) oder OpenCL (Open Computing Language) frei programmierbar.

Die in dieser Arbeit genutzte Implementierung der CumLBM ist in der Dissertation von Schönherr (2015) beschrieben. Die Berechnung von Kollision und Transport erfolgt dabei massiv parallel auf Grafikkarten.

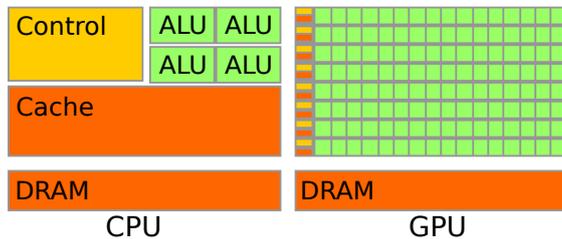


Abb. 1: Vergleich von CPU und GPU Architekturen (Grafik aus der CUDA Toolkit Documentation, <http://docs.nvidia.com/cuda/cuda-c-programming-guide/>)

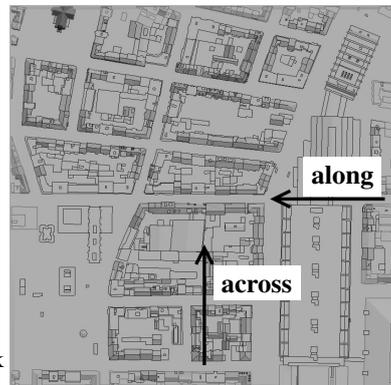


Abb. 2: Das Berechnungsgebiet mit den beiden simulierten Windrichtungen

3 Simulationsaufbau

LBM Simulationen benötigen strukturierte, equidistante Gitter. Das bedeutet, dass der Abstand benachbarter Zellen in allen Raumrichtungen gleich sein muss. Bei vielen Simulationen ist eine hohe Auflösung des Rechengitters (kleines Δx) nur in bestimmten Bereichen notwendig. Um diese Anforderungen zu erfüllen werden mehrere überlappende Rechengitter mit verschiedenen räumlichen Schrittweiten Δx gekoppelt. Das in dieser Arbeit verwendete Rechengitter wurde mit dem Tool LBMHexMesh (Pasquali et al., 2013) erzeugt. Die Gebäude sind in dem Rechengitter explizit, durch Entfernen der Knoten in den Gebäuden, dargestellt.

Abbildung 2 zeigt das genutzte Berechnungsgebiet. Es umfasst einen quadratischen Ausschnitt Basels mit dem Messturm der BUBBLE Kampagne (Rotach et al., 2005) in der Sperrstraße als Mittelpunkt und einer Größe von $512m \times 512m$. Die Höhe des Berechnungsgebietes beträgt $152m$. Die Größe des Berechnungsmodells ist der Referenzsimulation von Giometto et al., 2016 entnommen.

In dieser Arbeit werden drei verschiedene Gitter untersucht. Das grobe Gitter (coarse) hat eine räumliche Schrittweite von $\Delta x = 1m$ am Boden und an den Gebäuden. Nach oben hin vergrößert sich das Gitter in zwei Stufen bis auf $\Delta x = 4m$. Das feine Gitter (fine) startet mit $\Delta x = 0.5m$ und vergrößert sich in drei Stufen ebenfalls bis auf $\Delta x = 4m$ am oberen Rand des Gitters. Das dritte Gitter (uniform) hat eine konstante räumliche Schrittweite von $\Delta x = 1m$. Dies entspricht der Auflösung in der Referenzsimulation. Das grobe Gitter hat ca. 8 Millionen Knoten. Das feine und uniforme Gitter haben jeweils ca. 37 Millionen Knoten.

Die Randbedingungen der Simulation sind periodische Ränder in den beiden horizontalen Richtungen. Das Berechnungsgebiet besteht somit virtuell aus mehrmals dem gleichen Gebiet nebeneinander, was für eine Stadt mit relativ homogener Bebauung eine gute Näherung darstellt. Dadurch produziert die Simulation selbst die Turbulenz im einströmenden Wind. Weitere Randbedingungen sind Haften (keine Geschwindigkeit) am Boden und an den Gebäuden und Slip (keine Reibung, bzw. keine Scherrspannung) an der Oberseite des Berechnungsgebietes. Die Strömung wird von einem konstanten Druckgradienten (Volumenkraft) von $F = 8.56 \cdot 10^{-4} \frac{Pa}{m}$ angetrieben. Die Dichte und die Viskosität entsprechen den realen Werten für Luft: $\rho = 1.2 \frac{kg}{m^3}$ und $\nu = 1.51 \cdot 10^{-5} \frac{m^2}{s}$. Für die Schallgeschwindigkeit wird aufgrund der Rechengenauigkeit

der verwendeten GPUs (single precision) ein künstlicher Wert verwendet. Die resultierenden Machzahlen sind aber weiterhin klein genug, um schwach kompressibel rechnen zu können.

Für jedes Rechengitter wurden des weiteren zwei Windrichtungen simuliert. Die Windrichtungen sind ebenfalls in Abbildung 2 eingezeichnet. Sie entsprechen einer Windrichtung Ost-Nord-Ost (66°) entlang der Sperrstrasse (along) und einer Windrichtung Süd-Süd-West (156°) quer über die Sperrstrasse (across).

4 Ergebnisse

4.1 Das momentane Geschwindigkeitsfeld

In diesem Abschnitt wird zunächst das momentane Geschwindigkeitsfeld untersucht. Abbildung 3 zeigt das momentane horizontale Geschwindigkeitsfeld in der Sperrstrasse auf einer Höhe von $4m$. Die Windgeschwindigkeit auf dieser Höhe ist aufgrund des Reibungswiderstandes des Bodens und der Gebäude relativ gering. Dadurch werden einige Merkmale der turbulenten Strömung sichtbar. Zunächst ist zu beobachten, dass sich auf der linken Seite der Abbildung eine Böe mit höheren Geschwindigkeiten entlang der Straße bewegt. Darüber hinaus sind mehrere turbulente Wirbel sichtbar. Durch die niedrige Geschwindigkeit zwischen den Gebäuden führen diese Wirbel auch zu Rückflüssen an den Gebäuden. Dies entspricht Strömungsablösungen und ist dahingehend interessant, als dass viele andere Arbeiten (z. B. Giometto et al. (2016)) Wandfunktionen an den Gebäuden nutzen.

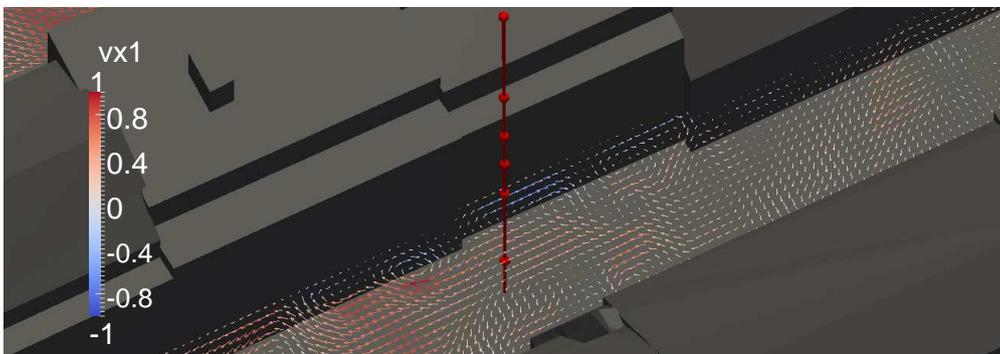


Abb. 3: Momentanes horizontales Geschwindigkeitsfeld der feinen Simulation ($\Delta x = 0.5m$) in einer Höhe von $4m$ in der Sperrstrasse. Der Messturm mit den Messpunkten ist in rot zu sehen. Der Wind kommt von oben rechts (along, 66°).

Wandfunktionen sind Modellierungen der turbulenten Grenzschicht in Wandnähe, da diese oft nicht aufgelöst werden kann. Wandfunktionen sind allerdings in der Regel nur für anliegende Strömungen formuliert. Diese Diskrepanz wird bereits von Tominaga et al. (2008) beschrieben. Die Einflüsse der inkorrekten Wandfunktionen werden dort aber als klein eingeordnet. Im folgenden wird gezeigt, dass die CumLBM, zumindest für normierte Größen, auch ohne Wandfunktion sehr gute Ergebnisse liefert.

In der Windsimulation im Bauingenieurwesen werden häufig Strömungen um einzelne wenige Hochhäuser betrachtet. Die Strömung um schlanke Hochhäuser teilt sich in vier verschiedene Phänomene (Blocken und Carmeliet, 2004). Diese sind in der rechten Grafik in Abbildung 4 zu sehen. Im oberen Bereich fließt der Wind über das Dach des Hauses (1), während sich in den

unteren Bereichen der Wind seinen Weg um die Seiten des Gebäudes sucht (2), ohne vertikal abgelenkt zu werden. Vor dem Hochhaus zeigen sich darüber hinaus steile Abwinde (3). An den unteren Ecken bilden sich stehende Wirbel (4). Alle diese vier Phänomene zeigen sich ebenfalls in der LBM-Simulation. Die linke Grafik in Abbildung 4 zeigt die momentanen Stromlinien um das Hochhaus.

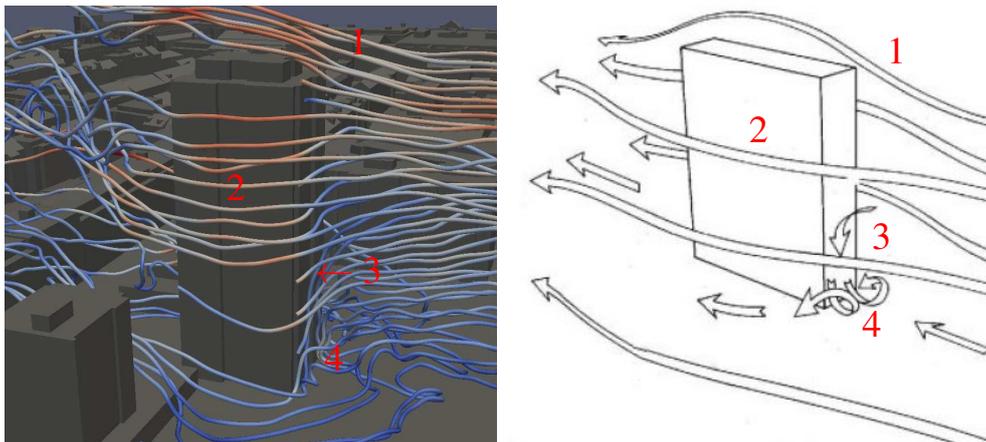


Abb. 4: links: Stromlinien um ein Hochhaus. Die Stromlinien wurden auf dem groben Gitter aus dem momentanen Geschwindigkeitsfeld berechnet. Der Wind kommt von rechts (across, 156°). rechts: Schematische Strömung um ein schlankes Hochhaus (Grafik aus Blocken und Carmeliet (2004), modifiziert)

Dieser Abschnitt zeigt, dass die LBM-Simulation die Turbulenz auflöst und plausible Ergebnisse für die Umströmung von Gebäuden liefert. Validieren lässt sich die Simulation anhand dieser Untersuchungen aber nicht. Daher werden die LBM-Ergebnisse im folgenden Abschnitt mit Messdaten verglichen.

4.2 Auswertung lokaler Profile

In diesem Abschnitt werden nun die LBM-Simulationen quantitativ anhand von Messungen der BUBBLE Kampagne validiert. Zum Vergleich der turbulenten Strömungen wird das Reynolds-Averaging (siehe RANS) verwendet und die Strömung wird in ein zeitliches Mittel und turbulente Fluktuationen zerlegt. Neben dem zeitlichen Mittel verwenden Giometto et al. (2016) ein lokales räumliches Mittel auf horizontalen Ebenen von $6m \times 6m$ zur Berechnung der Profile. Diesem Ansatz wird in dieser Arbeit gefolgt. Durch die verschiedenen Vergrößerungsstufen des Berechnungsgitters können allerdings nicht die selben Ebenen genutzt werden. Für die räumliche Mittlung werden daher 4×4 und 2×2 Knoten um den Ort der Messungen verwendet. Die Geschwindigkeitsprofile ergeben sich dann nach folgender Berechnung:

$$\bar{u}_a(z) = \frac{1}{n_x(z)} \frac{1}{n_t} \sum_{i=1}^{n_x(z)} \sum_{j=1}^{n_t} u_a(x_i, t_j) \quad (5)$$

Dabei ist $n_x(z)$ die Anzahl der Knoten pro Ebene und n_t die Anzahl der gemittelten Zeitschritte. Neben den Geschwindigkeitsprofilen werden auch turbulente Größen untersucht. Dazu

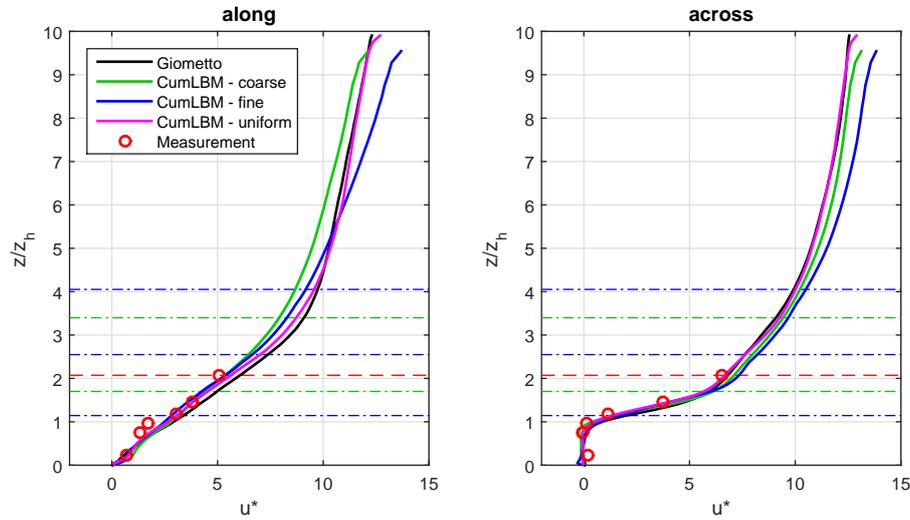


Abb. 5: Windgeschwindigkeitsprofile in Hauptstromrichtung

werden die Varianzen und die Kovarianzen der Geschwindigkeiten benötigt. Die Mittlung erfolgt nach dem gleichen Schema.

$$\overline{u'_a u'_b}(z) = \frac{1}{n_x(z)} \frac{1}{n_t} \sum_{i=1}^{n_x(z)} \sum_{j=1}^{n_t} (u_a(x_i, t_j) - \bar{u}_a(z)) (u_b(x_i, t_j) - \bar{u}_b(z)) \quad (6)$$

Aus der Summe der Geschwindigkeitsvarianzen lässt sich dann die zeitlich gemittelte turbulente kinetische Energie (TKE, \bar{k}) berechnen:

$$\bar{k}(z) = \frac{1}{2} \left(\overline{u'_1 u'_1}(z) + \overline{u'_2 u'_2}(z) + \overline{u'_3 u'_3}(z) \right) \quad (7)$$

Die Simulationen nutzen einen konstanten Druckgradienten, um die Strömung anzutreiben. Dies ist zwar eine gute Näherung an die physikalischen Prozesse, die Wind antreiben. Trotzdem ergeben sich andere Geschwindigkeiten als in den Messungen. Dazu schreiben Giometto et al. (2016), dass die hier untersuchte Strömung unabhängig von Einflüssen der Reynolds-Zahl ist. Dies wird auch von Z. Xie und Castro (2006) bestätigt. Der eigentliche Wert der Windgeschwindigkeit ist hier daher nicht von Interesse. Die untersuchten Profile werden deshalb mit der lokalen Schubspannungsgeschwindigkeit $u_\tau = \sqrt{-\overline{u'_1 u'_3}}$ an der Spitze des Messturmes normiert ($(\cdot)_*$ kennzeichnet eine Größe).

$$\bar{u}_*(z) = \frac{\bar{u}(z)}{u_\tau(z_t)}, \quad \bar{k}_*(z) = \frac{\bar{k}(z)}{u_\tau^2(z_t)} \quad (8)$$

Abbildung 5 zeigt die Geschwindigkeitsprofile der LBM-Simulationen, Giomettos Simulationsergebnisse, sowie die Messdaten der BUBBLE Kampagne. Die gestrichelte rote Linie markiert die Höhe des Messturmes. Die Strich-Punkt Linien markieren die Übergänge der

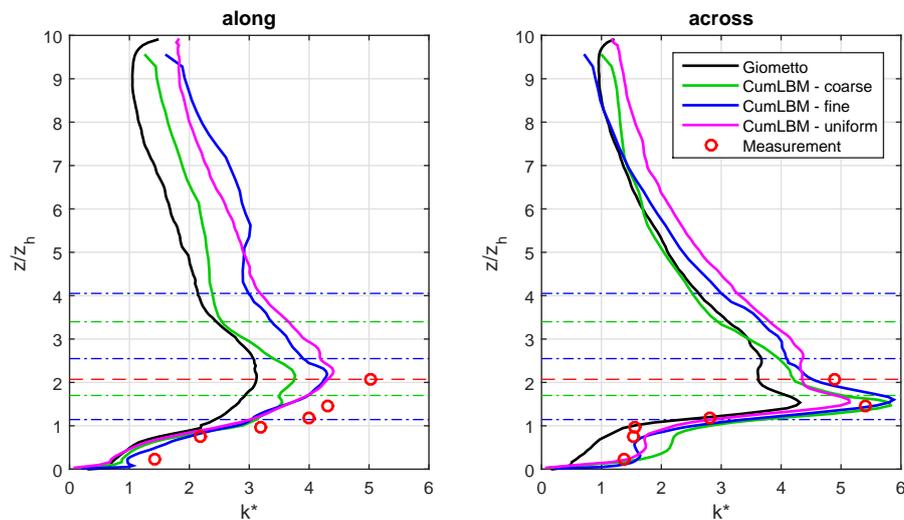


Abb. 6: Profile der turbulenten kinetischen Energie

verschiedenen Vergrößerungsstufen. Die Höhe ist mit der durchschnittlichen Gebäudehöhe $z_h = 15.3m$ normiert.

Die Geschwindigkeitsprofile für beide Windrichtungen liegen sehr nah beieinander. Für die Strömung entlang der Straße (along) liegen die CumLBM Ergebnisse etwas näher an den Messungen. Im oberen Bereich laufen die Simulationen stark auseinander. Da hier aber keine Messdaten vorliegen, kann keine Aussage über die Qualität der Simulationsergebnisse in diesem Bereich getroffen werden. Obwohl alle Simulationen quantitativ sehr gute Ergebnisse zeigen, prognostiziert keine den korrekten qualitativen Verlauf. Die Messungen zeigen einen relativ starken Knick im Profil zwischen dem dritten und vierten Messpunkt von unten. Grund hierfür könnte sein, dass die Simulationen keine Hindernisse (z.B. Fahrzeuge) in der Straße berücksichtigen. Eine andere Erklärung könnte die nicht hinreichende Darstellung der Fluidreibung an den Gebäuden sein. Wandunebenheiten wie Fenster, Türen oder Fallrohre werden nicht modelliert. Bei den Geschwindigkeitsprofilen für den Wind quer zur Straße zeigen sich keine signifikanten Unterschiede zwischen den Simulationen.

Wesentlich größere Unterschiede zeigen sich in Abbildung 6 bei den Profilen der turbulenten kinetischen Energien. Zunächst ist anzumerken, dass für den Wind entlang der Straße (along) alle Simulationen zu wenig TKE vorhersagen. Der Grund dafür liegt zumindest teilweise in der LES. Da nicht die gesamte Turbulenz aufgelöst wird, muss ein Teil der TKE fehlen. Dann ist ersichtlich, dass die CumLBM Simulationen pauschal mehr TKE als Giomettos Simulation auflösen und damit fast überall näher an die gemessenen Daten kommen. Der Hintergrund ist, dass Giometto explizite Feinskalenmodelle verwendet. Diese dissipieren offenbar zu viel TKE. Die CumLBM hingegen nutzt die numerische Viskosität als implizites Feinskalenmodell und scheint zumindest für diesen Testfall ein besseres Maß an TKE zu dissipieren. Die geringere TKE in der groben LBM Simulation (along) erklärt sich durch die größere Auflösung. Bei der unteren grünen Strich-Punkt-Linie springt die Auflösung von $1m$ auf $2m$. Viele kleine Wirbel können darüber nicht mehr aufgelöst werden und die darin enthaltene TKE fehlt abrupt. Dass sich dieses Verhalten nur in der groben Simulation zeigt könnte ein Zeichen dafür sein, dass die anderen Simulationen in den entsprechenden Höhen bereits den größten Anteil der TKE auflösen. Ein weiterer massiver Unterschied zwischen Giomettos Simulation und den

LBM Simulationen zeigt sich bei der Querströmung (across). Giometto sagt hier einen nahezu linearen TKE Verlauf bis zur Höhe der Gebäude voraus. Die Messungen zeigen hingegen eine relativ konstante TKE zwischen den Gebäuden. Die CumLBM Simulationen zeigen dieses Verhalten qualitativ, wobei die feine und die uniforme Simulation auch quantitativ passende Werte vorhersagen. Der Unterschied zwischen grober und uniformer Simulation (welche beide eine Auflösung von $1m$ in diesem Bereich haben) zeigt, dass für die Menge der TKE auch die Auflösung in den höheren Schichten einen starken Einfluss hat.

4.3 Rechenzeiten

Neben der Qualität der Ergebnisse soll auch die Effizienz ein Aspekt in dieser Arbeit sein. Für Nutzer solcher Simulationen sind kurze Rechenzeiten und günstige Hardware ebenso wichtig. Die Referenzsimulation von Giometto (Giometto et al., 2016) wurde auf einem Computer-Cluster durchgeführt und benötigte eine Rechenzeit von ca. 5 Tagen. Zur Messung der Performance wurden alle Simulationen auf einer aktuellen Verbraucher-Grafikkarte (NVIDIA GeForce GTX TITAN X) durchgeführt. Diese Karte hat 3072 Kerne auf denen simultan gerechnet werden kann. Die grobe Simulation benötigte daher nur ca. 3 Stunden. Die feine Rechnung benötigte hingegen ca. einen Tag ($26h$), da sie wesentlich mehr Knoten umfasst. Zum Vergleich mit Giomettos Simulation bietet sich die uniforme Rechnung an. Diese benötigte wegen der fehlenden Vergrößerungsstufen lediglich 9 Stunden. Sie ist somit bei der gleichen räumlichen Auflösung signifikant schneller, als Giomettos Pseudo-Spektraler Code.

5 Zusammenfassung

In dieser Arbeit wurde gezeigt, dass die CumLBM in der Lage ist Windströmungen genau und schnell zu simulieren. Die Ergebnisse sind im Vergleich zu der Referenzsimulation in einigen Bereichen näher an den Messungen. Vor allem die turbulente kinetische Energie wird besser dargestellt. Dies ist bemerkenswert, da im Gegensatz zur Referenzsimulation keine komplexen expliziten Turbulenz- und Wandmodelle verwendet wurden. Dies macht die CumLBM für Nutzer leichter zugänglich. In Bezug auf die fehlenden Turbulenz- und Wandmodelle ist aber anzumerken, dass nur normierte Profile untersucht wurden. Welchen Einfluss diese Modelle auf die absoluten Werte für Windgeschwindigkeit und Turbulenz haben ist unklar und sollte in Zukunft mit entsprechender Modellierung untersucht werden.

Dank effizienter GPU-Implementierung können die Ergebnisse in kurzer Zeit berechnet werden. Der Wind über einem Stadtviertel wurde in einem halben Tag berechnet, wohingegen der bereits sehr effiziente Pseudo-Spectral Code von Giometto fünf Tage für die vergleichbare Simulation benötigte. Alle CumLBM Simulationen wurden darüber hinaus auf Desktop Computern durchgeführt. Durch die effiziente Nutzung von Grafikkarten sind keine teuren Computercluster notwendig. Dadurch wird diese Methode auch für kleine und mittlere Unternehmen und Behörden zugänglich.

Zusammenfassend zeigt die CumLBM viele Eigenschaften die für ein Werkzeug zur Simulation von Windströmungen im urbanen Raum wünschenswert sind.

Literatur

- Bhatnagar, P. L., Gross, E. P. und Krook, M. (1954). „A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems“. In: *Physical review* 94.3, S. 511.
- Blocken, B. und Carmeliet, J. (2004). „Pedestrian wind environment around buildings: Literature review and practical examples“. In: *Journal of Thermal Envelope and Building Science* 28.2, S. 107–159. ISSN: 1744-2591. DOI: 10.1177/1097196304044396.
- Fröhlich, J. (2006). *Large Eddy Simulation turbulenter Strömungen*. 1. Aufl. Teubner. ISBN: 978-3-8351-0104-3.
- Geier, M., Schönherr, M., Pasquali, A. und Krafczyk, M. (2015). „The cumulant lattice Boltzmann equation in three dimensions: Theory and validation“. In: *Computers & Mathematics with Applications* 70.4, S. 507–547. ISSN: 08981221. DOI: 10.1016/j.camwa.2015.05.001.
- Giometto, M., Christen, A., Meneveau, C., Fang, J., Krafczyk, M. und Parlange, M. (2016). „Spatial characteristics of roughness sublayer mean flow and turbulence over a realistic urban surface“. In: *Boundary-Layer Meteorology*, S. 1–28. ISSN: 1573-1472. DOI: 10.1007/s10546-016-0157-6.
- Kukačka, L., Fuka, V., Nosek, Š., Kellnerová, R., Jaňour, Z., Vít, T., Dančová, P. und Novotný, P. (2014). „Ventilation of idealised urban area, LES and wind tunnel experiment“. In: *EPJ Web of Conferences*, 67:02062. DOI: 10.1051/epjconf/20146702062.
- Michioka, T., Sato, A., Takimoto, H. und Kanda, M. (2011). „Large-Eddy Simulation for the Mechanism of Pollutant Removal from a Two-Dimensional Street Canyon“. In: *Boundary-Layer Meteorology* 138.2, S. 195–213. ISSN: 0006-8314. DOI: 10.1007/s10546-010-9556-2.
- Nations, U. (2014). *World urbanization prospects: The 2014 revision: highlights*. New York. ISBN: 978-92-1-151517-6.
- Nozu, T., Tamura, T., Okuda, Y. und Sanada, S. (2008). „LES of the flow and building wall pressures in the center of Tokyo“. In: *Journal of Wind Engineering and Industrial Aerodynamics* 96.10, S. 1762–1773. ISSN: 01676105. DOI: 10.1016/j.jweia.2008.02.028.
- NVIDIA (2015). *Programming Guide :: CUDA Toolkit Documentation*: <http://docs.nvidia.com/cuda/cuda-c-programming-guide/>.
- Pasquali, A., Schönherr, M., Geier, M. und Krafczyk, M. (2013). „LBMHexMesh: an OpenFOAM based grid generator for the Lattice Boltzmann Method (LBM)“. In: *7th Open Source CFD International Conference*.
- Rotach, M., Vogt, R., Bernhofer, C., Batchvarova, E., Christen, A., Clappier, A., Feddersen, B., Gryning, S.-E., Martucci, G., Mayer, H., Mitev, V., Oke, T. R., Parlow, E., Richner, H., Roth, M., Roulet, Y.-A., Ruffieux, D., Salmond, J. A., Schatzmann, M. und Voogt, J. A. (2005). „BUBBLE—an Urban Boundary Layer Meteorology Project“. In: *Theoretical and Applied Climatology* 81.3-4, S. 231–261. ISSN: 0177-798X. DOI: 10.1007/s00704-004-0117-9.
- Schönherr, M. (2015). „Towards reliable LES-CFD computations based on advanced LBM models utilizing (Multi-) GPGPU hardware“. Dissertation. Braunschweig: TU Braunschweig.
- Tominaga, Y., Mochida, A., Yoshie, R., Kataoka, H., Nozu, T., Yoshikawa, M. und Shirasawa, T. (2008). „AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings“. In: *Journal of wind engineering and industrial aerodynamics* 96.10, S. 1749–1761. ISSN: 01676105. DOI: 10.1016/j.jweia.2008.02.058.
- Xie, Z.-T. und Castro, I. P. (2009). „Large-eddy simulation for flow and dispersion in urban streets“. In: *Atmospheric Environment* 43.13, S. 2174–2185. ISSN: 13522310. DOI: 10.1016/j.atmosenv.2009.01.016.

Xie, Z. und Castro, I. P. (2006). „LES and RANS for turbulent flow over arrays of wall-mounted obstacles“. In: *Flow, Turbulence and Combustion* 76.3, S. 291–312. ISSN: 1386-6184. DOI: 10.1007/s10494-006-9018-6.

Applying Visual Programming Methods for Parametric Bridge Modeling and Generation of Model Variations

Ali Ismail^{1a}, Yaseen Srewil^{2a} and Saiful Abedin^{3b}

^a*Institut für Bauinformatik, Technische Universität Dresden Nürnbergerstr. 31a, 01062 Dresden*

¹*E-Mail: ali.ismail@tu-dresden.de, ²E-Mail: yaseen.srewil@tu-dresden.de*

^b*hkc Architekten-Ingenieur Consult GmbH, Brückenweg 5 18146 Rostock*

³*E-Mail: saifulabedin@hkc-gmbh.com*

Abstract The design of infrastructure projects like complex bridges is challenging task for planners and engineers. The traditional design and modeling process for bridge models needs a large number of repetitive design processes and a lot of manual work for each model variation or design modification. The integration of visual programming language (VPL) methods and Building Information Modeling (BIM) software enables automated repetitive design process and simplify the generation of model variations especially in the primary design phase, for example to check the design effectiveness for wind analysis. This paper presents in details a workflow for geometry parameterized bridges modeling with help of VPL tools and study cases to generate bridge model variations. It gives also a brief overview about the semantic enrichment of generated models based on the IFC standard.

Keywords: Visual programming VP, Parametric bridge modeling, BIM Annotation, Semantic enrichment, Dynamo

Introduction

In today's infrastructure construction field, the design process includes a large number of repetitive processes during the complex design procedure. The application of visual programming methods can be considered as a great potential for large number of similar task in bridge design process and parametric model variations of the bridge, to optimize the bridge alignment and orientation to fit in the surrounding environment. Here, the term of "parametric" refers to the relationships among all elements of the model that enable coordination, variation and change management. While semantic data refers to the classification of model elements, assigning attributes and defining relationships between model elements. The visual design methods enables users to design a process rather just objects, it allows the designer to find new solutions and step beyond the limitations of traditional CAD/BIM software and 3D modelers. In fact, the distinction between parametric and non-parametric models is basically in the geometry and rules. Parametric models have explicit rules and implicit geometry, and in reverse the non-parametric models. Changing a rule or constraint, or modifying a part of the model itself, has implications

for the entire model. The rules, relationships and constraints among model's parts determine the geometry (Mackenzie, 2013).

This research aims to deal with challenges for bridge design in the early phase of design with focus on the work flow to optimize the bridge performance due to wind loads. It proposes a workflow and parametric visual templates for typical bridge types that is more intuitive and efficient for generation of model variation. Next, the semantic quality of 3D generated models can be improved and linked with specific domain information from various domains with focus on bridge models based on the IFC standards and the under development extension IFC-BRIDGE. The main target of this research is to optimize the bridge design in the primary design phase for a better wind analysis performance. The prototype implementation is mainly based on the open source code tool "Dynamo" and the BIM authoring software "Autodesk Revit".

Related work

Although, visual programming (VP) paradigm is relatively new in AEC industry, it is a very promising tool to solve variation problems in this domain (Tedeschi, 2014). This paradigm is used successfully for simplification of building models query like the BIMcarft filter tool for personal at construction site (Wülfing et al., 2014). Also Ritter et al. (2015) investigated the fundamental of visual programming and evaluated its possible application in AEC industry. Several CAD and 3D software vendors have developed visual tools which are based on node diagrams to make scripting more accessible to user with limited programming skills.

For bridge modeling, VP design tools have been represented and demonstrated in several articles and tutorials videos. A feasible study is proposed by Langwich (2016) demonstrates an example for bridge modeling according to German bridge standards and requirements with Revit and the visual design tool Dynamo. Singer (2015) presented the application of knowledge-based engineering and VP methods in methods in early stages of bridge design. Other applications demonstrate how to extract bridge alignment data to Revit from Autodesk Civil3D and use Dynamo for bridge detailed modeling and comparing alternative of bridge model (Stark, 2015; Younghwi, 2016).

Modeling a detailed bridge model using visual programming

In this section we present a workflow to generate bridge design model variations using visual programming methods with help of visual programming tool Dynamo. The most important advantage of this method is the automation of modeling process, the possibility to replace model elements with different level of details and change the geometry parameters in consistent way, using advanced modeling concepts like form-finding algorithms and NURBS curves for complex geometry which satisfy a set of geometric and logical constraints. It saves a lot of manual modeling time and automates as well the process of linking external data source (AutoCAD civil 3D, excel etc.) for creating variable profiles and the bridge alignments.

Figure 1 depicts the main steps for creating a parametric and bridge model variations and equipped this models with necessary semantic data based on IFC standard. The result is a BIM enrichment semantic model which can be driven into diverse applications like structural analysis or CFD wind analysis.

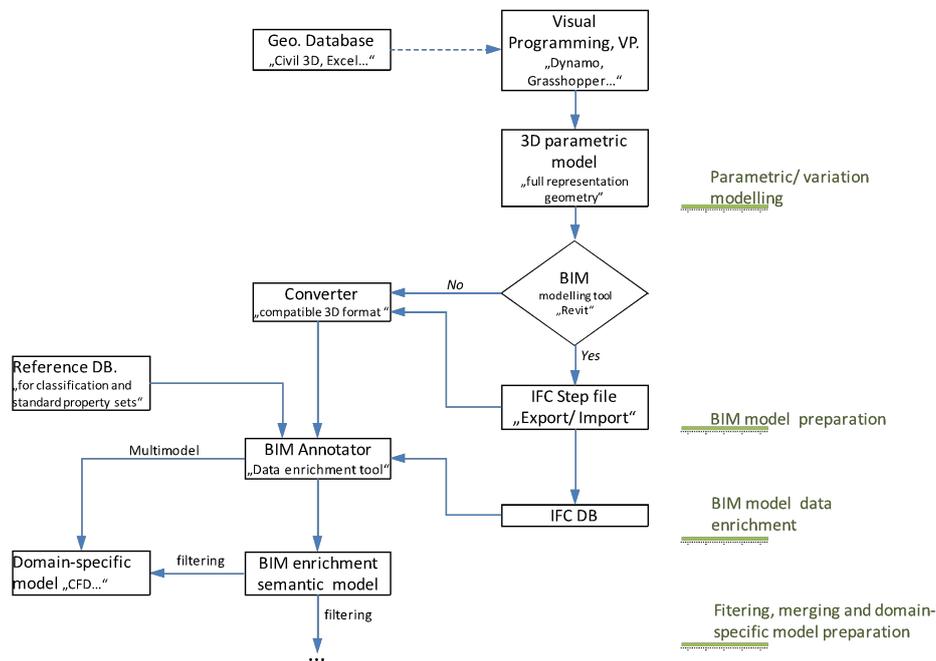


Fig. 1: A generic workflow for modeling a parametric and semantic enrichment of bridge models

Study case 1: Bridge modeling using Dynamo and Autodesk Revit

Dynamo is an open source tool, which can be used standalone or as extension inside Autodesk Revit. It enables designers to apply computational design and automation processes through a node-based visual programming interface without the need to write a lot of code. It gives the designer the added advantage of being able to leverage computational design workflows within the context of a BIM environment. Users can create node-based design definitions for creating, positioning, and visualizing geometry. The visual programming framework lets the user create unique systems and relationships and expand how BIM can be used to drive design ideation (www.dynamobim.org).

While designing a bridge, some issues must be considered. Nowadays the key challenges bridge design in Germany are not the economic aspects but due to alignment, the complex geometry at curved alignment, at longitudinal inclination, skewed substructures, variable bottom edges etc. (Stark, 2015).

In this study case, Dynamo is used only to assist in the construction of bridge route. Easy to understand workflow of visual programming and step results at the same time motivates to know how the visual programming can be integrated with Revit to smooth and automate workflows of bridge design. This approach is carried out based on the adaptive families to create the bridge model in Revit. Bridge cross-section can be correctly placed along the route and sweep volume elements can be created using “Create Form” Revit functionality. The use of adaptive families carries certain advantages, such as:

- They can be oriented to vertical direction at the placement.
- Start and end can be precisely defined.

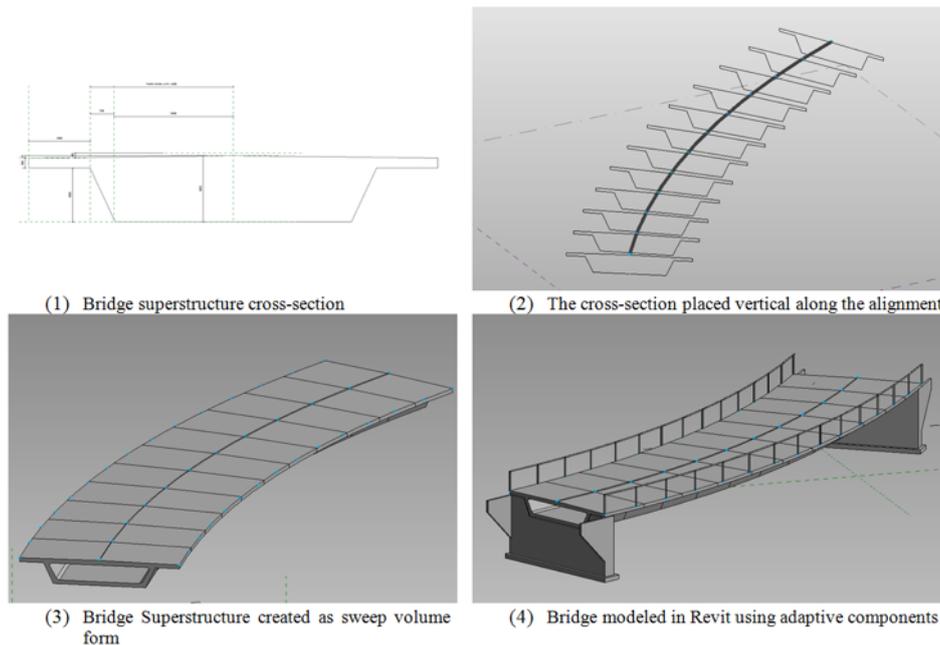


Fig. 3: Study-case modeling phases

Study case 2: Bridge modeling using Dynamo as standalone tool

The second study case presents a fully Visual Programming Language (VPL) approach for the geometric modeling of arch bridge. VPL allows creating geometry relationships based on rules and logic. The bridge geometries slab, railing, pier modelled as surface with certain thickness with the help of nodes in Dynamo visual programming. The geometry parameters can be modified accordingly by the adjustment of the input parameter values.

The modeling process of arch bridge is described below in sub-process and the corresponding implementation is shown.

1. Specifying the start and end station of the bridge: The basic geometric modeling begins with the definition of the bridge arch length and radius. The road length is considered as the tangent to the vertex of the arch (Figure 4). Three independent points representing start, vertex of arch and end of the bridge are defined. These points are manipulated later to succeeding parts to complete the modeling.
2. Creating arch column support: The vertical arch column has been created at equal distance on the arch. Each column is generated as a function, joining line between the curve points on arch and intersecting points on tangent road length. The column form is modeled by translating the column line in parallel direction and lofting the cross-section between lines to certain thickness. Dynamo nodes “Line.ByStartPointEndPoint”, “Geometry.Translate” and “Surface.ByLoft” used in this operation. The input parameter for the number of columns can be explicitly set by “Integer Slider” node.

Applying Visual Programming Methods for Parametric Bridge Modeling and Generation of Model Variations

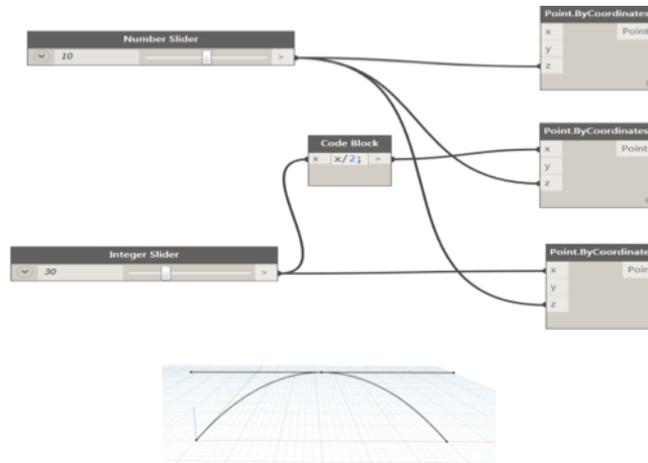


Fig. 4: Start and end station of the bridge

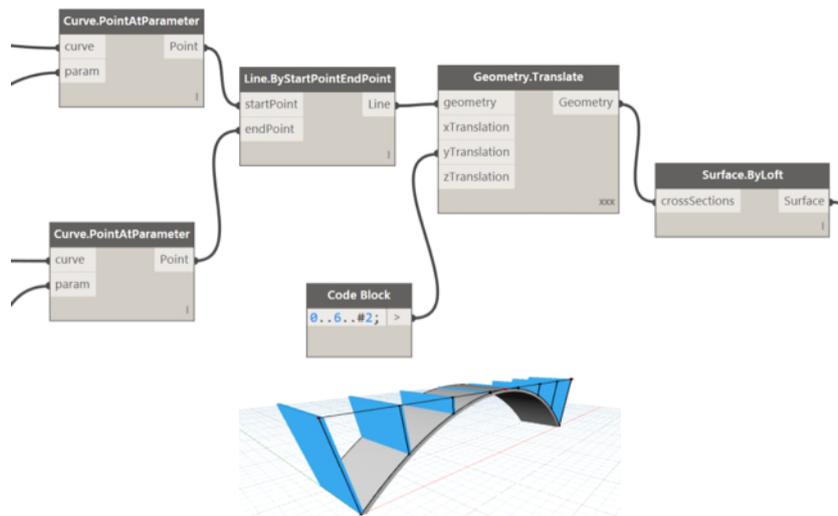


Fig. 5: Arch column creation

3. Creating road surface: The road surface is created as same as arch column geometry by translating the tangent of the arch in the direction of width and lofting inside of the surface. The surface thickness parameter can be controlled by “Number Slider” node.
4. Creating railing: To complete the geometry of the bridge, we need to create railing on both side of the roadway. To limit the railing start and end point from the bridge length, the tangent curve has been trim by “Curve.TrimByParameter”. Translating the trim curve in vertical direction, manipulating both curves into certain segments and after that intersection points are joined through the node “Line.ByStartPointandEndPoint”. A series of planes created across the line curve by “Curve.PlaneatParameter” to adjust the geometry according to the curve profile. Generating series of circle on all planes the curve path sweep as solid circular shape. Finally, using the node “Geometry.Translate” the created geometry is mirrored on the other side of the road.

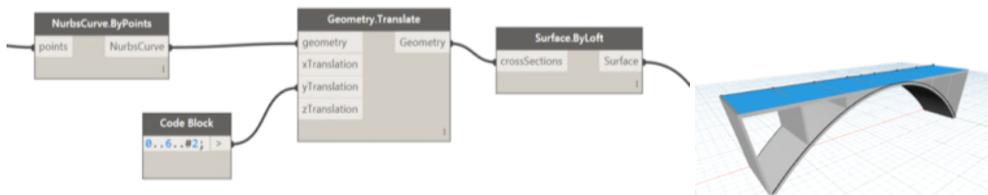


Fig. 6: Road surface creation

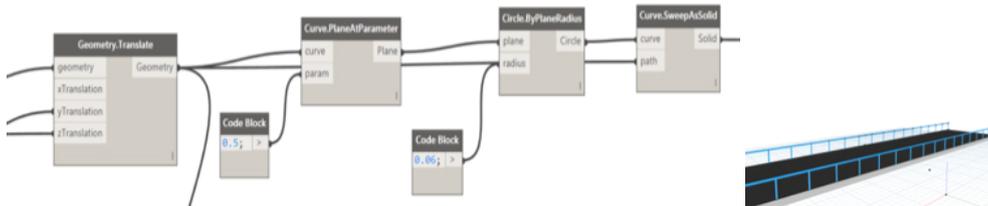


Fig. 7: Railing creation

Figure 8 shows the arch bridge model generated in Dynamo. Bridge modeled in Dynamo can be imported in Revit as generic model.

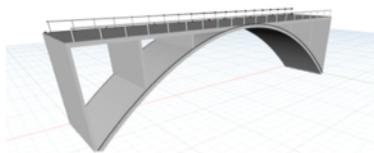


Fig. 8: Arch bridge modeled in Dynamo

Bridge modeled in Dynamo can be imported in Revit as generic model. Dynamo Plug-in for Robot Structural Analysis/React Structures helps to create parametric and complex structural frames models in Dynamo, then bring it to Autodesk Robot Structural Analysis or Autodesk React Structures for simulation, and review the analysis results.

The geometry modeled in dynamo can be transformed into mesh geometry using Dynamo package “Mesh Toolkit”. Meshes are most flexible form of geometry in the computational

design. The Mesh geometry can be exported direct as COLLADA format. Which can optimize the process of the geometric model first pushed into Revit and converting into IFC. After that an IfcOpenShell converter tool needs to make this IFC model as COLLADA format. To exchange standard information through multimodal method, COLLADA format is needed to link the data model for semantic enrichment. Therefore study and judgment should be carried out to see the feasibility of the process directly exporting the model from Dynamo to other 3D formats.

1 Study case 3: Work flow of a long span bridge model with Dynamo

In this modeling approach we created a Dynamo script that enables designer to generate a long span bridge model from Revit family directly in the project environment following the bridge alignment. As excel can be considered as management of building information, in this area Dynamo equipped to parametric data manipulation of the model. To create the bridge model first Revit families for superstructure, substructure, abutment etc. loaded in the Revit project template. The parametric variation of geometry can be easily created and changed through the Dynamo script.

1. Creation of bridge Superstructure

The modeling process begins with the bridge alignment creation from excel data point co-ordinates. The workflow for creating the alignment is same as study case 1. Dynamo can create alignment importing X, Y, Z co-ordinates directly from excel.

The superstructure form of the bridge can be created along the alignment curve using Dynamo nodes “StructuralFraming.BeamByCurve” and with the help of superstructure family. This node will search for loaded Revit family in the project.

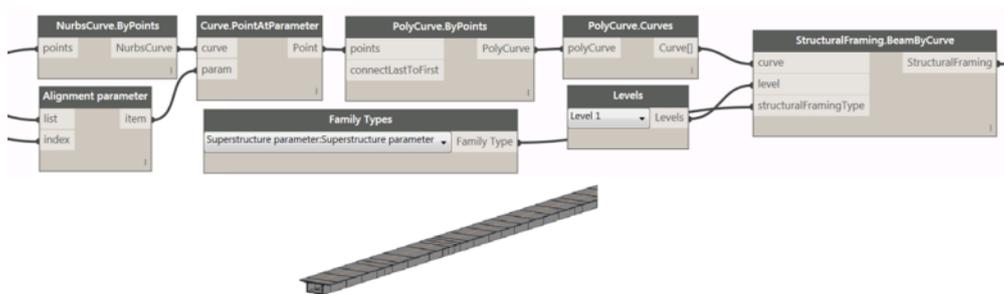


Fig. 9: Creation of bridge superstructure

2. Parametric variation of superstructure

Based on the parameters defined on the Revit family, the parameters can be controlled through Dynamo node “Element.SetParameterByname”. The variable parameters for the cross-section such as width, thickness, height of the superstructure data can be assigned from excel through this node. Running the Dynamo script can generate variable section of superstructure. Figure 10 shows how excel and Dynamo can be equipped to create free form geometry in Revit.

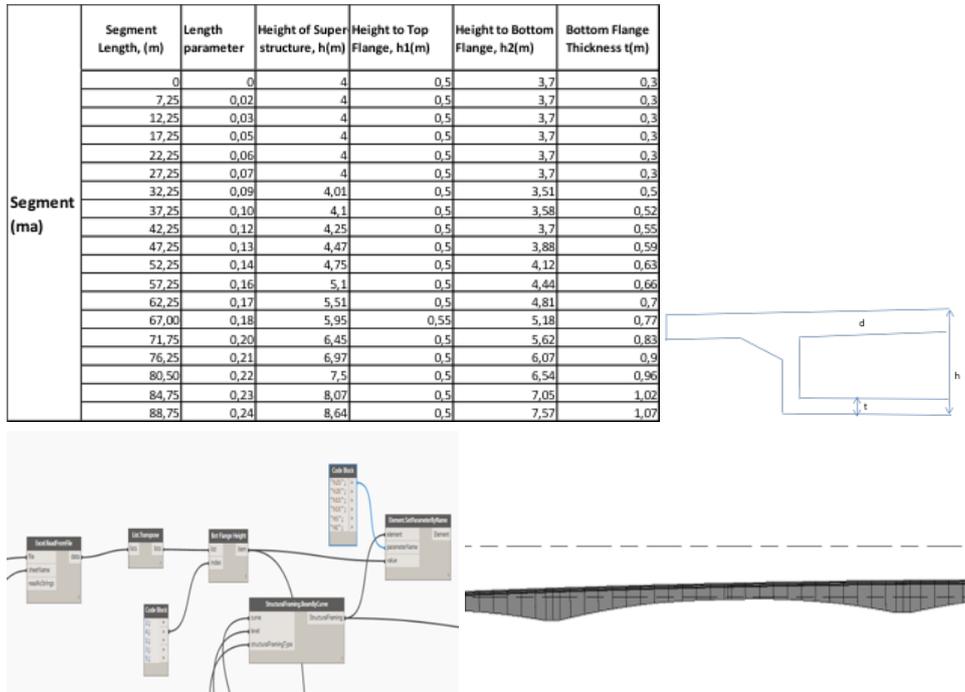


Fig. 10: Extract Superstructure parameters from Excel file

3. Pier creation

As the bridge has different span length, first pier locations on the alignment are determined. Using Dynamo node “Point.ByCartesianCoordinates” and connect it to “Line.ByStartPoint DirectionalLength” node two vertical lines created. These 2 lines represent as the curve path for pier and similarly as the superstructure, piers have been created using Dynamo node “StructuralFraming.BeamByCurve”.

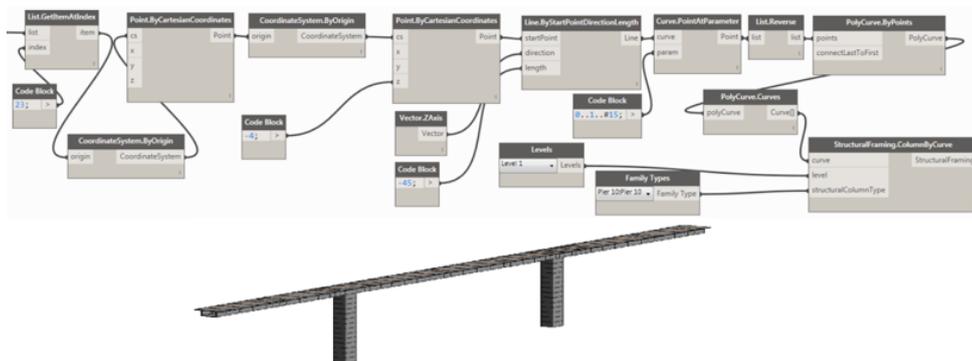


Fig. 11: Pier creation

4. Placing the abutment

At the beginning and end of the bridge alignment abutment are placed. The points for placing the abutment family are determined from the start and end points of the curve. After loading the abutment family can be placed at the points with the aid of the node “FamilyInstance.ByPoint”. The abutment parameters width, wall thickness, height can be controlled through Dynamo node. The width of the abutment is chosen from the superstructure width to act parametric together with superstructure. Orientation of the abutment can be directed to the bridge alignment by Dynamo node “FamilyInstance.Set Rotation”.

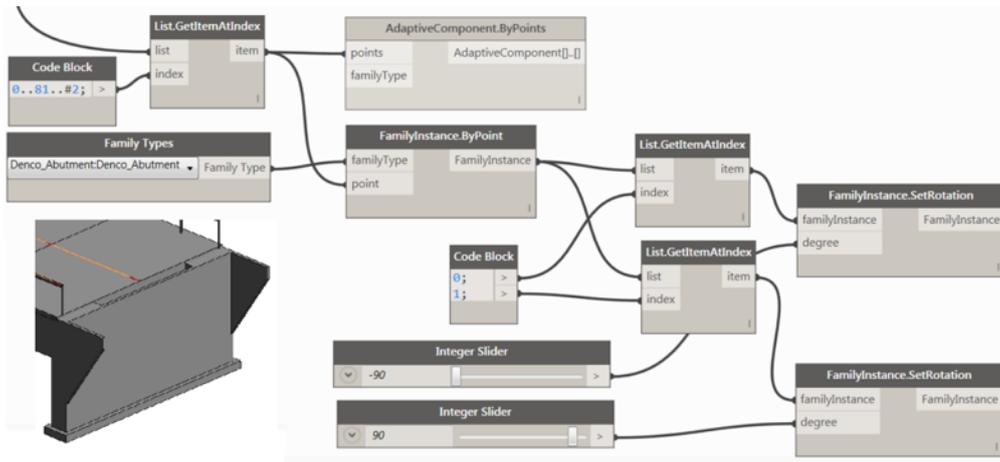


Fig. 12: Loading and placing the abutment

Semantic data enrichment of 3D models

So far the 3D geometry model of the bridge using VP with Dynamo is generated. The resulting model has a comprehensive geometric representation. However, it has poor semantic information (classification of elements, attributes). To handle this issue and get the ultimate potential of using the BIM models for different data exchange scenarios during the construction life cycle, an annotation tool “BIM-Annotator” with the help of a reference database for classification and standard attributes of typical bridge elements was developed (Ismail et al., 2016). Here, BIM-Annotator is used for:

- Collaborative online data enrichment tool by adding, editing and extending information to original model.
- A consistent annotation process by using a reference database (accessible and searchable) for classification and standard attributes of typical bridge element.

In a rich semantic model each element in the model should donate a very specific component. This means, it should be assigned to a specific class and described with a set of predefined properties and relations with other elements. The result will be a virtual 3D bridge model which consists of components such as piers, beams and abutments and hence partial views and specific information can be extracted through filtering and semantic reasoning.

For this purpose, all IFC classes including extra classes of the IFC-Bridge extension and standard property sets of IFC model have been integrated into a reference database. BIM-Annotator allows the user to assign classes of elements, edit, delete and add new property sets, define and organize group of objects hierarchically and assign attributes to them. The semantic data which are entered through the BIM-Annotator can be exported in XML format with an additional XML-Linkmodel, which links the semantic information with 3D objects (Figure 13).

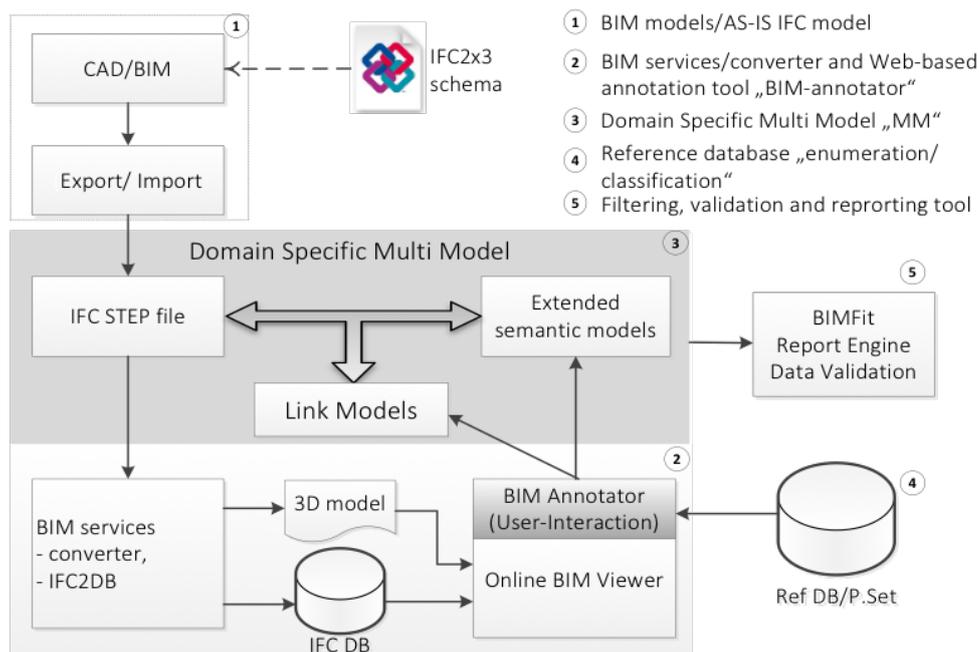


Fig. 13: BIM-Annotator data flow and system architecture

Figure 14 shows the process of assigning new class and use of the new added property set template of a bridge segment part element. (<http://150.254.208.60:3000/projects/1/documents/7>)

Conclusion and future work

In this paper we presented a workflow and study cases to generate bridge model variations based on the visual programming approach and the semantic enrichment of these models based on the IFC standard. Bridges are modeled using Dynamo standalone and as extension inside Revit. First, Dynamo plugin for Revit has been used to create the alignment and the bridge is fully modeled in Revit from adaptive families. Limitation of Revit with the Civil Structures plugin are bridge definition, substructure (supports, abutments etc.) from the templates are freely modifiable, but not all bridge types can be modeled. Here, the users have less choice to create complex shape of bridge geometry through this plug in.

The second study case shows the possibilities for the designer to model the bridge from scratch fully standalone in Dynamo visual programming workspace. The parameters such as

Applying Visual Programming Methods for Parametric Bridge Modeling and Generation of Model Variations



Fig. 14: Change the class and add new properties of a bridge segment part element

adjustment of dimensions, insertion or movement of components can be managed automatically through nodes in Dynamo. Finally the bridge model can be driven into the BIM Annotator tool to improve the semantic quality of bridge models and link specific domain information from various domains with focus on bridge models based on the IFC standards.

Acknowledgements

The work presented in this paper was enabled by the financial support of the European Commission in the Eurostars research projects SE-LAB (E!7521) and BridgeCloud (E!7987). This support is herewith gratefully acknowledged.

References

- Ismail, A., Srewil, Y., Scherer, R., and Mansperger, T. (2016). „Semantic Enrichment and Multimodel Data Exchange Approach for CFD Analysis of Bridges“. In: *23rd International Workshop: European Group for Intelligent Computing in Engineering (EG-ICE)*. (Krakow, Poland, June 29–July 1, 2016).
- Langwisch, O. (2016). „Bridge modeling“. In: *Revit Structure Magazine*. http://www.autocad-magazin.de/sites/default/files/magazine-pdf/acm_2016-01_archiv_reduziert.pdf.
- Mackenzie, D. (2013). „Parametric Infrastructure Design“. In: *Lightweight Structures Association of Australia (LSAA) Conference*. (Auckland, Nov. 13–15, 2013).
- Ritter, F., Preidel, C., Singer, D., and Kaufmann, S. (2015). „Visuelle Programmiersprachen im Bauwesen - Stand der Technik und aktuelle Entwicklungen“. In: *Proceedings of the 27th Forum Bauinformatik*. (Aachen, Germany).
- Singer, D. (2015). „Einsatz wissensbasierter Methoden in frühen Phasen des Brückenentwurfs“. In: *Proceedings of the 27th Forum Bauinformatik*. (Aachen, Germany).
- Stark, M. (2015). *AutoDesk University: REVitalize Bridge Design*. <http://au.autodesk.com/au-online/classes-on-demand/class-catalog/2015/revit-for-construction/ci11198>.
- Tedeschi, A. (2014). *AAD Algorithms-aided design: Parametric strategies using Grasshopper*. Le Penseur, p. 27.

- Wülfing, A., Windisch, R., and Scherer, R. (2014). „A visual BIM query language“. In: *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2014*. Ed. by A. Mahdavi, B. Martens, and R. J. Scherer, pp. 157–164.
- Younghwi, K. (2016). *Bridge Modeling using Dynamo*. video tutorial. <https://www.youtube.com/watch?v=LgaySpdjljA>.

Automatisierte Dimensionierung von Baustelleneinrichtungsgegenständen

Katrin Jahr

*Lehrstuhl für Computergestützte Modellierung und Simulation,
Technische Universität München, Arcisstr. 21, 80333 München, Deutschland.
E-Mail: katrin.jahr@tum.de*

Abstract Eine angemessene und zweckmäßige Baustelleneinrichtungsplanung unterstützt die fristgemäße, kosteneffiziente und qualitative Umsetzung eines Bauvorhabens. Die Nutzung von wissenschaftlich fundierten Planungswerkzeugen kann sowohl den Vorgang der Baustelleneinrichtungsplanung beschleunigen, als auch zu besseren Ergebnissen führen. Zur automatisierten Erstellung eines Baustelleneinrichtungsplanes sind drei Schritte notwendig: Die Identifizierung, die Dimensionierung und die Positionierung der Baustelleneinrichtungsgegenstände. Dieser Beitrag beschäftigt sich mit dem zweiten Schritt – der Dimensionierung. Dazu wird das Vorgehen bei der Automatisierung der Dimensionierung präsentiert und klare Dimensionierungsregeln entwickelt.

Keywords: Baustelleneinrichtung, Automatisierte Baustelleneinrichtungsplanung, Dimensionierung

Einleitung

Die Baustelleneinrichtung (BE) umfasst alle produktiven und unproduktiven Anlagen, die vor Ort zur Errichtung oder Sanierung eines Bauwerks benötigt werden (Meyran, 1973). Abhängig von Standort, Gegebenheiten der Baustelle und genutzten Bauverfahren ergeben sich verschiedene Anforderungen an die Baustelleneinrichtung. Diese Anforderungen werden in die Entscheidungen der frühen Planungsphasen wenig einbezogen - die Baustelleneinrichtungsplanung wird erst im Rahmen der Ausführungsplanung durchgeführt. Abweichungen der verschiedenen Ausführungen und damit einhergehende stark abweichenden Kosten werden bei der Entscheidungsfindung wenig beachtet. Zudem ist die Baustelleneinrichtungsplanung trotz des großen Einflusses auf die Baustellengemeinkosten und die Produktivität des Bauvorhabens bisher wenig formalisiert. Die Planung der einzelnen Elemente erfolgt zumeist händisch anhand von Faustformeln. Wissenschaftlich fundierte Unterstützungswerkzeuge sowie qualitative oder quantitative Bewertungen der Ergebnisse werden nicht genutzt. Der Erfolg der Baustelleneinrichtungsplanung ist somit allein vom Fachverstand und der praktischen Erfahrung des Planers abhängig. Durch die Nutzung wissenschaftlich fundierter Unterstützungswerkzeuge könnte die Baustelleneinrichtung effizient und unter Berücksichtigung einer Vielzahl von Faktoren geplant und die Bauausführung so erheblich unterstützt werden. In der Vor-, Entwurfs- und Genehmigungsplanung wurde mit der Einführung des Building Information Modeling

(BIM) die Planungseffizienz und Planungsqualität, und damit die Erfolgsaussichten eines Bauprojekts, maßgeblich erhöht (Borrmann et al., 2015). Die Vorzüge semantischer Bauwerkselemente und verlustfreier Datenübertragung können auch in der Baustelleneinrichtungsplanung genutzt werden, indem die Planung der Baustelleneinrichtung unterstützt oder sogar vollständig automatisiert wird. So können bereits in frühen Planungsphasen die Auswirkungen verschiedener Entscheidungen auf die Baustelleneinrichtung evaluiert werden und aktiv in den Entscheidungsprozess einfließen. Diese Möglichkeit wird im Rahmen des Projektes BIMsite untersucht (Krepp et al., 2016) Aus einem BIM-Modell werden verschiedene Ausführungsvarianten generiert. Die resultierenden Informationen über verwendete Verfahren und benötigte Materialien werden genutzt, um die Baustelleneinrichtung zu planen. Im Anschluss werden die verschiedenen Ansätze mit Hilfe einer Bauablaufsimulation evaluiert. Zur teil- oder vollständig automatisierten Planung der Baustelleneinrichtung wurden drei Teilschritte identifiziert: Die Identifikation, die Dimensionierung und schließlich die Positionierung der benötigten Baustelleneinrichtungsgegenstände.

Dieser Beitrag beschäftigt sich mit der Dimensionierung der Baustelleneinrichtungsgegenstände. Im ersten Abschnitt wird eine Einführung zur Baustelleneinrichtungsplanung und den benötigten Elementen gegeben. Im zweiten Abschnitt wird die Vorgehensweise bei der automatisierten Dimensionierung von Baustelleneinrichtungsgegenständen erläutert. Anhand aktueller Verordnungen, Regeln der Technik und logischer Schlussfolgerung werden im dritten Teil Formeln und Regeln zur Dimensionierung dieser Elemente entworfen, die im Rahmen der automatisierten Baustelleneinrichtungsplanung genutzt werden können. Im letzten Teil dieser Arbeit werden die gewonnenen Erkenntnisse zusammengefasst und die zukünftige Arbeitsrichtung aufgezeigt.

Baustelleneinrichtungsplanung

Im Rahmen der Baustelleneinrichtungsplanung werden alle Elemente und Einrichtungen, die zur Errichtung oder Sanierung eines Gebäudes benötigt werden, identifiziert, dimensioniert und auf der verfügbaren Fläche platziert. Ziel der Baustelleneinrichtung ist dabei die bestmögliche Vorbereitung aller Bauprozesse, um einen flüssigen und kontinuierlichen Bauablauf zu ermöglichen. Die Elemente der Baustelleneinrichtung werden in sieben Gruppen eingeteilt (Schneider, 2016):

1. Gerätschaften
2. Sozial- und Büroeinrichtungen
3. Lagerflächen
4. Verkehrsflächen und Transportwege
5. Medienversorgung und Entsorgung
6. Baustellensicherung
7. Sicherung der Baugrube

Besondere Relevanz während der Planung kommt Kranen und Betonpumpen (Gruppe 1), Büro-, Aufenthalts- und Sanitätscontainern (Gruppe 2), innen und außen liegenden Lagerflächen (Gruppe 3) sowie Transportwegen auf der Baustelle (Gruppe 4) zu. Diese Elemente sind zum einen sehr variabel je nach Anforderungen und Gegebenheiten der Baustelle, zum anderen haben sie einen großen Einfluss auf die Gemeinkosten der Baustelle und einen flüssigen Bauablauf.

Hauptaugenmerk der Baustelleneinrichtungsplanung liegt auf kurzen Transportwegen, kostengünstigen und sicheren Arbeitsbedingungen aller an der Bauausführung Beteiligten. Durch die verwendeten Bauweisen, das vorliegende Grundstück, die angrenzenden Liegenschaften und lokale Besonderheiten können verschiedene Randbedingungen und Anforderungen an

die Baustelleneinrichtung entstehen. Diese Randbedingungen bewirken Abweichungen in den benötigten Materialien, Lagerräumen, Baumaschinen und Prozessmustern, so dass gute Lösungen für bestimmte Projekte nicht auf andere Projekte übertragbar sind. Um sinnvolle Maße und Kennzahlen zu ermitteln, muss für jedes Bauprojekt eine erneute Planung der Baustelleneinrichtung erfolgen. Auch bei Änderungen des Bauvorhabens oder des Bauablaufs müssen Berechnungen und Planungen wiederholt werden, was einen zusätzlichen Aufwand erfordert.

Im Rahmen der Baustelleneinrichtungsplanung werden die benötigten BE Elemente identifiziert, dimensioniert und platziert. Die Identifikation der Elemente erfolgt anhand von Abhängigkeiten, welche sich durch die verwendeten Bauweisen ergeben. Die Dimensionierung erfolgt anhand normativer Regelungen, dem Platzbedarf der einzelnen Elemente und Überlegungen aus dem Materialfluss. Eine genauere Beschreibung findet sich im folgenden Abschnitt. Zur Platzierung der Baustelleneinrichtungs-elemente werden bestimmte Bereiche der Baustelle für bestimmte Elemente eingeschränkt oder vollständig gesperrt (Schwabe, 2015). So sollten beispielsweise die Krane einen ausreichenden Sicherheitsabstand zur Baugrube aufweisen und entsprechend der zu hebenden Güter und der Traglastkurven ausreichend dimensioniert sein (Schach und Otto, 2011). Zur Platzierung der Elemente und Erzeugung eines möglichst optimalen Baustellenlayouts kann ein Algorithmus genutzt werden. Dazu werden alle BE-Elemente entsprechend der Platzierungsregeln auf der Baustelle platziert, wobei Verschiebungen und Verdrehungen erlaubt sind. Zielgrößen dieser multikriteriellen Optimierung sind möglichst geringe Kosten und eine möglichst hohe Sicherheit. Positiven Einfluss auf Kosten und Sicherheit haben zum Beispiel kurze Wege zwischen den BE-Elementen ohne die Durchquerung gesundheitsgefährdender Bereiche und eine geringe Anzahl an Umsetzungsvorgängen.

Verschiedene Forschergruppen haben sich mit der Baustelleneinrichtungsplanung befasst. Die Forschung umfasst dabei vorrangig die Optimierung der Platzierung vorgefertigter Baustelleneinrichtungsgegenstände. Erste Entwicklungen eines automatisierten Programmes zur Erstellung eines Baustelleneinrichtungsplanes wurden veröffentlicht von Tommelein et al. (1992). Dabei wurde ein Model entwickelt, das mit Hilfe künstlicher Intelligenz die Handlungen eines Planers während der Ausführungsplanung abbildet. Im ersten Schritt wird der zu betrachtende Bereich und dessen Grenzen festgelegt. Danach werden dauerhafte Objekte und schließlich temporäre Objekte erzeugt und positioniert. Bauvorhaben mit einer großen Anzahl von Randbedingungen, beispielsweise größere Bauvorhaben im innerstädtischen Bereich, können nicht abgebildet werden. Huang und Wong (2015) nutzen einen binary-mixed-integer-linear-Algorithmus, den sie mit Hilfe eines Branch-and-Bound-Algorithmus aus einer kommerziellen Software lösen. Durch die Vorgabe von elf diskreten erlaubten Bereichen unterschiedlicher Größe wird der Suchraum hier stark vereinfacht. Ein weiterer Ansatz von Andayesh und Sadehghpour, 2013 nutzt die Theorie der potentiellen Energie. Hier werden die Elemente durch ihren Umkreis repräsentiert, die sich in einem kontinuierlichen Baufeld bewegen. Elemente, die nicht zeitgleich benötigt werden, können den gleichen Platz einnehmen. Es existieren somit keine fest definierten Bauphasen, Umbauten können jederzeit notwendig werden. Shawki et al. (2010) und Kumar und Cheng (2015) schlagen die Nutzung genetischer Algorithmen zur Erstellung von Baustelleneinrichtungsplänen vor. Dabei beginnen die Algorithmen mit zufälligen Layouts, die durch Kreuzung und Mutation variiert werden. Kumar und Cheng (2015) unterteilen das Baufeld zur Verringerung des Suchraums in ein Raster. Der Algorithmus wird mit dem BIM-Modell des zu erstellenden Gebäudes verknüpft. Es erfolgt eine Abschätzung der benötigten Dimensionen der Lagerflächen anhand des Materialflusses. Lagerflächen innerhalb schon fertig gestellter Gebäudeteile werden in die Berechnungen mit einbezogen. Die Forschung umfasst außerdem die Nutzung metaheuristischer Algorithmen, wie die Schwarmin-telligenz. Yahya und Saka (2014) nutzen einen Algorithmus, der auf dem Verhalten innerhalb eines Bienenvolks basiert (artificial bee colony algorithm), Ning et al. (2010) nutzen einen Amei-

senalgorithmus (max-min ant system). Wang et al. (2015) planen optimale Kranpositionen auf Großbaustellen unter Nutzung von Schwarmintelligenz in Form des Firefly-Algorithmus. Dabei sind bestimmte mögliche Positionen und Krantypen vorgegeben. Die Positionen der anderen Baustelleneinrichtungsgegenstände sowie der benötigt Transportaufwand von und zu jedem Element sind bekannt.

Die bisherigen Lösungsansätze schaffen eine gute Grundlage für die weitere Forschung. Seit Einführung des Building Information Modeling sind einige Ansätze entstanden, die benötigten Informationen aus dem Bauwerksmodell zu beziehen. Eine umfassend integrierte Implementierung des Baustelleneinrichtungsproblems gibt es bisher jedoch nicht, bei allen Ansätzen sind händische Eingaben notwendig. Besonders die automatische Dimensionierung von Baustelleneinrichtungsgegenständen anhand der auf der Baustelle vorherrschenden Gegebenheiten wird bisher kaum betrachtet. Zumeist erfolgt die Auswahl und Positionierung der Turmdrehkrane gesondert und fließt nur passiv in die weitere Planung ein. Im Großteil der Arbeiten werden lediglich rechteckige Flächen untersucht, wodurch sich der Suchraum erheblich verkleinert.

Automatisierte Dimensionierung von Baustelleneinrichtungsgegenständen

Für jedes Bauprojekt muss eine Dimensionierung der benötigten Baustelleneinrichtungselemente erfolgen, um sinnvolle Maße und Kennzahlen zu ermitteln. Unterdimensionierte Elemente können zur Verzögerung des Baufortschrittes führen (zum Beispiel bei zu geringen Lagerflächen); einzelne Arbeitsschritte könnten unausführbar werden (zum Beispiel bei zu geringer Traglast eines Kranes). Überdimensionierte Elemente erhöhen die Kosten (zum Beispiel bei einem zu hohen Kran) und die Wegzeiten (bei zu großen Lagerflächen). Zur Dimensionierung der Elemente werden genauere Informationen über das Bauvorhaben benötigt. Durch die verwendeten Bauweisen, das vorliegende Grundstück, die angrenzenden Liegenschaften und lokale Besonderheiten können verschiedene Randbedingungen und Anforderungen an die Baustelleneinrichtungselemente entstehen. Diese Randbedingungen bewirken Abweichungen in den benötigten Materialien, Lagerräumen, Baumaschinen und Prozessmustern, so dass gute Lösungen für bestimmte Projekte nicht auf andere Projekte übertragbar sind. Die Dimensionierung der Elemente erfolgt zum einen anhand normativer Richtlinien, zum anderen nach dem Bedarf und Materialfluss.

Zur Dimensionierung der Elemente wird ein BIM-Modell des Bauwerkes mit einem zuvor erstellten Bauablaufplan verknüpft. Durch die Hinterlegung von Process Pattern können benötigte Gerätschaften und Arbeitskräfte identifiziert werden. Process Pattern – oder Prozessmuster – können zur Modellierung einzelner Prozesse der Baustelle genutzt werden, um die Planung des Bauablaufs zu vereinfachen und zu unterstützen. Da sich die Bedingungen von Baustelle zu Baustelle stark unterscheiden und Bauwerke im Allgemeinen Einzelstücke sind, bedeutet die Modellierung des gesamten Bauablaufs für ein einzelnes Bauwerk sehr viel Aufwand (Scherer und Schapke, 2014). Um diesen Aufwand zu verringern, wird der Bauablauf in standardisierte und abstrahierte Subprozesse unterteilt, die für verschiedene Baustellen identisch und daher universell wiederverwendbar sind (Benevolenskiy et al., 2012). Durch weitere Zerlegung der Subprozesse in einzelne Teilschritte entstehen dann die Process Pattern. Die Zerlegung des Subprozesses „Erstellen eines Betonpfahles“ führt beispielsweise zu folgenden Einzelschritten:

1. bohren
2. bewehren
3. betonieren
4. aushärten lassen

Durch Angabe von benötigter Dauer, Ressourcen und Platzbedarf können die Process Patterns auch zur Ressourcenplanung eingesetzt werden. Für den Einzelschritt „bohren“ können sich beispielsweise die benötigten Ressourcen „Drehbohrgerät“, „Arbeiter“ und „Bohrbereich“ ergeben. Durch Verknüpfung der Informationen aus dem BIM-Modell, dem Bauablaufplan und den Process Pattern können beispielsweise die maximalen Hublasten der Krane, Anzahl der zeitgleich Beschäftigten, Vorhaltungsmengen der Materialien und Mindestgrößen von Lagerflächen automatisch generiert werden. Die automatisierte Dimensionierung läuft in den folgenden Schritten ab:

1. Extraktion aus dem BIM-Modell: Aus dem BIM-Modell des zu errichtenden Gebäudes können Informationen zu den benötigten Materialien, Lasten, Abhängigkeitsbeziehungen und Einbauorten gewonnen werden. Diese werden in einem Extraktionsschritt in ein XML Dokument exportiert und von dort eingelesen.
 - a) Ermittlung der benötigten Materialmengen: Die Materialien der Bauteile sind mit den Komponenten des BIM-Modells verknüpft, wodurch die Quantität und Qualität, sowie der Einbauort extrahiert werden können. Bei Fertigteilen und großen Bauteilen aus Stahl oder Holz werden die Abmessungen ermittelt, bei vor Ort gefertigten Betonbauteilen die benötigte Menge an Frischbeton sowie Anzahl und Abmessungen von Betonstahl und Schalelementen.
 - b) Ermittlung der maximalen Lasten: Die maximalen vom Kran zu hebenden Lasten ergeben sich normalerweise aus dem Transport von Betonkübeln, großen Fertigteilen oder großen Schalungselementen. Anhand des BIM-Modells können die Elemente mit den höchsten Gewichte und die Einbauorte bestimmt und die Krane entsprechend dimensioniert werden.
2. Informationen über den Bauablauf: Um den zeitlichen Ablauf und die Dauer der einzelnen Tätigkeiten zu bestimmen, wird der Bauablaufplan ausgewertet. Durch Verknüpfung der Elemente im BIM-Modell mit dem Bauablaufplan können die einzelnen Verbrauchsdaten bestimmt werden.
3. Verknüpfung mit Process Pattern: Zur Bestimmung der benötigten Geräte und der Anzahl der auf der Baustelle beschäftigten werden die Elemente aus dem BIM-Modell und die Tätigkeiten aus dem Bauablaufplan mit Process Pattern verknüpft.
4. Dimensionierung der Baustelleneinrichtungsgegenstände nach Regeln: Mit Hilfe der ermittelten Daten können die notwendigen Maße und Eigenschaften der verschiedenen Baustelleneinrichtungsgegenstände anhand der im folgenden Teil beschriebenen Formeln berechnet werden.

Entwicklung von Dimensionierungsregeln für Baustelleneinrichtungsgegenstände

Zur Dimensionierung der Baustelleneinrichtungsgegenstände ergeben sich verschiedene Regeln aus Normen und technischen Regeln, ungeschriebenen Richtlinien und aus dem Bedarf und Materialfluss. Im folgenden wird für die Elemente Kran, Betonpumpe, Lagerflächen und Container beschrieben, wie diese Regeln erfasst werden können.

Kran

Hebezeuge werden für den horizontalen und vertikalen Transport von großen Mengen an Material auf Baustellen benötigt. Die meistgenutzten Hebezeuge auf Baustellen sind Turmdrehkrane, welche Transporte in allen drei Dimensionen ausführen können. Die wichtigsten Elemente eines Turmkranes sind der Ausleger mit Lastaufnahmeeinrichtung, der Turm mit Dreheinheit und die Turmbasis mit Fundament (Abbildung 1a). Zur Abdeckung der gesamten Baustelle können mehrere Krane betrieben werden. Diese sollten so platziert werden, dass sie sich nicht gegenseitig behindern.

Zur Auswahl eines Kranes werden im wesentlichen drei Aspekte betrachtet: Die benötigte Hakenhöhe, die abgedeckte Fläche, und die maximale Last. Diese Korrelieren mit der Höhe des Turms, der Länge des Auslegers und der Tragfähigkeit des Krans. Der Turm des Kranes muss hoch genug sein, um mit einer angehängten Last die umgebenden Hindernisse und das im Bau befindliche Gebäude überschwenken zu können. Zur Ermittlung der Hakenhöhe müssen der Höhenunterschied zwischen dem Fußpunkt des Kranes und dem zu überschwenkenden Baukörper, der benötigte Arbeitsraum inklusive eines Sicherheitsabstandes, sowie die Höhe der angehängten Last und der Lastaufnahmeeinrichtung aufsummiert werden (Abbildung 1b):

$$h_{\min} = (h_{\text{obst.}} - h_{\text{foot}}) + h_{\text{workspace}} + h_{\text{safety}} + h_{\text{load}} + h_{\text{hanger}} \quad (1)$$

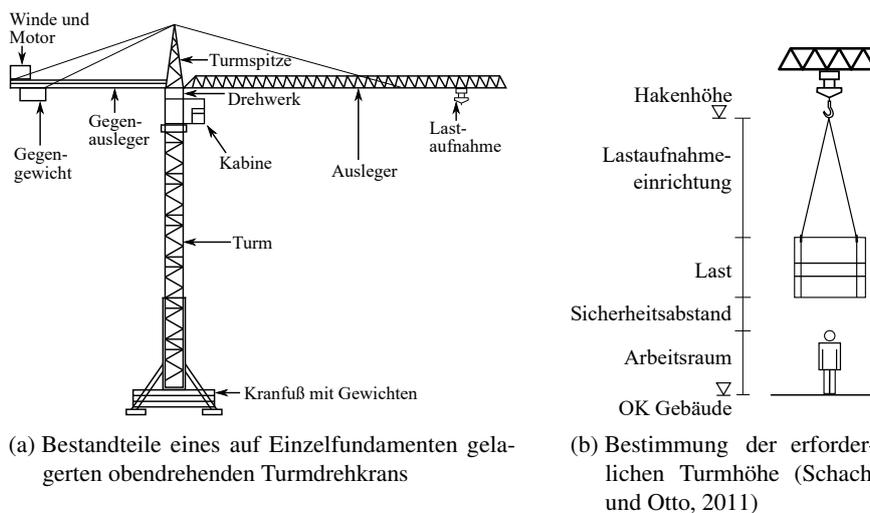


Abb. 1: Bestandteile und erforderliche Turmhöhe eines Turmdrehkrans

Da zur Ermittlung der nötigen Auslegerlänge und des erforderlichen Traglastmoments die Entfernungen der einzelnen Lasten zum Kran benötigt werden, kann die endgültige Dimensionierung erst nach der Positionierung des Kranes erfolgen. Bei der Bestimmung der notwendigen Auslegerlänge sollte beachtet werden, dass alle Lagerflächen und das zu erstellende Gebäude komplett überschwenkt werden können. Zudem sollte jeder Kran die Baustraße an mindestens einer Stelle überschwenken. Die zulässige Traglast ist abhängig von der Auslegerlänge und der Ausladung, also der Entfernung der zu hebenden Last von der Turmachse. Die zunehmender Ausladung nimmt die Tragfähigkeit ab (siehe Abbildung 2). Die Traglastkurve ist dabei in zwei Bereiche unterteilt: Im ersten, konstanten, Bereich, ist die Tragfähigkeit des Hubseil maßgebend für die Tragfähigkeit des Krans; im zweiten, abfallenden, Bereich ist das maximale

Traglastmoment des Kranes maßgebend. Zur Dimensionierung des Krans wird das maximal erforderliche Traglastmoment M_{\max} benötigt. Zur Ermittlung dieser Traglastmomente werden die Produkte aus der Gewichtskraft F aller n zu hebenden Lasten und deren Entfernung r von der Turmachse gebildet:

$$M_{\max} = \max(r_i \cdot F_i) \quad \text{mit } i \in \mathbb{N}, i \in [0, n] \quad (2)$$

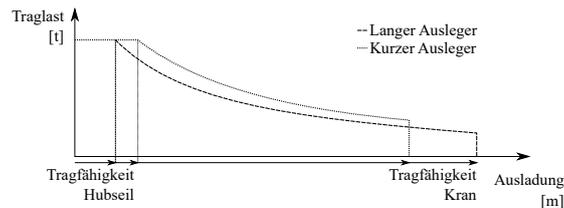


Abb. 2: Parameter zur Bestimmung der erforderlichen Auslegerlänge: Traglastkurven

Betonpumpe

Während des Betoniervorgangs großer Bauteile müssen erhebliche Materialmengen bewegt werden. Zum Transport von Ortbeton zur Einbaustelle können Betonkübel verwendet werden. Diese werden mit einem Kran zur Einbaustelle gehoben und in die Schalung gekippt. Um eine höhere und kontinuierliche Verarbeitungsgeschwindigkeit zu erreichen, und den Kran zu entlasten, können Betonpumpen verwendet werden. Betonpumpen bestehen aus der Pumpe, einem Förderrohr und einem beweglichen Verteilermast (Abbildung 3). Die Pumpen werden im Regelfall durch Fahrmischer mit Beton versorgt. Am häufigsten werden mobile Autobetonpumpen eingesetzt, was den Transport zur Baustelle und den Aufbau erleichtert. Die Betriebsdauer und damit die Mietdauer sind auf den Betoniervorgang begrenzt und können im Voraus geplant werden, um die Kosten zu senken. Stationäre Pumpen können höhere Fließgeschwindigkeiten erreichen und größere Distanzen überbrücken, indem die Förderrohre an der Schalung befestigt werden. Da Transport und Aufbau stationärer Pumpen verhältnismäßig aufwendig sind, werden sie vornehmlich auf Großbaustellen verwendet.

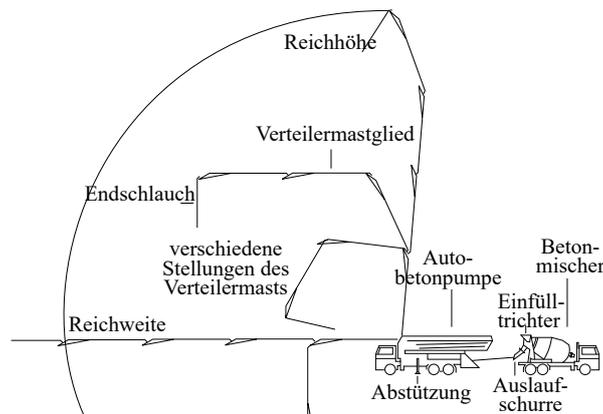


Abb. 3: Bestandteile einer Autobetonpumpe und eines Betonmischers, sowie Reichweite des Verteilermastes mit vier Gliedern in Roll-Faltung

Entscheidend für die Auswahl einer Betonpumpe sind die Auslegerreichweite und die Förderleistung. Alle zu betonierenden Bauteile sollten innerhalb der Reichweite des Auslegers liegen. Die Reichweite des Auslegers ist abhängig von der Ausführung des Verteilermastes. Der Mast besteht in der Regel aus drei bis sechs Abschnitten, und kann gefaltet werden, wenn er nicht in Gebrauch ist. Eine Unterteilung in mehr Glieder erlaubt eine größere Flexibilität. Je nach Art der Faltung und der Länge der Glieder können verschiedene vertikale und horizontale Entfernungen überbrückt werden. Gefaltet werden die Maste entweder in „Roll-Faltung“ durch Aufrollen der Glieder von außen nach innen („Überkopf-Roll-Faltung“ und „Unterkopf-Roll-Faltung“) oder in „Z-Faltung“ durch abwechselndes Einklappen wie bei einem Gliedermaßstab („Multi-Z-Faltung“ und „Roll-Z-Faltung“). Die Reichweite gängiger Autobetonpumpen beträgt etwa 24 m bis 36 m, bei besonderen Anforderungen sind Reichweiten bis 63 m möglich. Die maximalen Fördermengen der Betonpumpen reichen von 15 m³/h bis 200 m³/h. Ein tatsächlicher Einbau in dieser Geschwindigkeit ist allerdings nicht durchführbar. Wird die Betonpumpe durch zwei Fahrmaschinisten bedient, können Förderleistungen von etwa 70 m³/h erreicht werden. Die Einbauleistung von Normalbeton beträgt etwa 30 m³/h (Schach und Otto, 2011).

Lagerflächen

Zur Lagerung der Baumaterialien zwischen Anlieferung und Einbau werden Lagerflächen benötigt. Diese werden entsprechend des zu lagernden Materials dimensioniert. Dabei haben die Dauer der Lagerung und die Vorhaltemenge der jeweiligen Materialien großen Einfluss auf die benötigte Lagerfläche. Bei Direkteinbau großer Bauteile nach Just-in-time-Lieferung werden keine Lagerflächen benötigt. Zunächst müssen der benötigte Flächeninhalt und die Mindestmaße der Seitenlängen bestimmt werden. Zur Vereinfachung der Problemstellung werden hier nur rechteckige Flächen betrachtet.

Der benötigte Flächeninhalt A_{\min} lässt sich bestimmen aus der maximalen Anzahl der zu lagernden Elemente N_{\max} , der benötigten Fläche einer Einheit der zu lagernden Elemente A_e sowie der Anzahl der maximal übereinander stapelbaren Einheiten m :

$$A_{\min} = N_{\max} \cdot \left\lceil \frac{A_e}{m} \right\rceil \quad (3)$$

Zur optimalen Festlegung der maximalen Anzahl der zu lagernden Elemente ist genaue Kenntnis der Liefer- und Lagerkosten notwendig. Zur Abschätzung der Anzahl der zu lagernden Elemente schlagen Kumar und Cheng (2015) daher die Berechnung des Flächeninhalts aus dem Materialverbrauch und dem Rhythmus der Anlieferung vor. Wenn zyklische Anlieferungsrythmen angenommen werden (Lieferungen zu konstanten Zeitintervallen), kann die benötigte Lagerfläche wie folgt berechnet werden:

$$A_{\min} \approx A_e \cdot R \cdot \Delta t \quad (4)$$

Dabei bezeichnet R den maximalen Verbrauch und Δt die Zeit zwischen zwei aufeinander folgenden Lieferungen. Stapelbare Elemente werden von dieser Formel nicht erfasst, zudem hat diese Berechnungsformel nur Gültigkeit, wenn nach Rhythmusystem geliefert wird. Dies bildet gegebenenfalls die Realität auf der Baustelle nicht korrekt ab, da bei Lieferung nach Bestellsystem die Lieferungen nicht in konstanten Intervallen, sondern nach Bedarf ausgelöst werden. Dabei wird eine Bestellung ausgelöst, sobald der Lagerbestand eine festgelegte Menge (Bestellpunkt) unterschreitet. Die Höhe des Bestellpunktes N_{\min} richtet sich nach dem Verbrauch des Materials R , der Lieferdauer Δt_{order} und einer Sicherheitszeit Δt_{safety} .

$$N_{\min} = R \cdot (\Delta t_{\text{order}} + \Delta t_{\text{safety}}) \quad (5)$$

Der maximalen Lagerbestand kann aus der optimalen Bestellmenge ermittelt werden. Die optimale Bestellmenge ergibt sich durch Minimierung der Summe der Kosten für die Lieferung und Lagerung der Elemente (Abbildung 4a). Durch größere Lieferungen können die Fixkosten pro geliefertem Element gesenkt und mögliche Rabatte genutzt werden. Bei zu großen Lieferungen steigen die Lagerkosten durch Kapitalbindung, Lagermaterialien und Beschädigung. Die optimale Bestellmenge n_{opt} kann abgeschätzt werden, indem der Verbrauch R , die Fixkosten pro Bestellung C_{fix} , die variablen Kosten pro Bestellung C_{var} und die Lagerkosten C_{stor} in die Andlersche Losgrößenformel eingesetzt werden (Andler, 1929):

$$n_{\text{opt}} = \sqrt{\frac{200 \cdot R \cdot C_{\text{fix}}}{C_{\text{var}} \cdot C_{\text{stor}}}} \quad (6)$$

Aus der optimalen Bestellmenge und dem Sicherheitsbestand n_{safety} lässt sich der maximale Lagerbestand ermitteln:

$$\begin{aligned} N_{\text{max}} &= n_{\text{opt}} + n_{\text{safety}} \\ \text{mit } n_{\text{safety}} &= R \cdot \Delta t_{\text{safety}} \end{aligned} \quad (7)$$

Die Mindestmaße a_{min} und b_{min} der Seitenlängen der Lagerflächen erschließen sich aus den Maßen der zu lagernden Güter. Bei Lagerflächen für Güter, die auf Paletten oder in Paketen geliefert und gelagert werden, liegt die Mindestseitenlänge bei einer Paletten- bzw. Paketbreite. Bei anderen Stückgütern berechnet sich die Mindestseitenlänge nach den Maßen der jeweiligen Stücke. Lagerflächen für 6,00 m Betonstabstahl beispielsweise benötigen entsprechend mindestens eine Seite der Länge 6,00 m. Bei der Lagerung von Schüttgütern muss der Schüttwinkel beachtet werden, der je nach Schüttgut zwischen 25° und 40° liegt. Liegen Anforderungen an Mindestmaße nicht oder nur für eine Seite vor, so kann ein Mindestmaß von 1,00 m angenommen werden, um sehr schmale Lagerflächen auszuschließen. Im Normalfall genügen die minimalen Seitenlängen nicht zur Abdeckung des benötigten Flächeninhaltes A_{min} . Die genauen Seitenlängen der Lagerflächen können entsprechend des Platzangebots gewählt werden. Dazu werden die Seiten um einen Summanden a_{var} bzw. b_{var} vergrößert:

$$A_{\text{min}} \leq a \cdot b, \quad a = a_{\text{min}} + a_{\text{var}}, \quad b = b_{\text{min}} + b_{\text{var}} \quad (8)$$

Durch Umstellen von Gleichung 8 nach a_{var} wird die Beziehung zwischen a_{var} und b_{var} dargestellt – zu jedem a_{var} gibt es genau ein korrespondierendes b_{var} , so dass die eingeschlossene Fläche genau A_{min} entspricht (vgl. Kumar und Cheng (2015)):

$$a_{\text{var}} \geq \frac{A_{\text{min}} - a_{\text{min}} \cdot b_{\text{min}} - b_{\text{var}} \cdot a_{\text{min}}}{b_{\text{min}} + b_{\text{var}}} \quad (9)$$

Abbildung 4b stellt diese Beziehung beispielhaft für die Werte $A_{\text{min}} = 15 \text{ m}^2$, $a_{\text{min}} = 2 \text{ m}$ und $b_{\text{min}} = 3 \text{ m}$ dar.

Container

Zur Unterbringung der auf der Baustelle beschäftigten Personen und zum Schutz von feuchtigkeitsempfindlichen und diebstahlgefährdeten Gegenständen und Werkzeugen werden Schuppen, Sozial- und Büroräume benötigt. Diese können in bestehende Gebäude auf oder in der Nähe der Baustelle untergebracht werden, oder es werden temporäre Baracken, Bauwagen oder Container aufgestellt, wobei vorrangig Container genutzt werden. Für eine schnelle und fehlerfreie

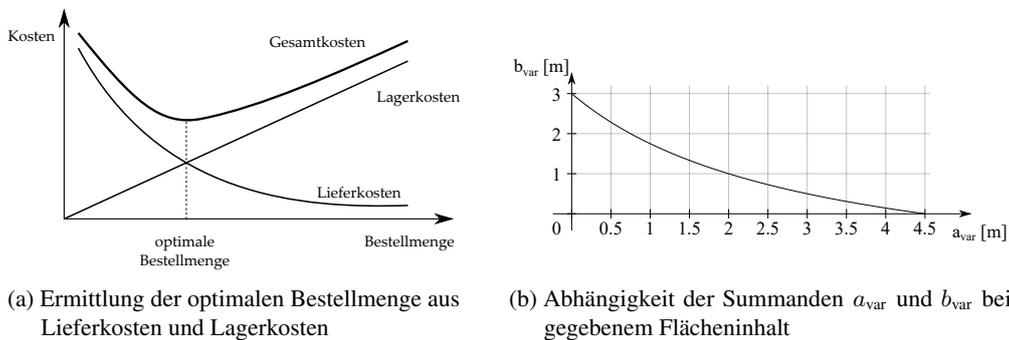


Abb. 4: Graphische Darstellung der Bestimmung verschiedener Parameter im Rahmen der Dimensionierung von Lagerflächen

Montage werden Container in einem Stück angeliefert und mit Hilfe eines Kranes auf der Baustelle platziert. Es gibt offene und geschlossene Container-Systeme. Geschlossene Container sind komplett vormontierte Einheiten mit vier geschlossenen Außenwänden; offene Container sind mit herausnehmbaren Wänden ausgestattet und können zu größeren Einheiten verbunden werden. Entsprechend den Anforderungen und der Größe des Bauvorhabens sind verschiedene Räumlichkeiten erforderlich:

- Tagesunterkünfte (Pausen- und Umkleieräume)
- Sanitäre Anlagen
- Schwarz-Weiß-Container
- Werkzeugschuppen / Lagerung
- Übernachtungsmöglichkeiten
- Erste-Hilfe-Raum
- Büroräume
- Labore

ISO 668:2016-03 definiert standardisierte Container, deren Abmessungen auf einer Grundform von 8 Fuß×40 Fuß (2,44 m×12,19 m) basieren. So können mehrere Container passgenau miteinander verbunden werden. Die am häufigsten verwendete Größen sind in Tabelle 1 dargestellt. Die Mindestanforderungen an die Räumlichkeiten, Flächen und Ausstattung sind in der Arbeitsstättenverordnung 2004 und in den technischen Regeln für Arbeitsstätten vorgegeben (BAuA, 2016). Einige ausgewählte Regeln werden vereinfacht in Tabelle 2 dargestellt.

Tabelle 1: Außenmaße einiger häufig genutzter Container nach ISO 668:2016-03

Label	Abmessungen [m]			Gewicht [kg]		Nutzung
	Länge	Breite	Höhe	von	bis	
1AA	12,19	2,44	2,59	3200	4000	Lagerung
1CC	6,06	2,44	2,59	1900	2900	Tagesunterkünfte, Toiletten, Sanitärobjekte, Lagerung
1DD	2,99	2,44	2,59	1300	2100	Tagesunterkünfte, Toiletten, Lagerung

Tabelle 2: Mindestanforderungen an Sozialeinrichtungen nach der Anzahl der Beschäftigten (BAuA, 2016)

Anzahl der Nutzer	Waschplätze	Duschplätze	Toiletten/Urinale
1 bis 5	1	0	1
6 bis 10	2	0	1
11 bis 20	3	1	2
21 bis 30	5	1	3
⋮	⋮	⋮	⋮
Pausenräume:	Grundfläche mindestens 6,00 m ² 1,00 m ² Fläche, Sitzgelegenheit und Tisch pro Beschäftigtem		
Erste Hilfe:	ein großer Verbandkasten je 50 Beschäftigte ab 50 Beschäftigten: Erste-Hilfe-Container mit 12,50 m ²		
Unterkünfte:	1 bis 6 Personen pro Schlafbereich: 8,00 m ² pro Person 6 bis 8 Personen pro Schlafbereich: 8,75 m ² pro Person		

Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Konzept zur automatisierten Dimensionierung von Baustelleneinrichtungsgegenständen vorgestellt. Dazu wurden maßgebende Elemente der Baustelleneinrichtung identifiziert. Dimensionierungsvorschriften aus normativen Regelwerken und logische Schlussfolgerungen wurden genutzt, um eindeutige Regeln zu erstellen. Das Vorgehen bei der computerbasierten Dimensionierung wurde erläutert und erforderliche Informationsquellen genannt. Im nächsten Schritt wird die automatisierte Positionierung der dimensionierten Baustelleneinrichtungsgegenstände untersucht. Dazu werden zunächst bestehende Regeln und Vorschriften ermittelt und zulässige sowie unzulässige Positionen der einzelnen Elemente identifiziert. Die anschließende Positionierung der Elemente erfolgt unter Nutzung eines Optimierungsalgorithmus, der ein möglichst effizientes und kostensparendes Layout entwickelt.

Literatur

- Andayesh, M. und Sadeghpour, F. (2013). „Dynamic site layout planning through minimization of total potential energy“. In: *Automation in Construction* 31, S. 92–102.
- Andler, K. (1929). *Rationalisierung der Fabrikation und optimale Losgröße*. München, Deutschland: Oldenbourg.
- BAuA (2016). *Arbeitsstätten. Arbeitsstättenverordnung, Technische Regeln für Arbeitsstätten*. Verordnung. Dortmund, Deutschland: Bundesanstalt für Arbeitsschutz und Arbeitsmedizin (Hrsg.)
- Benevolenskiy, A., Roos, K., Katranuschkov, P. und Scherer, R. J. (2012). „Construction processes configuration using process patterns“. In: *Advanced Engineering Informatics* 26.4, S. 727–736.
- Borrmann, A., König, M., Koch, C. und Beetz, J. (2015). *Building Information Modeling*. Wiesbaden, Deutschland: Vieweg + Teubner Verlag, p.
- Huang, C. und Wong, C. (2015). „Optimisation of site layout planning for multiple construction stages with safety considerations and requirements“. In: *Automation in Construction* 53, S. 58–68.

- ISO 668:2016-03 (2016). *Series 1 freight containers - Classification, dimensions and ratings*. ISO Standard. Genf, Schweiz: International Organization for Standardization.
- Krepp, S., Jahr, K., Bigontina, S., Bügler, M. und Borrmann, A. (2016). „BIMsite - Towards a BIM-based Generation and Evaluation of Realization Variants Comprising Construction Methods, Site Layouts and Schedules“. In: *Proceedings of the EG-ICE Workshop on Intelligent Computing in Engineering*. Krakau, Polen.
- Kumar, S. S. und Cheng, J. C. P. (2015). „A BIM-based automated site layout planning framework for congested construction sites“. In: 59, S. 24–37.
- Meyran, G. (1973). „Optimierungsfragen der Baustelleneinrichtung“. Diss. Technische Universität München, Deutschland.
- Ning, X., Lam, K. und Lam, M. (2010). „Dynamic construction site layout planning using max-min ant system“. In: *Automation in Construction* 19.1, S. 55–65.
- Schach, R. und Otto, J. (2011). *Baustelleneinrichtung: Grundlagen - Planung - Praxishinweise - Vorschriften und Regeln*. Leitfaden des Baubetriebs und der Bauwirtschaft. Wiesbaden, Deutschland: Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH.
- Scherer, R. J. und Schapke, S.-e., Hrsg. (2014). *Informationssysteme im Bauwesen 1*. Heidelberg, Deutschland: Springer VDI.
- Schneider, K.-J. (2016). *Bautabellen für Ingenieure: mit Berechnungshinweisen und Beispielen*. 22. Aufl. Köln, Deutschland: Bundesanzeiger.
- Schwabe, K. (2015). „BIM-basierte Baustelleneinrichtungsplanung“. Master Thesis. Bochum, Deutschland: RUB.
- Shawki, K. M., El-Razek, M. E. A. und Maqboly, S. A. (2010). „Optimal Arrangement of temporary Facilities in Construction Sites“. In: *Journal of Engineering Sciences* 38.4, S. 949–960.
- Tommelein, I. D., Levitt, R. E. und Hayes-Roth, B. (1992). „SightPlan Model for Site Layout“. In: *Journal of Construction Engineering and Management* 118.4, S. 749–766.
- Wang, J., Zhang, X., Shou, W., Wang, X., Xu, B., Kim, M. J. und Wu, P. (2015). „A BIM-based approach for automated tower crane layout planning“. In: *Automation in Construction* 59, S. 168–178.
- Yahya, M. und Saka, M. (2014). „Construction site layout planning using multi-objective artificial bee colony algorithm with Levy flights“. In: *Automation in Construction* 38, S. 14–29.

Optimierung von Gebietszerlegungen mit Hilfe der Partikelschwarmoptimierung

Marco Behrendt¹, Sebastian Brandt² und Christoph Eckert³

Institut für Risiko und Zuverlässigkeit, Leibniz Universität Hannover

¹E-Mail: marco-behrendt@web.de, ²E-Mail: brandt@irz.uni-hannover.de,

³E-Mail: eckert@irz.uni-hannover.de

Abstract An outstanding problem in the field of finite element mesh generation is still creating automatically analysis suitable, high quality and geometric exact meshes in a reasonable time. Therefore, we plug a particle swarm optimization in the mesh generation algorithms in order to increase the degree of automation. In a first step, we survey triangulations of polynomial bounded two-dimensional domains, which will later be extended to the three-dimensional case. The algorithm of Cavendish (1974) is analyzed with regard to the eligibility in terms of a particle swarm optimization. Finally, the optimized algorithm is verified by several examples.

Keywords: domain decomposition, mesh generation, algorithm of Cavendish (1974), particle swarm optimization

1 Einleitung

Viele Problemstellungen in den Ingenieurwissenschaften werden durch Differentialgleichungen beschrieben und numerisch gelöst, weil analytische Lösungen nicht existieren oder zu komplex sind. Ein etabliertes Verfahren zur Lösung von Differentialgleichungen ist die Finite-Element-Methode (FEM), die beispielsweise für das Lösen der Wärmeleitungsgleichung oder Wellengleichung verwendet wird. Der Grundgedanke der Methode besteht in der Zerlegung des betrachteten Gebietes in endlich viele Elemente, auf denen die Gleichung jeweils gelöst wird. Deshalb hängt das Ergebnis der Finiten-Element-Analyse maßgeblich von der Qualität der Zerlegung ab (Brenner und Scott, 1994). Koexistent zur FEM besteht das Problem der automatisierten Gebietszerlegung, das insbesondere im dreidimensionalen für beliebige Gebiete nicht trivial ist. Bern und Eppstein (1992) geben einen detaillierten Überblick über die wichtigsten Algorithmen in diesem Bereich. Für viele der Algorithmen sind vom Anwender a priori Parameter festzulegen, die insbesondere bei mannigfaltigen Geometrien einen essentiellen Einfluss auf die Qualität der Zerlegung haben. In dieser Arbeit untersuchen wir zunächst den Algorithmus von Cavendish (1974), der ein zu zerlegendes Gebiet mit sich überschneidenden Kreisen, dessen Radien vom Anwender festgelegt werden muss, ausfüllt, um anschließend in diese Kreise Punkte zu platzieren, die die Elemente der Zerlegung definieren. Damit sich die Wahl solcher und ähnlicher Parameter erübrigt, wird in der vorliegenden Arbeit der Algorithmus mit einer Partikelschwarmoptimierung gekoppelt.

Die Arbeit gliedert sich wie folgt: In Abschnitt 2 wird der kreisbasierte Algorithmus von Cavendish, 1974 hinsichtlich seiner Eignung für die Partikelschwarmoptimierung analysiert. Anschließend wird in Abschnitt 3 ein Partikelschwarmoptimierungsalgorithmus vorgestellt, der die Parameter des Algorithmus a priori für ein beliebiges Gebiet optimiert. Anwendungsbeispiele des optimierten Algorithmus werden in Abschnitt 4 illustriert, sowie die Ergebnisse diskutiert.

2 Analyse des Algorithmus von Cavendish

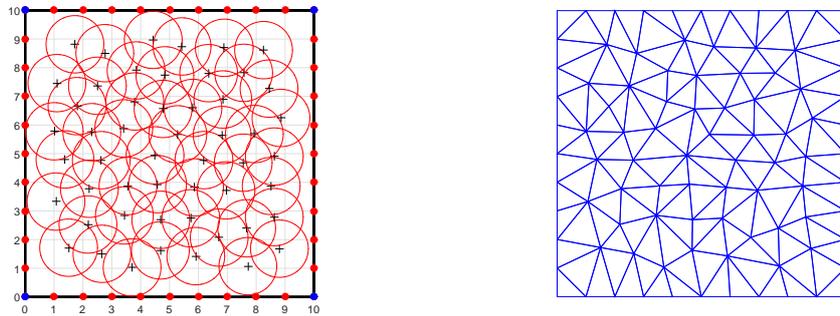
Das Verfahren von Cavendish (1974) soll im folgenden Abschnitt für Gebiete, die durch ein geschlossenes Polygon berandet sind, analysiert werden. Zunächst wird das Schema des Algorithmus erläutert und potentielle Parameter für die Partikelschwarmoptimierung beurteilt. Des Weiteren wird ein Qualitätsmaß für ein generiertes Netz aus Elementen eingeführt.

2.1 Schema

Ziel der gesamten Netzgenerierung ist die Zerlegung eines polynomiell berandeten Gebietes in Dreiecke. Zu Beginn wird dem Algorithmus das Randpolygon in einer geordneten Liste von Randpunkten übergeben. Aus dieser Information werden sogenannte Steinerpunkte im Inneren konstruiert, die zusammen mit den Randpunkten die Delaunay-Triangulierung (Delaunay, 1934) erzeugen. Damit für die Delaunay-Triangulierung nicht nur zusätzliche Punkte im Inneren des Gebietes zur Verfügung stehen und der Rand nur mit großen Elementen beschrieben wird, wird zunächst der Gebietsrand weiter diskretisiert. Dazu muss dem Algorithmus ein Parameter k übergeben werden, der den maximalen Abstand zwischen den neu erzeugten Randpunkten festlegt. Mit diesem Parameter werden alle Polygonseiten, die länger als k sind, in Teilstrecken mit einer maximalen Länge von k unterteilt. Anschließend kann der Nutzer auf dem Gebiet verschiedene Bereiche Z_i definieren, für die jeweils ein Radius r_i , der den minimalen Abstand zwischen zwei Punkten definiert, angegeben werden muss. Je kleiner r_i ist, desto mehr Punkte werden im Inneren des Teilgebietes Z_i erstellt. Der Einfachheit halber wird in dieser Arbeit das gesamte Gebiet als ein Bereich Z mit korrespondierendem Radius r definiert.

Um Steinerpunkte im Inneren des Gebietes zu setzen, wird die Bounding-Box des gesamten Gebietes bestimmt. Anschließend wird über die gesamte Bounding-Box ein Background-Mesh von Quadraten mit der Seitenlänge des Radius r erzeugt. Bei einem Durchlauf über alle Quadrate wird versucht, in jedem Quadrat einen Punkt zu setzen, der nur gesetzt werden darf, wenn in dem Umkreis des Steinerpunktes mit Radius r kein Randpunkt oder anderer Steinerpunkt aus den umliegenden Quadraten liegt - vgl. Abbildung 1 (a).

Das zufällige Setzen eines Steinerpunktes in ein Quadrat des Background-Mesh führt zu einem ungewollten Phänomen. Desto weiter der Punkt vom Zentrum seines Quadrates entfernt ist, desto schwerer ist es in ein benachbartes Quadrat einen anderen Steinerpunkt zu setzen. Deshalb ist es sinnvoll - wider der Strategie von Cavendish (1974) - keine Gleichverteilung über den Quadraten für die zufällige Wahl der Steinerpunkte heranzuziehen. Aufgrund dieser Feststellung wird eine Normalverteilung eingeführt, die sicherstellt, dass eine größere Entfernung vom Quadratmittelpunkt möglich ist, jedoch zunehmend unwahrscheinlicher wird. Als Mittelwerte der Verteilungen werden die jeweiligen x - und y -Koordinaten des Quadratmittelpunktes gewählt. Als Standardabweichung für alle Quadrate wird $\sigma \in [0, r]$ angenommen. Dabei bedeutet



- (a) Polygon mit den gegebenen Eckpunkten (blau), neu erstellten Randpunkten (rot) und gesetzten Kreisen, deren Mittelpunkte neue Punkte der Triangulation sind
- (b) Erzeugte Delaunay Triangulation aus den erstellten Punkten

Abb. 1: Vorgehen des Algorithmus

eine Abweichung von $\sigma = 0$, dass immer der Mittelpunkt des jeweiligen Quadrates verwendet wird, und $\sigma = r$, dass die durchschnittliche Abweichung vom Mittelpunkt des jeweiligen Quadrates r beträgt. Für die Standardabweichung σ sind in der Literatur keine eindeutigen empfohlenen Werte gefunden worden, weshalb der Einfluss dieses Parameters in Abschnitt 2.3 näher untersucht wird. Sollte der zufällige Steinerpunkt außerhalb seines Quadrates liegen, wird er auf den Rand des Quadrates projiziert.

Weil nicht sichergestellt werden kann, dass das Setzen eines Steinerpunktes in ein Quadrat erfolgreich ist, schlägt Cavendish (1974) vor den Vorgang fünf Mal zu wiederholen. Dieser willkürlich festgelegte Parameterwert eignet sich ebenfalls für eine genauere Analyse hinsichtlich Optimierungspotential (Abschnitt 2.3).

Das bis hier vorgestellte Schema beschreibt den Algorithmus von Cavendish (1974) - vgl. Abbildung 1 (a). Die übergebenen und konstruierten Randpunkte, sowie die im Inneren liegenden Steinerpunkte bilden die Grundlage für die Triangulierung, die mit dem Algorithmus von Delaunay (1934) erzeugt wird. Abbildung 1 (b) illustriert die Triangulierung eines Quadrates für das in Abbildung 1 (a) gezeigte Gebietszerlegungsproblem mit den Parametern $k = 1$, $r = 1$ und $\sigma = 0.2r$.

2.2 Qualitätsmaß

Wie bereits in Abschnitt 1 erwähnt, kann die Gebietszerlegung einen erheblichen Einfluss auf die Qualität der Analyse - beispielsweise eine FE-Berechnung - haben. Es existieren verschiedene Maße, die je nach Verwendungszweck die Netzqualität beziffern. Eine Auswahl von Qualitätsmaßen ist zum Beispiel in Bern und Eppstein (1992) oder Knupp (2001) zu finden.

In dieser Arbeit wird ein Qualitätsmaß verwendet, welches Dreieckselemente hinsichtlich ihrer Abweichung von einem gleichseitigen Dreieck bewertet. Für ein beliebiges Dreieck $T(p_1, p_2, p_3)$ mit den Eckpunkten p_1 , p_2 und p_3 wird das Maß

$$q(T) = \frac{\min(\overline{p_i p_j})}{\max(\overline{p_i p_j})} \in (0, 1] \quad \text{mit} \quad 1 \leq i, j \leq 3, i \neq j, \quad (1)$$

als Verhältnis der kürzesten zur längsten Kante eines Dreiecks definiert. Mit der besten Bewertung ($q = 1$) werden nach Gleichung (1) nur gleichseitige Dreiecke bewertet. Desto mehr ein Dreieck von diesem abweicht, desto größer wird die Differenz zwischen seiner größten und kleinsten Kante und die Qualität nimmt folglich ab. Bei einem entarteten Dreieck, welches eine sehr kleine kürzeste, sowie eine sehr große längste Seite besitzt, konvergiert das Qualitätsmaß gegen die schlechteste Bewertung ($q = 0$). Grund für die Wahl dieses Maßes ist vor allem die einfache Berechnung und Implementierung. Ein sinnvolleres Maß im Hinblick auf die Analyse ist sicherlich die Qualität über eine Transformationsabbildung auf ein Referenzelement zu messen, dessen Untersuchung für weiterführende Arbeiten angedacht ist.

Die Bewertung eines kompletten Dreiecksnetzes kann beispielsweise über den Mittelwert, das Maximum oder das Minimum aller Dreiecke erfolgen - vgl. Bern und Eppstein (1992). In dieser Arbeit soll für die Bewertung der Mittelwert aller einzelnen Dreiecksbewertungen verwendet werden. Dies hat den Vorteil, dass möglichst viele Dreiecke die Qualität der gesamten Gebietszerlegung bestimmen. Ein Nachteil dieser Gesamtbewertung ist jedoch, dass für Gebietszerlegungen mit unterschiedlichem r nicht die gleiche Anzahl an Dreiecken in die Gesamtbewertung einfließt.

2.3 Einfluss der Parameter

Der in Abschnitt 2.1 beschriebene Algorithmus von Cavendish, 1974 enthält drei variable Parameter:

- Radius r
- Randkantenlänge k
- Standardabweichung σ

Die Einflüsse dieser Parameter auf die Qualität der erzeugten Gebietszerlegungen werden in diesem Abschnitt näher untersucht. Durch den Radius r wird die Anzahl an Punkten innerhalb des Gebiets maßgeblich beeinflusst, und somit direkt die Anzahl und Größe der Elemente. Mit der Standardabweichung σ kann gesteuert werden, wie weit die inneren Punkte vom Mittelpunkt ihres jeweiligen Quadrates im Mittel abweichen dürfen. Über die Kantenlänge k wird die Anzahl an zusätzlich zu erzeugenden Gebietsrandpunkten ermittelt.

Anzahl der Gebietszerlegungen

Aufgrund des stochastischen Einflusses bei dem Setzen der inneren Punkte liefert der Algorithmus von Cavendish bei gleichbleibenden Parametereinstellungen jedes Mal eine unterschiedliche Gebietszerlegung für ein gegebenes Gebiet. Dies bedeutet, dass bei nur einer Gebietszerlegung nicht abschließend beurteilt werden kann, ob das gewählte Algorithmusparameter-Set generell zu einer besseren Gebietszerlegung führt oder dies nur zufällig für die erzeugte Gebietszerlegung der Fall ist. Daher wird in diesem Abschnitt zunächst untersucht, wie viele Wiederholungen mindestens erforderlich sind, um zwei unterschiedliche Algorithmusparameter-Sets qualitativ hinsichtlich besserer Gebietszerlegungsergebnisse zu vergleichen.

Wie in Abschnitt 2.1 bereits gefordert, steht ebenfalls eine abschließende Wahl der Anzahl der Versuche zum Setzen eines Punktes innerhalb eines Quadrates aus. Da nicht ausgeschlossen werden kann, dass die Anzahl der Versuche auch einen Einfluss auf die Varianz der mit einem

Parameter-Set durchgeführten Gebietszerlegungen besitzt, wird dessen Einfluss zusammen mit der erforderlichen Anzahl von Wiederholungen untersucht.

Für die Analyse wird ein quadratisches Beispielgebiet mit einer Seitenlänge von 10 definiert. Als Algorithmusparameter-Set wird $r = 0.8$, $\sigma = 0.2$ und $k = 1.0$ gewählt. In Abbildung 2 ist auf der x -Achse die Anzahl von Wiederholungen und auf der y -Achse die durchschnittliche Qualität des Gebiets für die jeweilige Anzahl von Wiederholungen aufgetragen. Die Qualität jedes einzelnen Dreiecks wird nach Gleichung (1) berechnet. Die Bewertung wird für die unterschiedliche Anzahl von Versuchen 5, 10, 25 und 50 zum Setzen eines Steinerpunktes innerhalb eines Quadrates durchgeführt.

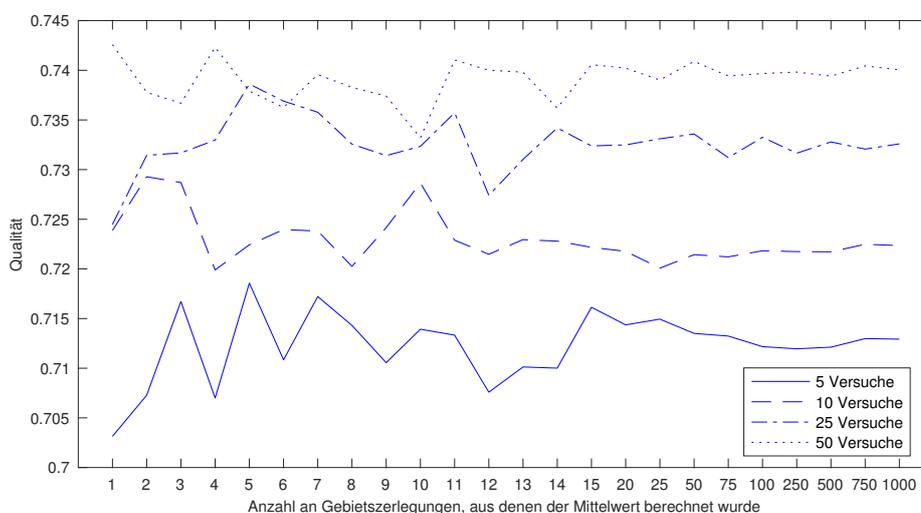


Abb. 2: Untersuchung der erforderlichen Anzahl von Gebietszerlegungen zur Mittelwertbildung

Die in Abbildung 2 dargestellten Ergebnisse zeigen, dass die Schwankung der sich ergebenden durchschnittlichen Bewertungen mit steigender Anzahl von Wiederholungen geringer werden. Es kann zusammenfassend gesagt werden, dass mindestens 100 Wiederholungen ein und derselben Gebietszerlegung für das gewählte Algorithmusparameter-Set erforderlich sind, um eine qualitativ gute Aussage zum durchschnittlichen Qualitätsniveau treffen zu können. Außerdem ist ersichtlich, dass sich mit steigender Anzahl von Versuchen ein Punkt innerhalb eines Quadrates zu setzen, die erreichte durchschnittliche maximale Qualität der Gebietszerlegungen erhöht. Es ist demnach allgemein gesehen offenbar besser einen Punkt in jedes Quadrat zu setzen, als keinen Punkt zu setzen.

Für den weiteren Verlauf dieser Arbeit wird eine minimale Wiederholung von 100 Gebietszerlegungen für die Bewertung eines Algorithmusparameter-Sets empfohlen. Außerdem wird im Sinne möglichst optimaler Gebietszerlegungsergebnisse eine Anzahl von 50 Versuchen für das Setzen eines Punktes in einem Quadrat für alle folgenden Analysen festgelegt.

Standardabweichung

In dem vorherigen Abschnitt wurden Empfehlungen für eine minimale Wiederholung von Gebietszerlegungen und die Anzahl an Versuchen zum Setzen eines Steinerpunktes gegeben.

Somit stehen der in diesem Abschnitt durchgeführten Analyse des Einflusses der Standardabweichung σ geeignete Grundeinstellungen für den Algorithmus und den Umgang mit den stochastischen Einflüssen zur Verfügung. Im Folgenden werden zwei unterschiedliche Analysen durchgeführt. Zum einen werden die (ungestörten) Einflüsse der Standardabweichung auf die Qualität der Gebietszerlegung im Inneren eines Gebietes untersucht. Zum anderen wird für das im vorherigen Abschnitt definierte Beispiel exemplarisch analysiert, wie sich die Standardabweichung auf die Qualität der gesamten Gebietszerlegung auswirkt.

Für die Analyse des ungestörten Einflusses der Standardabweichung auf die Qualität der Gebietszerlegung im Inneren eines Gebietes wird ein Modell des Gebietsinneren verwendet. Es wird ein Raster von 2×2 Quadraten betrachtet, die alle eine Kantenlängen von $r = 1$ besitzen. Bei dem Setzen der Punkte in den einzelnen Quadraten wird wie in dem in Abschnitt 2.1 vorgestellten Algorithmus von Cavendish vorgegangen. Die einzige Ausnahme besteht darin, dass die Randpunkte bzw. Eckpunkte des umhüllenden Quadrates der vier Einzelquadrate innerhalb des Radius der entsprechenden Punkte liegen dürfen. Mit den vier gesetzten Punkten wird abschließend eine Delaunay-Triangulierung durchgeführt und der Mittelwert der beiden entstandenen Dreiecke als Bewertung eines Versuches errechnet. Da dieser Versuch ebenfalls durch stochastische Effekte beeinflusst wird, werden pro untersuchter Standardabweichung 10000 Versuche durchgeführt, demnach insgesamt 20000 Dreiecke evaluiert. Die Standardabweichung wird dabei von 0.0 bis 1.0 mit einer Schrittweite von 0.01 erhöht. Das Ergebnis dieser Analyse ist in Abbildung 3 (a) dargestellt.

Die zweite Analyse wurde anhand eines Beispielgebietes der Größe 10×10 mit einem Radius $r = 0.8$ und einer Randkantenlänge $k = 1$ durchgeführt. In Abbildung 3 (b) sind die Ergebnisse dargestellt, welche den Einfluss der Standardabweichung auf die Qualität der gesamten Gebietszerlegung aufzeigen. Dabei wurde jede Gebietszerlegung 1000 Mal wiederholt. Die Standardabweichung ist dabei von 0.05 bis 1.0 mit einer Schrittweite von 0.1 erhöht worden und pro Schritt wurde aus 1000 Gebietszerlegungen der Durchschnitt ermittelt.

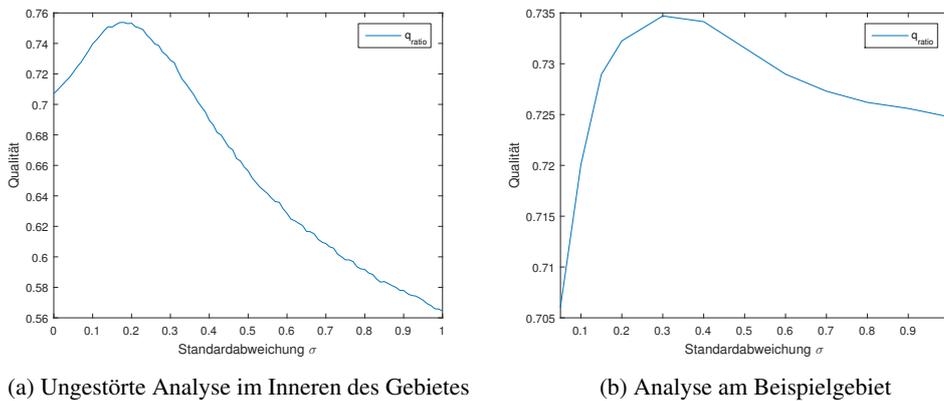


Abb. 3: Einfluss der Standardabweichung auf die Qualität der Gebietszerlegung

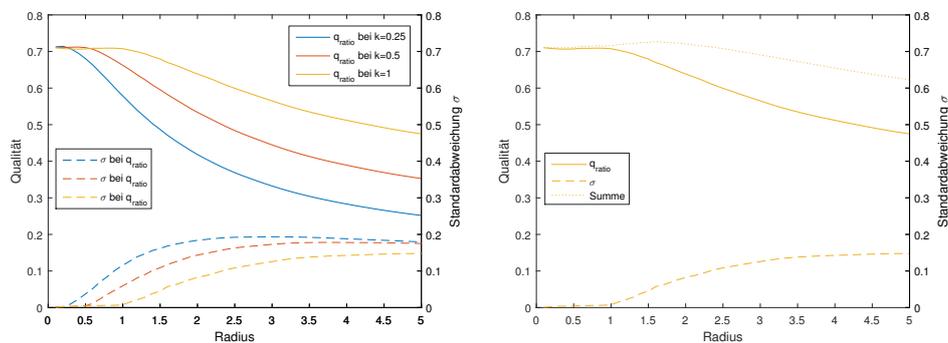
In Abbildung 3 (a) ist zu erkennen, dass bei einer Standardabweichung von $\sigma = 0.2$ im Mittel eine maximale Qualität von $q = 0.75$ erreicht wird. Standardabweichungen von $\sigma > 0.4$ sind offenbar im Gebietsinneren nicht sinnvoll, weil keine höhere mittlere Qualität erreicht wird als für $\sigma \in [0; 0.4]$. Der in 3 (b) dargestellte Verlauf der mittleren erreichten Qualität einer gesamten Gebietszerlegung, zeigt deutlich, dass für eine Standardabweichung nahe 0 die

Qualität im Mittel am schlechtesten ist. Eine optimale Wahl der Standardabweichung ergibt sich für $\sigma = 0.32$, dabei wird im Mittel eine maximale Qualität von $q = 0.734$ erreicht.

Im Vergleich ist deutlich zu sehen, dass sich der Einfluss der Standardabweichung im idealen Modell des inneren Gebietes deutlich von dem Einfluss im Beispielgebiet unterscheidet. Die erreichte maximale Qualität ist im Beispiel etwas geringer und die Wahl eines optimalen σ fällt etwas höher aus. Diese Veränderung lässt vermuten, dass für jedes Zusammenspiel von dem Radius r und der Randkantenlänge k immer auch eine optimale Standardabweichung σ existiert. Außerdem basiert sowohl das Modell des Inneren, als auch das Beispielgebiet auf einem quadratischen Gebiet. Da jedoch bereits der Algorithmus von Cavendish per Design quadratische Gebiete besser bewältigen kann, wird erst eine abschließende Analyse im Optimierungsteil dieser Arbeit (vgl. Abschnitt 4.1) zeigen, welchen Einfluss die Standardabweichung auf die Qualität der Zerlegung eines komplexeren Gebietes besitzt.

Radius und Randkantenlänge

In diesem Abschnitt soll der Einfluss des Zusammenspiels von dem Radius r und der Randkantenlänge k näher untersucht werden. Dazu wird erneut das Beispielgebiet aus den beiden vorherigen Abschnitten betrachtet. Die Standardabweichung wird auf $\sigma = 0.2$ gesetzt und jede Gebietszerlegung, wie in Abschnitt 2.3 gefordert, 100 Mal durchgeführt. Um die stochastischen Effekte besser einschätzen zu können wird außerdem die Standardabweichung unter allen Gebietszerlegungs-evaluierungen eines Parameter-Sets analysiert.



(a) Qualität bei fester Kantenlänge variierendem Radius (b) Summe der Qualität q und deren Standardabweichung σ

Abb. 4: Einfluss des Radius r und der Randkantenlänge k auf die Qualität der Gebietszerlegung

In Abbildung 4 (a) sind die Ergebnisse für die drei Randkantenlängen $k \in \{0.25, 0.5, 1.0\}$ für einen Radiuswertebereich zwischen $0.1 \leq r \leq 5$ (Schrittweite: 0.1) dargestellt. Es ist ersichtlich, dass unabhängig der Randkantenlänge k immer eine maximale Gebietszerlegungsqualität von 0.71 erreicht wird. Lediglich der zum Erreichen dieses Maximums erforderliche Radius variiert je nach gewähltem k . Es fällt außerdem auf, dass der Verlauf der Standardabweichung unter allen 100 evaluierten Gebietszerlegungen erst nach dem Erreichen des Höhepunktes der Qualität stark ansteigt.

Mit der Abbildung 4 (b) soll noch einmal stellvertretend für die drei unterschiedlichen Verläufe verdeutlicht werden, wie enorm der stochastische Einfluss durch das zufällige Setzen der Punkte innerhalb der Quadrate im Inneren des Gebietes ist. Die gepunktete Linie setzt sich

aus der Summe des Mittelwertes (durchgezogene Linie) und der Standardabweichung der 100 einzelnen Evaluationen (gestrichelte Linie) zusammen. Wie erwartet, führt die Wahl einer guten Kombination aus r und k nahe des Hochpunktes am zuverlässigsten zu einer guten Gebietszerlegung. Es ist jedoch ebenfalls ersichtlich, dass im Mittel zwar durch eine schlechte Wahl von r und k keine guten Gebietszerlegungen zu erwarten sind, die hohe Standardabweichung aber dennoch dafür sorgen kann, dass ebenfalls gute Gebietszerlegungen mit dieser Wahl möglich sind. Weiterhin verdeutlicht der Verlauf der Summe, wie wichtig es ist, die im Abschnitt 2.3 empfohlene Anzahl von Wiederholungen pro Parameter-Set bei der folgenden Optimierung zu beachten. Wird die Empfehlung missachtet, besteht die Gefahr, dass während der Optimierung in Abschnitt 4.1 Parameter-Sets für optimal befunden werden, die jedoch nicht repräsentativ sind, sondern viel mehr nur den stochastischen Einflüssen und einer zu kleinen Stichprobe entspringen.

Ein Vergleich der drei Verläufe untereinander zeigt, dass die besten Gebietszerlegungsergebnisse (Hochpunkte der Verläufe) immer bei einem Verhältnis von $0.8 \leq \frac{r}{k} \leq 0.9$ zu erwarten sind. Dieses Verhalten legt eine Kopplung der beiden Parameter nahe. Daher wird im Folgenden für die Optimierung das Verhältnisses $v = \frac{r}{k}$ eingeführt. Dies hat den Vorteil, dass nur noch die Randkantenlänge k und das Verhältnis v angegeben werden muss und der Radius mit $r = v \cdot k$ automatisch in einem lohnenswerten Bereich liegt. Außerdem wird so verhindert, dass r und k in einem schlechten Verhältnis stehen und zu dessen Gebietszerlegungsergebnissen eine hohe Standardabweichung existiert, was das Risiko von nicht aussagekräftigen Ergebnissen minimiert.

3 Partikelschwarmoptimierung

Die Partikelschwarmoptimierung (PSO) ist eine naturinspirierte Metaheuristik, die sich an dem Verhalten von Schwärmen im Tierreich orientiert. Der Partikelschwarmoptimierungsalgorithmus wurde erstmals von Eberhart und Kennedy, 1995 vorgestellt. Seit der ersten Formulierung sind zahlreiche Anpassungen des Grundalgorithmus für diverse spezifische Problemstellungen veröffentlicht worden. In dieser Arbeit wird eine Mehrzielvariante von Reyes-Sierra und Coello, 2006 verwendet. Obwohl die im folgenden Abschnitt durchgeführte Optimierung zunächst nur hinsichtlich des in Abschnitt 2.2 vorgestellten Qualitätskriteriums erfolgt, wird bereits durch die Wahl eines Mehrziel-PSO-Algorithmus (MOPSO) die in Zukunft geplante Integration noch weiterer parallel zu beachtender Optimierungsziele vereinfacht. Algorithmus 1 verdeutlicht den generellen Ablauf des verwendeten MOPSO-Algorithmus von Reyes-Sierra und Coello, 2006. Als erstes wird ein zufälliger Schwarm von Partikeln erzeugt. Anschließend werden alle Partikel dieses Schwarms evaluiert. Im Falle von mehreren zu optimierenden Zielen werden die Pareto-dominanten Partikel bestimmt. Dies sind Partikel für die es kein anderes Partikel im Schwarm gibt, das hinsichtlich aller Ziele besser bewertet wurde. Anschließend wird das Bewegen des Schwarms durch den Problemraum bis zum Erreichen eines Abbruchkriteriums iterativ simuliert. Dabei wird in jedem Durchlauf jedes Partikel mit einem Geschwindigkeitsvektor von seiner bisherigen Position im Problemraum etwas weiter in eine - aus Sicht des Partikels - lohnenswerte Richtung bewegt - Gleichung (2a).

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (2a)$$

$$\vec{v}_i(t) = w\vec{v}_i(t-1) + c_1r_1(\vec{x}_{p,best} - \vec{x}_i(t)) + c_2r_2(\vec{x}_{g,best} - \vec{x}_i(t)) \quad (2b)$$

Der Geschwindigkeitsvektor v_i wird für jedes Partikel in jedem Bewegungsschritt gemäß Gleichung (2b) neu bestimmt. Dabei werden verschiedene Einflüsse des einzelnen Individuums und

des gesamten Schwarms berücksichtigt. Der erste Term simuliert eine Trägheit des Partikels, die dafür sorgt, dass das Partikel nicht sofort von seiner bisher lohnenswerten Bewegungsrichtung abgebracht wird. Der zweite Term lässt das Partikel zu einem gewissen Grad an seiner bisher persönlich besten besuchten Position im Problemraum $x_{p,best}$ festhalten. Der letzte Term stellt den Einfluss der bisher jemals besten vom Schwarm gefundenen Position dar. Das Partikel wird so langsam versuchen, auf seinem Weg durch den Problemraum, der bisher global besten Position näher zu kommen. Um den Einfluss der persönlich und global besten Position nicht statisch vorzugeben und dem Partikel somit zu ermöglichen auch abseits der direkten Richtung nach einer neuen persönlichen oder global besten Position zu suchen, werden diese Einflüsse mit einem zufälligen Faktor zwischen $0.5 \geq r_i \geq 1.0$ skaliert. Die Stärke der drei Einflüsse wird über die Parameter w , c_1 und c_2 global festgelegt. Clerc und Kennedy, 2002 konnten beweisen, dass bei einer Wahl von $c_1 + c_2 = 4$ der PSO-Algorithmus immer konvergiert. Daher werden $c_1 = 2$ und $c_2 = 2$ festgelegt. Der Trägheitseinflussparameter wird zu $w = 0.5$ definiert.

Für eine Mehrzieloptimierung muss bei jeder Geschwindigkeitsberechnung zunächst ein nicht-pareto-dominiertes Partikel ausgewählt werden, das für das Partikel als stellvertretend global beste Position ausgewählt wird. Da in dieser Arbeit jedoch nur eine Einzieloptimierung durchgeführt wird, wird an dieser Stelle für detailliertere Informationen auf Reyes-Sierra und Coello, 2006 verwiesen.

```

Partikel initialisieren;
Partikel evaluieren;
Bestimme Liste der nicht-dominierten Partikel;
it = 0;
while it < MaxIt do
    for Partikel do
        Bestimme global bestes Partikel aus der Liste der nicht-dominierten Partikel;
        Position, Geschwindigkeit und neue Lösungen der Partikel updaten;
    end
    Liste der nicht-dominierten Partikel updaten;
    it++;
end

```

Algorithm 1: Mehrziel-Partikelschwarmoptimierung

4 Anwendungsbeispiele

In diesem Abschnitt werden drei Beispielgebiete für die Optimierung definiert - vgl. Abbildung 5. Das erste Gebiet ist das in den bisherigen Abschnitten verwendete Quadrat mit der Seitenlänge 10. Dieses Gebiet soll mit Hinsicht auf das Rastern des Algorithmus für ein sehr einfaches Beispielgebiet stehen. Das zweite Gebiet ist ein rechtwinkliges gleichschenkliges Dreieck. Dieses Beispiel steht aufgrund seiner beiden 45° -Innenwinkel für ein bereits etwas komplexeres, jedoch noch konvexes Gebiet. Das dritte Beispielgebiet ist ein Quadrat dessen obere Kante jedoch halbiert wurde und der so entstandene Punkt auf den Mittelpunkt des Quadrates verschoben wurde. Dieses Beispiel ist somit nicht konvex und enthält neben zwei 45° - auch einen 270° -Innenwinkel. Dies soll im Rahmen dieser Arbeit stellvertretend für ein bereits etwas komplexeres Beispielgebiet stehen. Alle drei Gebiete haben eine Bounding-Box-Größe von 10×10 , um die Ergebnisse für den Radius r und die Randkantenlänge k besser vergleichen zu können. Für das Erzeugen eines Startschwarms während der PSO muss für jede Problemraumdimensi-

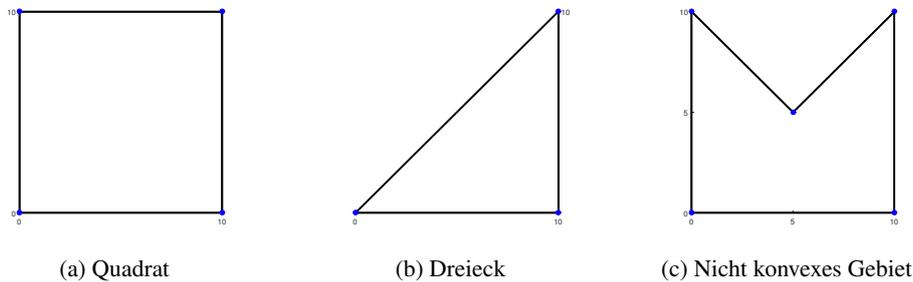


Abb. 5: Beispielgebiete

on, d.h. für jeden zu optimierenden Parameter, ein Wertebereich festgelegt werden. Gemäß den Resultaten der Untersuchungen in Abschnitt 2.3 wird für das Verhältnis v ein Bereich von $0.7 \leq v \leq 1.0$, für die Standardabweichung σ ein Bereich von $0.05 \leq \sigma \leq 0.8$ und für die Randkantenlänge k ein Bereich von $0.75 \leq k \leq 1.25$ definiert. Demnach ergibt sich aus dem Wertebereich von v und k für den Radius r ein Wertebereich von $0.525 \leq r \leq 1.25$.

4.1 Ergebnisse

In diesem Abschnitt wird zunächst das Ergebnis der Optimierung für die Beispielgebiete vorgestellt. Anschließend werden die Ergebnisse diskutiert und Empfehlungen für die Algorithmusparameter formuliert. Jede der drei Partikelschwarmoptimierungen ist mit einer Schwarmgröße von 24 Partikeln über 10 Iterationsschritte (Generationsen) durchgeführt worden. Insgesamt sind demnach 240 Positionen im Problemraum mit je 250 Wiederholungen evaluiert worden.

Abbildung 6 stellt den Verlauf der Entwicklung der bisher besten gefundenen Qualität aller evaluierten Algorithmusparameter-Sets dar. Im Verlauf der Optimierungen ist jeweils eine Steigerung über die Generationen zu erkennen. Tabelle 1 zeigt die Ergebnisse der Partikelschwarmoptimierung für die drei Beispiele aus Abbildung 5. Aufgrund der sich in einzelnen Optimierungen ergebenden optimalen Parameter am Rand ihres Wertebereichs, wurde eine weitere Optimierung für das Dreieck mit einem größeren Wertebereich durchgeführt - vgl. Tabelle 1, Beispiel b*. In Abbildung 7 ist für jede der drei ursprünglich als optimal identifizierten Parametereinstellungen eine Beispielgebietszerlegung abgebildet. Aus den drei Gebietszerlegungen

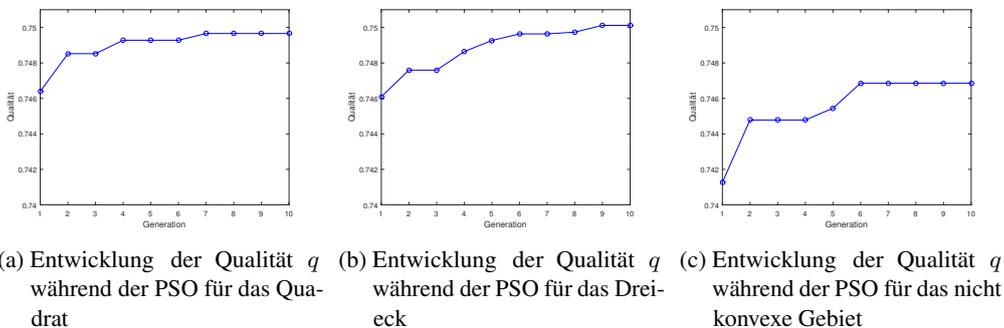


Abb. 6: Entwicklung der Qualitäten während der PSO

Beispiel	q	σ	k	v	r
a	0.7497	0.209	0.795	0.7	0.557
b	0.7501	0.199	0.75	0.7	0.525
c	0.726	0.261	0.75	0.726	0.545
b*	0.7529	0.1773	0.7356	0.6996	0.517

Tabelle 1: Ergebnisse der Optimierung

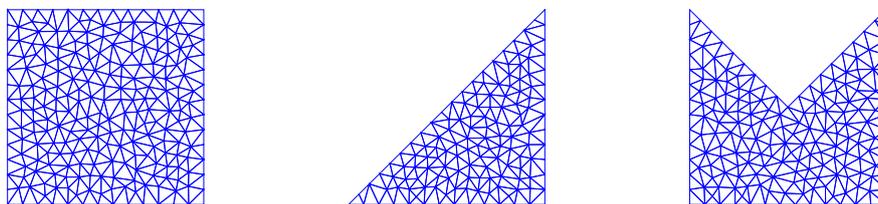
ergeben sich rein optisch keine Auffälligkeiten. Obwohl die als optimal ermittelten Parametereinstellungen teilweise am Rand des Wertebereichs einzelner Parameter liegen, konnte mit der offeneren Kontrolloptimierung gezeigt werden, dass die gefundenen Parameter in der Nähe des Optimums liegen sollten. Die Ergebnisse können demnach als plausibel eingestuft und im Folgenden diskutiert werden.

Ein Vergleich der Ergebnisse zeigt, dass die Standardabweichung für jede der drei Beispielgebiete offenbar entsprechend ihrer Komplexität anders gewählt werden sollte. Demnach sollte dem Algorithmus bei dem komplexeren dritten Beispielgebiet im Vergleich zu den anderen Beiden mit einer größeren Standardabweichung von $\sigma = 0.25$ mehr Abweichung zum Setzen der Punkte innerhalb eines Quadrates gewährt werden. Das Dreieck dagegen ist offensichtlich insgesamt bereits in einer vorteilhaften Grundgeometrie, für die der Algorithmus mit weniger Freiheit bei dem Setzen von Punkten bessere Gebietszerlegungen liefert. Das Verhältnis $v = \frac{r}{k}$ ist mit $v = 0.6996$ bis $v = 0.726$ etwas niedriger als erwartet - vgl. Abschnitt 2.3.

Insgesamt zeigt sich, dass der Algorithmus absolut gesehen mit dem komplexesten dritten Beispielgebiet durchschnittlich trotz der gefundenen optimalen Parameterkonfiguration nicht das Qualitätsniveau der beiden anderen einfacheren Gebietszerlegungen erreicht. Für komplexere Gebiete kann damit offensichtlich mit dem Algorithmus von Cavendish (1974) schwerer eine qualitativ hochwertige Gebietszerlegung mit dem in Abschnitt 2.2 vorgestellten Qualitätskriterium erreicht werden.

5 Fazit

Es wurden für den Algorithmus von Cavendish (1974) Parameter beschrieben und analysiert, die sich für eine Optimierung mittels der PSO eignen. Dabei wurde festgestellt, dass das Iden-



- (a) Beispiel einer Zerlegung des Quadrates mit optimalen Parametern
 (b) Beispiel einer Zerlegung des Dreiecks mit optimalen Parametern
 (c) Beispiel einer Zerlegung des nicht konvexen Gebietes mit optimalen Parametern

Abb. 7: Beispiele für Gebietszerlegungen der drei untersuchten Gebiete

tifizieren von optimalen Algorithmusparametern für das Erzeugen von im Durchschnitt sehr guten Gebietszerlegungen für unterschiedlich komplexe Gebiete schwieriger ist, als erwartet. Vor allem das in Abschnitt 2.3 aufgezeigte Gesamtverhalten des Algorithmus zeigt, dass die stochastischen Einflüsse in dem Algorithmus von Cavendish (1974) zu groß sind, um explizite Aussagen über die Optimierungsqualität zu postulieren. Demnach ist das Verwenden guter Algorithmusparameter zwar förderlich für den Erhalt guter Gebietszerlegungen, jedoch auf keinen Fall eine Garantie. Die ermittelten optimalen Algorithmusparameter können lediglich als eine erste Hilfestellung dienen. Die Frage nach konkreten Algorithmusparametern für bestimmte Arten von Gebieten ist noch nicht abschließend beantwortet. Dazu müssen zunächst weitere Beispielgebiete gewählt und optimiert werden und dessen Ergebnisse anschließend für eine Kategorisierung genutzt werden. Um das Identifizieren von optimalen Parameterkonfigurationen zu erleichtern sollte ebenfalls versucht werden, den Einfluss der stochastischen Effekte besser zu verstehen und gegebenenfalls im Sinne einer optimalen Gebietszerlegung zu nutzen.

Insgesamt ist festzuhalten, dass die Fluktuationseffekte beim Setzen der Steinerpunkte für die PSO schwierig zu kontrollieren sind, obwohl diesen stochastischen Einflüssen durch sehr häufiges Wiederholen stark entgegengesetzt wurde. Deshalb ist es offensichtlich sinnvoller, einen deterministischen Algorithmus in Betracht zu ziehen, der dennoch genügend Parameter zur Optimierung bereitstellt. Die Klasse der Quadtree-Algorithmen bieten eine erste Möglichkeit für die Analyse.

Literatur

- Bern, M. und Eppstein, D. (1992). „Mesh generation and optimal triangulation“. In: *Computing in Euclidean geometry* 1, S. 23–90.
- Brenner, S. und Scott, R. (1994). *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, New York.
- Cavendish, J. C. (1974). „Automatic triangulation of arbitrary planar domains for the finite element method“. In: *International Journal for Numerical Methods in Engineering* 8.4, S. 679–696.
- Clerc, M. und Kennedy, J. (2002). „The particle swarm-explosion, stability, and convergence in a multidimensional complex space“. In: *Evolutionary Computation, IEEE Transactions on* 6.1, S. 58–73.
- Delaunay, B. (1934). „Sur la sphere vide“. In: *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7.793-800, S. 1–2.
- Eberhart, R. C., Kennedy, J. et al. (1995). „A new optimizer using particle swarm theory“. In: *Proceedings of the sixth international symposium on micro machine and human science*. Bd. 1. New York, NY, S. 39–43.
- Knupp, P. M. (2001). „Algebraic Mesh Quality Metrics“. In: *SIAM Journal on Scientific Computing* 23.1, S. 193–218.
- Reyes-Sierra, M. und Coello, C. C. (2006). „Multi-objective particle swarm optimizers: A survey of the state-of-the-art“. In: *International journal of computational intelligence research* 2.3, S. 287–308.

Anwendung genetischer Algorithmen zur Kalibrierung Typologie-basierter energetischer Gebäudemodelle

Robert Irmeler

*Institut für Numerische Methoden und Informatik im Bauwesen, Technische Universität
Darmstadt, Franziska-Braun-Str. 7, 64287 Darmstadt, Deutschland
E-Mail: irmeler@iib.tu-darmstadt.de*

Abstract Das Energiekonzept der Bundesregierung sieht eine Verdopplung der Sanierungsrate im Gebäudesektor von 1 auf 2 % vor. Dies erfordert unter anderem eine Steigerung der Transparenz und Aussagekraft entsprechender Informationsinstrumente. Energetische Gebäudesimulationen können in diesem Kontext für Entscheidungsträger eine wichtige Grundlage zur Auswahl geeigneter Sanierungsmaßnahmen bilden. Die energetische Gebäudemodellierung stellt dabei einen oftmals zeitintensiven Prozess dar, welcher eine Vielzahl an detaillierten Nutzereingaben erfordert. Ansätze zur Vereinfachung und Automatisierung der Modellgenerierung stützen sich in der Regel auf Gebäudetypologien, um fehlende Nutzereingaben durch statistisch fundierte Annahmen zu ersetzen. Die Verwendung nicht-verifizierter Annahmen schränkt jedoch die Validität und Aussagekraft der resultierenden Simulationsergebnisse deutlich ein. Das vorliegende Paper stellt deshalb einen Ansatz zur Kalibrierung Typologie-basierter energetischer Gebäudemodelle auf Basis von Smart Meter Daten vor. Mit Hilfe eines genetischen Algorithmus soll dabei ein Set von Parametern ermittelt werden, welche das typologische Modell derart verändern, dass dieses das reale Gebäude hinreichend genau abbildet.

Keywords: Energetische Gebäudemodellierung, Gebäude-Typologie, Kalibrierung, Smart Meter Daten, Genetischer Algorithmus

1 Einleitung

Gemäß dem im September 2010 beschlossenen Energiekonzept sollen bis zum Jahr 2050 ein nahezu energieneutraler Gebäudebestand und eine Gesamtreduzierung des Primärenergieverbrauchs im Gebäudesektor von 80 % erreicht werden. Als wichtigste Maßnahme wird hierbei die energetische Sanierung des Gebäudebestands angesehen. Aus diesem Grund sieht das Energiekonzept eine Verdopplung der Sanierungsrate von 1 auf 2 % vor. Die momentanen Zahlen liegen jedoch deutlich hinter diesen Vorgaben zurück. Das von der Bundesstelle für Energieeffizienz in Auftrag gegebene aktuelle „Hintergrundpapier zur Energieeffizienzstrategie Gebäude“ schätzt die zukünftig erreichbaren Sanierungsraten auf 1,3 % für Ein- und Zweifamilienhäuser und 1,5 % für Mehrfamilienhäuser (Prognos, 2015). Für die Erreichung der Zielszenarien werden unter anderem Maßnahmen zur Steigerung der Sanierungsaktivität gefordert. Neben einer Aufstockung von Investitionsförderungen wird eine Verstärkung der Beratungsleistungen in Form eines „stringenten, qualitätsgesicherten und langfristigen Energieberatungs- und

Informationsangebots“ gefordert (Prognos, 2015). Hierfür müsse auch die Transparenz und Aussagekraft entsprechender Informationsinstrumente gesteigert werden. Energetische Gebäudesimulationen können in diesem Kontext für Eigentümer und Entscheidungsträger eine wichtige Grundlage zur Auswahl geeigneter Sanierungsmaßnahmen und der Abschätzung ihrer energetischen Auswirkungen und Wirtschaftlichkeit darstellen.

Bauphysikalische Simulationen simulieren den Energieverbrauch eines Gebäudes auf Basis von Wärmebilanz- und Transfergleichungen. Hierfür wird ein detailliertes Modell des zu untersuchenden Gebäudes benötigt, welches üblicherweise mit Hilfe spezieller Software wie bspw. eQuest, WUFI, EnergyPlus oder TRNSYS erstellt wird. Zur Erstellung eines bauphysikalischen Modells werden umfangreiche Informationen, u.a. über die geometrischen und thermischen Eigenschaften des Gebäudes, benötigt. Aus Sicht eines Hauseigentümers können derartige Informationen jedoch u.U. schwer zu beschaffen sein, z.B. wenn Planungsunterlagen veraltet oder nicht vollständig vorhanden sind. Weiterhin erfordert die korrekte Modellierung Erfahrung und ein tiefes Verständnis der eingesetzten Software. Aus diesem Grund wurden bereits verschiedene Ansätze entwickelt, um den Modellierungsprozess zu vereinfachen und die Menge an Nutzereingaben zu reduzieren. Diese Ansätze verwenden in der Regel Gebäudetypologien bzw. Prototypen ausgewählter Gebäudeklassen, um fehlende Nutzereingaben durch sinnvolle Default-Werte zu ersetzen (Kalogirou et al., 1997; Parekh, 2005; Rhodes et al., 2015). Weiterhin gibt es Ansätze zur Nutzung geografischer Informationsmodelle als Datengrundlage für energetische Analysen auf Stadt- oder Quartiersebene (Nouvel et al., 2013). Untersuchungen haben gezeigt, dass derartige Ansätze teilweise gute Ergebnisse erzielen können, jedoch führten nicht-verifizierte Default-Annahmen und zu stark vereinfachte Modelle auch oft zu signifikanten Abweichungen des simulierten Verbrauchs.

Das vorliegende Paper präsentiert deshalb einen Ansatz zur Erstellung und Kalibrierung energetischer Gebäudemodelle für den deutschen Wohnsektor. Ziel des Ansatzes ist es, ein bauphysikalisches Modell eines realen Gebäudes auf Basis minimaler Nutzereingaben zu generieren, welches im Rahmen eines Informationstools zur Simulation von Sanierungsmaßnahmen genutzt werden kann. Der Modellierungsprozess soll dabei weitgehend automatisiert erfolgen und kann grob in zwei Schritte unterteilt werden. Im ersten Schritt wird ein typologisches Modell erzeugt, wobei in Anlehnung an die zuvor beschriebenen Modellierungsansätze, u.a. geografische Informationsmodelle, sowie Gebäude- und Bauteiltypologien als Datenquellen herangezogen werden. Im zweiten Schritt erfolgt die Kalibrierung der typologischen Default-Werte mit Hilfe von Smart Meter Daten und einem genetischen Algorithmus. Ziel der Kalibrierung ist es, das reale Gebäude trotz minimaler Nutzereingaben so genau wie möglich abzubilden.

Der vorliegende Beitrag umfasst 5 Abschnitte einschließlich der Einleitung. Abschnitt 2 gibt einen kurzen Überblick über wesentliche Forschungsarbeiten zu Modellen und Methoden für die energetische Gebäudesimulation auf Basis reduzierter Nutzereingaben. Die verschiedenen Teilaspekte des Konzepts zur Modellgenerierung werden in Abschnitt 3 kurz vorgestellt. In Abschnitt 4 werden der Ansatz zur Kalibrierung genauer beschrieben, sowie erste Ergebnisse präsentiert. Schließlich werden die gewonnenen Erkenntnisse in Abschnitt 5 diskutiert und das weitere Vorgehen skizziert.

2 Modelle und Methoden für die energetische Gebäudemodellierung auf Basis reduzierter Nutzereingaben

Es existiert eine Reihe von Ansätzen zur Modellierung und Simulation von Gebäudeenergieverbräuchen auf Basis reduzierter Nutzereingaben. Im Folgenden wird versucht die Bandbreite

dieser Ansätze anhand einiger wesentlicher Beispiele abzubilden.

Kalogirou et al. (1997) trainierten ein künstliches neuronales Netzwerk um den Wärmebedarf eines Gebäudes basierend auf einzelnen Räumen zu berechnen. Die Autoren wählten 5 Eingabeparameter um Informationen zu Fensterflächen, Außenwandflächen, Partitions- und Geschossfläche, sowie der Design-Temperatur eines Raums zu erfassen. Weitere 3 Eingabeparameter repräsentierten Type-Codes zur Abbildung unterschiedlicher thermischer Bauteilqualitäten für Fenster, Wände und Decken. Die Klassifizierung der thermischen Bauteilqualitäten kann hierbei bereits als Definition einer einfachen Typologie verstanden werden. Das Netzwerk wurde anhand von 255 Datensätzen trainiert und zeigte einen maximalen Fehler von 10 % auf den verwendeten Testdatensätzen. Die Untersuchungen zeigen, dass die Verwendung typologischer Parameter zu akzeptablen Ergebnissen führen kann. Für die Simulation der gesamten Bandbreite an möglichen Sanierungsmaßnahmen wären jedoch zunächst ein deutlich größeres Trainingsdatenset, sowie weitere Eingabeparameter erforderlich. Und selbst dann wäre die Simulation des Einsatzes neuer Technologien nicht möglich, da hierfür keine oder zu wenige Trainingsdaten existieren würden.

Angesichts der Vielfalt an möglichen Sanierungsmaßnahmen erscheint die Verwendung bauphysikalischer Modelle am geeignetsten, da sie sowohl große Flexibilität in der Modellkonfiguration bieten, als auch die Definition neuer Komponenten erlauben. Um die Menge der hierfür erforderlichen Eingabedaten zu reduzieren wird oftmals auf Gebäudetypologien zurückgegriffen. Gebäudetypologien bieten geeignete Default-Werte für geometrische und thermische Charakteristiken eines Gebäudes. Die typologischen Daten umfassen dabei üblicherweise Informationen zu charakteristischen Abmessungen, Materialeigenschaften und Aufbau von Bauteilen, sowie vorherrschender Anlagentechnik. Eine Gebäudetypologie basiert in der Regel auf einer detaillierten Analyse des Gebäudebestands und bietet eine gewisse Form von Klassifizierung in Bezug auf z.B. Baujahr, Gebäudetyp und Region.

Parekh (2005) beschreibt die Entwicklung einer Gebäudetypologie für den kanadischen Wohngebäudesektor. Auf Basis einer Analyse und Korrelation unterschiedlicher statistischer Erhebungen wurden verschiedene Gebäudeprototypen in Bezug auf geometrische, thermische und Betriebscharakteristiken definiert. Die geometrischen Charakteristiken unterschieden hierbei verschiedene Wohnformen, Grundrisstypen, die Stockwerksanzahl, sowie die Art des unteren und oberen Gebäudeabschlusses. Die thermischen Charakteristiken umfassten Eigenschaften der Gebäudehülle, Art der Heiz- und Warmwasserbereitung, sowie Parameter zur Beschreibung der Luftdichtheit und Belüftung. Auf Basis eines Datensets detaillierter Verbrauchsmessungen mehrerer Gebäude wurden weiterhin charakteristische Betriebsprofile definiert. Diese Profile umfassten Annahmen zum elektrischen Verbrauch für Beleuchtung und Haushaltsgeräte, Aufenthaltszeiten der Bewohner, Thermostateinstellungen und typische Warmwasserverbräuche. Die Erzeugung eines Gebäudeprototyps konnte somit schließlich auf Basis der folgenden Eingaben erfolgen: Ort, Baujahr (bzw. Jahr der letzten Sanierung), Wohnform, Grundrissfläche, Grundrissform, Stockwerkszahl, Gründungs- und Dachtyp. Der Autor kommt zu dem Schluss, dass typologische Daten den Eingabeprozess vereinfachen können und als Basis für schnelle Energieverbrauchsanalysen dienen können. Er weist jedoch auch darauf hin, dass unverifizierte Annahmen zu nicht vertretbaren Abweichungen der Verbrauchsabschätzung führen können. Dies betraf vor allem Gebäude, welche bereits umfassenden Renovierungsmaßnahmen unterzogen worden waren. Um diese Abweichungen zu vermeiden schlägt der Autor vor, Angaben zu Gebäudehülle und Heizsystemen zu überprüfen, sowie die Anzahl der Default-Werte grundsätzlich zu begrenzen.

Ein Ansatz zur Begrenzung der Menge an Default-Werten kann die Nutzung von vorhandenen Geodatenmodellen sein. Je nach Datenmodell können so geometrische Informationen über

die Gebäudehülle oder auch weitere semantische Daten für die Modellgenerierung herangezogen werden. Nouvel et al. (2013) entwickelten einen integrativen Prozess für Wärmebedarfsberechnungen auf Stadt- und Quartiersebene auf Basis von 3D CityGML Modellen. Die Autoren kommen zu dem Schluss, dass CityGML auf Grund seines räumlich-semantischen Datenmodells eine exzellente Datengrundlage für automatisierte Wärmebedarfsanalysen darstellen kann. Da CityGML Modelle in unterschiedlichen Detaillierungsgraden vorliegen können, kann die Genauigkeit der Simulationsergebnisse jedoch variieren. Die Ergebnisse zweier Testfälle zeigten mittlere Abweichungen von weniger als 10 % zwischen simuliertem und gemessenem Wärmeverbrauch, sofern die Datengrundlage detaillierte Informationen über thermische Gebäudeattribute beinhaltet. Waren keine Informationen über thermische Eigenschaften vorhanden, nutzten die Autoren Daten einer Gebäudetypologie um diese zu ergänzen. In diesen Fällen betragen die Abweichungen rund 20 %.

Neben den beschriebenen Ansätzen existieren auch verschiedene Softwarelösungen, welche eine vereinfachte Generierung energetischer Gebäudemodelle unterstützen. Ein Beispiel ist die vom U.S. National Renewable Energy Laboratory (NREL) entwickelte Software BEoptTM (Building Energy Optimization). Die Anwendung erlaubt die Modellierung von Ein- und Mehrfamilienhäusern auf Basis eines simplen Geometriegenerators. Weitere Parameter zur Beschreibung der Gebäudehülle, Geräten, etc. werden mit Default-Werten belegt und können vom Nutzer innerhalb bestimmter Grenzen angepasst werden. BEopt zielt auf die einfache Analyse bestehender und geplanter Gebäude in Bezug auf ein kostenoptimiertes Design und Energieeffizienzmaßnahmen. Rhodes et al. (2015) nutzten BEopt um den Energieverbrauch von 54 Wohngebäuden in Austin (Texas) zu berechnen. Um die Genauigkeit des Simulationstools zu bewerten, untersuchten die Autoren 4 verschiedene Varianten für Eingabedaten und verglichen die Simulationsergebnisse mit den tatsächlichen Verbrauchsmessungen. Das beste Ergebnis, ein aggregierter Gesamtfehler von 2,4 % konnte erreicht werden, wenn einige der Default-Werte auf Basis von Vorort-Erhebungen (Energy Audits) angepasst wurden. Für einzelne Gebäude traten mitunter aber auch Fehler von bis zu 30 % auf, speziell bei Gebäuden mit besonders geringem Verbrauch.

Anhand der beschriebenen Beispiele lässt sich feststellen, dass Typologie-basierte energetische Gebäudemodelle ein nützliches Werkzeug für Energiesimulationen auf Stadt- und Quartiersebene darstellen können und geeignet sind, vereinfachte Analysen durchzuführen. Allerdings wird auch deutlich, dass bei der Betrachtung einzelner Gebäude oftmals erhebliche Abweichungen in den simulierten Verbrauchsdaten auftreten, wenn die verwendeten Default-Annahmen zu stark von den realen Bedingungen abweichen. Häufige Fehlerquellen sind dabei nicht-verifizierte Daten zur Geometrie der Gebäudehülle und ihrer thermischen Eigenschaften, sowie ein unzureichend modelliertes Nutzerverhalten. Eine Verifizierung von Konstruktionseigenschaften kann durch den Nutzer oder durch Vor-Ort-Untersuchungen erfolgen. Beide Varianten stünden jedoch dem ursprünglichen Ziel, der Vereinfachung des Modellierungsprozesses, entgegen. Demzufolge wird ein anderes Verfahren zur Anpassung typologischer Default-Werte benötigt. Auf Grund der in absehbarer Zeit steigenden Verbreitung von Smart Metern, könnten die damit verfügbaren hochfrequenten Verbrauchsmessdaten neue Möglichkeiten der Kalibrierung typologischer Annahmen eröffnen.

3 Framework zur semi-automatisierten Generierung energetischer Gebäudemodelle

Im Folgenden wird ein Konzept zur Erstellung energetischer Gebäudemodelle vorgestellt, welches im Rahmen eines Web-basierten Informationstools zur Simulation von Sanierungsmaßnahmen eingesetzt werden soll. Ziel des Konzepts ist es den Modellierungsprozess weitgehend zu automatisieren und die erforderliche Menge an Nutzereingaben auf ein Minimum zu reduzieren. Fehlende Nutzereingaben werden dabei durch die Fusion verschiedener Datenquellen ergänzt. Hierzu zählen Daten geografischer Informationssysteme, Point-Cloud-Scans mobiler Endgeräte, sowie Bauteil- und Gebäudetypologien. Um die zuvor beschriebenen Schwächen bei der Verwendung typologischer Annahmen zu kompensieren schließt das Konzept weiterhin einen Prozess zur Kalibrierung typologischer Default-Werte auf Basis von Smart Meter Daten ein.

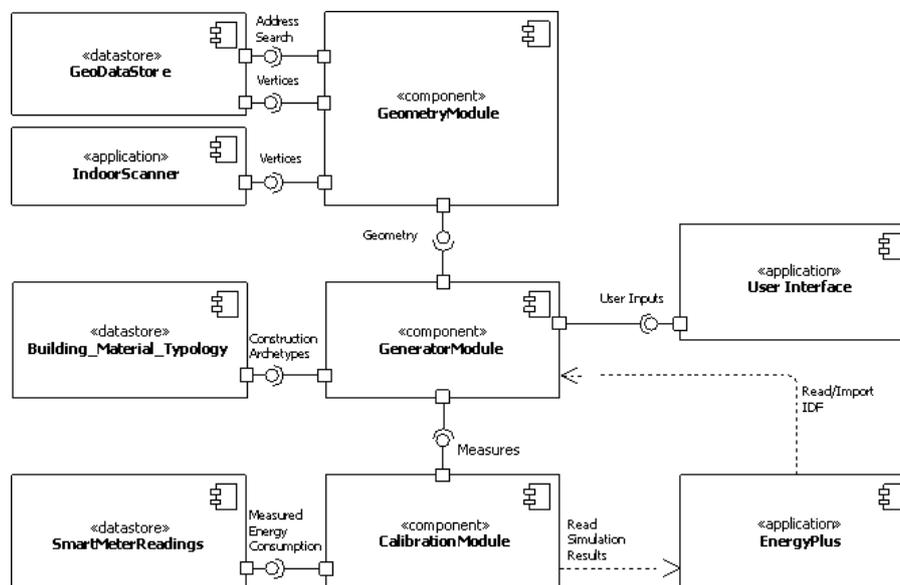


Abb. 1: Komponenten-Modell des Frameworks zur Modellgenerierung und Kalibrierung (Irmeler et al., 2016)

Das Framework umfasst drei wesentliche Komponenten: das Geometry Module, das Generator Module und das Calibration Module (siehe Abb. 1). Die Komponenten werden in den nachfolgenden Abschnitten genauer beschrieben. Als Simulations-Engine wird die weit verbreitete Open-Source Engine EnergyPlus des U.S. Department of Energy verwendet (vgl. Maile et al. (2007)). Um die Modellgenerierung weitgehend zu automatisieren kommt weiterhin die freie Middleware OpenStudio zum Einsatz, welche durch eine umfangreiche API den objekt-orientierten Zugriff auf das Gebäudemodell erlaubt.

3.1 Geometry Module

Das Geometry Module dient der Erfassung und Aufbereitung der Geometriedaten für das Generator Module.

Geometrie der Gebäudehülle

Die Geometrie der Gebäudehülle wird auf Basis von CityGML Modellen ermittelt. Seit 2014 ist der gesamte deutsche Gebäudebestand in CityGML modelliert. Der OGC Standard CityGML ist ein offenes räumlich-semantisches Modell zur Repräsentation von 3D Stadt- und Landschaftsmodellen (Gröger et al., 2012) und definiert 5 Levels of Detail (LoD). Der deutsche Gebäudebestand ist derzeit überwiegend in LoD1 modelliert. Der Übergang zu LoD2 ist jedoch mittelfristig geplant und einige Bundesländer haben den Prozess auch bereits abgeschlossen. Amtliche CityGML Modelle beinhalten eine GUID welche mit dem amtlichen Liegenschaftskataster (ALKIS) verknüpft ist. Dadurch kann das zu einer Gebäudeadresse korrespondierende CityGML Modell leicht ermittelt werden.

Je nachdem ob ein LoD1 oder ein LoD2 Modell eines Gebäudes vorliegt können verschiedene Informationen für die energetische Modellegenerierung genutzt werden. Im Fall eines LoD2 Modells können der Grundriss, äußere Wandflächen, sowie die Dachform ermittelt werden. Im Fall eines LoD1 Modells können der Grundriss und Informationen über die Höhe des Gebäudes erfasst werden. Hierbei ist anzumerken, dass die Genauigkeit der Höheninformationen für LoD1 Modelle oftmals nur 5 Meter betrifft, was diese Modelle in der Regel als Datenbasis ungeeignet macht. Demzufolge konzentriert sich das hier vorgestellte Framework auf CityGML Modelle des LoD2.

Da EnergyPlus den Import von CityGML Modellen nicht nativ unterstützt, wurde eine eigene Importfunktion entwickelt, welche die geometrischen Informationen aus LoD2 Modellen in das IDF (Input Definition File) Format von EnergyPlus überführt. Da viele semantische Attribute der amtlichen CityGML Modelle lediglich optional sind und oftmals nicht gepflegt werden, muss dabei jedoch bspw. die Anzahl der Geschosse auf Basis der Gebäudehöhe abgeschätzt werden.

Innere Geometrie und Zonen

EnergyPlus berechnet den Wärmebedarf eines Gebäudes basierend auf Temperaturdifferenzen und Übergangsbedingungen zwischen verschiedenen thermischen Zonen bzw. Räumen. Zwar ist es grundsätzlich möglich lediglich eine einzige thermische Zone zu definieren, jedoch erlaubt die Definition mehrerer Zonen die genauere Simulation der Heizlastverteilung innerhalb des Gebäudes. Mittelfristig soll das vorgestellte Framework deshalb auch die Einbindung und Aufbereitung von Point-Cloud-Scans mobiler Endgeräte umfassen. Franz und Rüppel (2015) stellten einen semi-automatisierten Prozess zur Generierung von IFC-Modellen auf Basis von Tiefensensordaten des Google Project Tango vor. Da derartige Technologien nach und nach alltagstauglicher werden (Smartphones), kann dieser Prozess weiterentwickelt werden, um so die Geometriedaten der CityGML Modelle auf einfache Weise um Informationen über die innere Raumaufteilung eines Gebäudes zu erweitern. Alternativ werden derzeit für jedes Stockwerk standardmäßig zwei thermische Zonen definiert.

3.2 Generator Module

Aufgabe des Generator Modules ist es die aus den verschiedenen Datenquellen gewonnenen zu einem vollständigen energetischen Gebäudemodell zusammenzufügen. Auf Basis der Nutzereingaben und unter Zuhilfenahme von Bauteil- und Gebäudetypologien ergänzt das Generator Module die durch das Geometry Module bereitgestellten Geometriedaten um Informationen

zu: Öffnungen (Fenster und Türen), Bauteilaufbauten und Materialien, Heizsystem, Kalt- und Warmwasserkreisläufe, Geräte und deren Betriebszeiten, sowie Aufenthaltszeiten von Bewohnern.

Bauteilaufbauten und Materialien

Das vorgestellte Framework nutzt zwei Typologien um fehlende Informationen zu Bauteilaufbauten und Materialien zu ergänzen. Dies geschieht auf Basis der Nutzereingaben zu Baujahr und Gebäudetyp (Einfamilienhaus oder Reihenhaus).

Die erste verwendete Typologie ist die Gebäudetypologie des Instituts für Wohnen und Umwelt (IWU). Die Typologie unterscheidet 12 Baualtersklassen, 4 Basistypen, sowie eine Reihe von Spezialfällen. Diese Klassifizierung resultiert in einer Gesamtzahl von 50 charakteristischen Gebäude-Prototypen (Loga et al., 2015). Für jeden Prototyp stehen verschiedene Informationen zu charakteristischen Eigenschaften zur Verfügung, wie bspw. typische Wohnfläche, Dachform, Fensterflächen und typische U-Werte für verschiedene Bauteile. Da die IWU-Typologie keine genaue Beschreibung der Bauteilaufbauten beinhaltet, diese Informationen jedoch speziell für die Simulation von Sanierungsmaßnahmen relevant sind, wird eine weitere Typologie als Datengrundlage herangezogen. Das Deutsche Zentrum für Umweltbewusstes Bauen e.V. (ZUB) entwickelte einen Katalog typischer Materialien im Gebäudebestand und eine Ableitung typischer Bauteilaufbauten mit Bezug auf die Baualtersklasse eines Gebäudes (ZUB, 2009).

Auf Basis dieser Typologien wurden sogenannte Construction-Sets für EnergyPlus Modelle definiert, welche einem zu generierenden Modell und seinen Bauteilen zugewiesen werden können.

Öffnungen

Da die verfügbaren CityGML Modelle keine Informationen über Gebäudeöffnungen beinhalten, werden diese zunächst anhand der Daten der Gebäudetypologie abgeschätzt. Die IWU-Typologie beinhaltet Angaben zu charakteristischen Fensterflächen für verschiedene Gebäudeklassen. Hieraus werden Fensterflächenanteile für Nord-, Ost-, Süd- und Westwände abgeleitet und anschließend die Fensterflächen entsprechend der tatsächlich vorhandenen Gebäudefläche skaliert und dem Modell hinzugefügt.

Heizsystem und Wasserkreisläufe

In Abhängigkeit der Nutzereingaben zum Typ des Heizungssystems, sowie der Art der Warmwasserbereitung werden Standard-Heizsysteme und Wasserkreisläufe generiert.

Gerätebetrieb und Aufenthaltszeiten

Der Fokus der Modellierung liegt momentan auf dem Wärmeverbrauch, weshalb der Stromverbrauch derzeit vereinfacht über Standardlastprofile für Sommer, Winter und Übergangszeiten abgebildet wird. Die Aufenthaltszeiten der Bewohner werden auf Basis einer Zeitverwendungserhebung (Destatis, 2015) modelliert und entsprechend der durch den Nutzer eingegebenen Anzahl der Bewohner skaliert.

3.3 Calibration Module

Das durch das Generator Modul erstellte Gebäudemodell beinhaltet noch zahlreiche Annahmen auf Basis der Bauteil- und Gebäudetypologien. Dies betrifft in erster Linie Fenstergrößen und thermische Eigenschaften der Gebäudehülle. Da anzunehmen ist, dass einige Eigenschaften des untersuchten Gebäudes von diesen Annahmen abweichen, ist es die Aufgabe des Calibration Module die Default-Werte zu kalibrieren. Der hierfür entwickelte Prozess wird im folgenden Abschnitt genauer beschrieben.

4 Kalibrierung typologischer Annahmen mittels Smart Meter Daten und einem genetischen Algorithmus

Der vorgestellte Prozess der Modellgenerierung resultiert zunächst in einem Gebäudemodell dessen Gebäudehülle geometrisch auf Grund des verwendeten CityGML Modells weitgehend dem realen Gebäude entspricht. Die Bauteilaufbauten, sowie die Fensterflächen und deren Verteilung auf der Gebäudehülle basieren jedoch auf typologischen Annahmen. Es ist nun anzunehmen, dass diese Default-Werte zwar eine Gebäudeklasse statistisch korrekt charakterisieren, für ein einzelnes Gebäude jedoch nicht vollständig zutreffen. Abweichungen könnten beispielsweise aus zwischenzeitlich durchgeführten Sanierungsmaßnahmen o.Ä. resultieren (z.B. Fensteraustausch). Dem im Folgenden beschriebenen Kalibrierungsprozess liegt nun die Annahme zugrunde, dass sich einige dieser Abweichungen durch einen Abgleich des simulierten Wärmeverbrauchs mit realen Messdaten anpassen lassen. Abbildung 2 verdeutlicht das Grundprinzip.

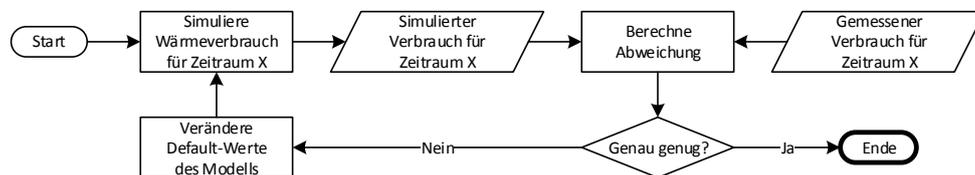


Abb. 2: Iterativer Prozess zur Kalibrierung typologischer Annahmen

4.1 Measures zur Anpassung von Modelleigenschaften

Um den Kalibrierungsprozess zu automatisieren müssen zunächst Möglichkeiten geschaffen werden einzelne Eigenschaften des Modells programmatisch zu verändern. OpenStudio bietet hierfür die Möglichkeit der Definition von sogenannten „Measures“. Dies sind parametrierbare Sets von programmatischen Anweisungen, welche Änderungen am Gebäudemodell vornehmen. Es ist somit beispielsweise möglich, ein Measure zu definieren, welches in Abhängigkeit eines übergebenen U-Werts den Aufbau aller Außenwände eines Modells derart verändert, dass der gewünschte Wert erreicht wird. Im Extremfall könnte somit für jeden nicht-verifizierten Aspekt des Gebäudemodells ein eigenes Measure definiert werden. Tabelle 1 gibt einige Beispiele für Measures welche im Rahmen des Kalibrierungsprozesses zum Einsatz kommen können.

Tabelle 1: Beispiele für Measures zur Anpassung nicht-verifizierter Default-Werte

Name	Parameter	Beschreibung
M_{EWALL}	U_{EWALL}	Wähle das Construction-Set, das dem übergebenen U-Wert am nächsten ist und weise es allen Außenwänden zu.
M_{ROOF}	U_{ROOF}	Wähle das Construction-Set, das dem übergebenen U-Wert am nächsten ist und weise es der Dachkonstruktion zu.
M_{WIN}	U_{WIN}, A_{WIN}	Wähle das Construction-Set, das dem übergebenen U-Wert am nächsten ist und weise es allen Fenstern zu. Ändere die Fensterflächen gleichmäßig über alle Wände, sodass die übergebene Fensterfläche erreicht wird.
M_{TEMP}	T_{SPACES}	Setze die Ziel-Temperatur aller beheizten Räume auf den übergebenen Wert.

4.2 Bestimmung der Measure-Parameter

Wird während des Kalibrierungsprozesses eine signifikante Abweichung des simulierten Verbrauchs von den realen Messwerten ermittelt, schließt sich die Frage an, welche Measure-Parameter gewählt werden sollten um das Modell anzupassen. Ein geeignetes Set an Measure-Parametern zu bestimmen stellt ein Optimierungsproblem dar.

Zur Lösung des Optimierungsproblems wird ein genetischer Algorithmus verwendet. Wie jeder andere genetische Algorithmus basiert dieser Algorithmus auf der Evolution einer Population von Individuen (siehe R. Haupt und S. Haupt (2004) und Mitchell (1999)). Jedes Individuum repräsentiert dabei ein Set von Measure-Parametern und somit eine Lösung für das Optimierungsproblem. Genetische Algorithmen nutzen Fitness-Funktionen um die Fitness eines jeden Individuums, und somit die Güte der jeweiligen Lösung, zu bewerten. Um zu bewerten wie gut ein bestimmtes Individuum dazu geeignet ist das Gebäudemodell zu kalibrieren, wird der Variationskoeffizient (CV: Coefficient of Variation) zwischen gemessenem und simuliertem Wärmeverbrauch bestimmt. Das Optimierungsproblem P kann somit wie folgt formuliert werden:

$$P : \min CV(f(X), m)$$

wobei m die gemessenen und $f(X)$ die simulierten Verbrauchswerte repräsentieren und $X = \{U_{EWALL}, U_{ROOF}, U_{WIN}, A_{WIN}, T_{SPACES}, \dots\}$ ein Set von Measure-Parametern beschreibt, d.h. das Genom eines Individuums.

Abbildung 3 gibt einen schematischen Überblick über den beschriebenen Prozess. Die Measure-Parameter der Start-Population werden basierend auf einer Normalverteilung um die Parameter des typologischen Basis-Modells zufällig ausgewählt. Das typologische Modell dient somit in erster Linie dazu, einen möglichst guten Startpunkt im Problemraum zu ermitteln.

Die Fitness-Evaluation für jedes Individuum besteht aus drei Teilschritten: Zunächst werden das Basis-Modell kopiert und die Measures mit den jeweiligen Parametern des Individuums angewendet. Anschließend erfolgt die Simulation des Wärmeverbrauchs. Schließlich wird der CV-Wert zwischen den simulierten und den gemessenen Verbrauchswerten bestimmt. Sowohl die simulierten als auch die gemessenen Verbrauchswerte sollten hierbei in einer zeitlichen Auflösung vorliegen, die es erlaubt charakteristische Verbrauchsverläufe hinreichend genau abzubilden. Im Rahmen der vorliegenden Untersuchungen wurden stündliche Verbrauchswerte betrachtet. Sofern der berechnete CV-Wert eine gewünschte Genauigkeit erreicht hat bricht das

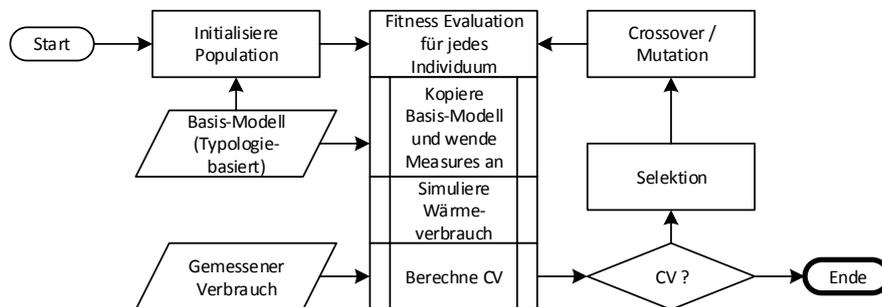


Abb. 3: Prozess des genetischen Algorithmus zur Kalibrierung typologischer Default-Werte

Verfahren ab. Andernfalls werden die fittesten Individuen ausgewählt und eine neue Population mittels üblicher genetischer Operatoren generiert.

4.3 Erste Untersuchungen

Um die Anwendbarkeit des beschriebenen Ansatzes zu überprüfen und notwendige weitere Anpassungen, speziell hinsichtlich der genetischen Operatoren und Selektionsmechanismen, zu identifizieren, wurde ein erstes vereinfachtes Testszenario implementiert. Da hierfür jedoch keine realen Messdaten eines Gebäudes vorlagen, ist die Aussagekraft dieser Ergebnisse entsprechend eingeschränkt. Als Testszenario diente ein in EnergyPlus modelliertes Einfamilienhaus. Dieses Modell wurde als Basis-Modell verwendet und repräsentiert in Hinblick auf den beschriebenen Kalibrierungsalgorithmus das typologische Modell, welches es zu verändern gilt. Im Anschluss wurden 3 Parameter des Modells signifikant verändert: die Dicke der Außenwanddämmung, die Dicke der Dachdämmung und der U-Wert der Fenster (siehe Tabelle 2). Der für das veränderte Modell simulierte Wärmeverbrauch substituiert im Rahmen dieser Untersuchung die „realen Messwerte“. Die für die Anpassung der Parameter verwendeten Measures wurden derart implementiert, dass die entsprechenden Parameter direkt verändert werden und kein vorheriger Abgleich mit der Bauteiltypologie erfolgt. Die Ergebnisse zweier Testläufe sind in Tabelle 2 dargestellt. Abbildung 4 zeigt die Fitness-Entwicklung der besten Individuen während des 2. Tests.

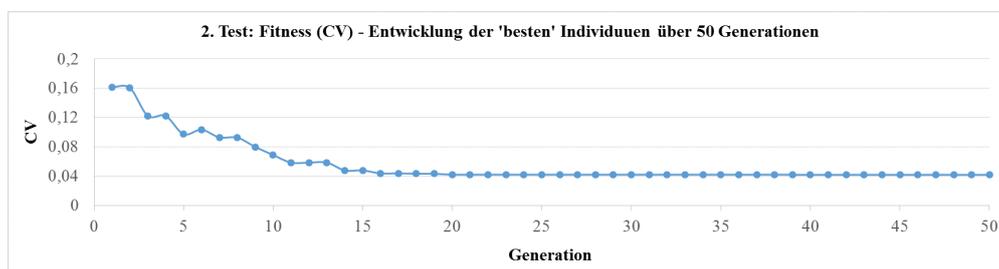


Abb. 4: Fitness(CV)-Entwicklung der 'besten' Individuen über 50 Generationen während des 2. Testlaufs

Die hier präsentierten Ergebnisse stellen lediglich den Anfang der Untersuchungen dar, bieten jedoch auch bereits erste Erkenntnisse. Die dargestellten Ergebnisse des 1. Tests zeigen 2 Individuen der letzten Generation. Es ist zu erkennen, dass in beiden Fällen die U-Werte

Tabelle 2: Ergebnisse zweier Tests zur Kalibrierung von Modellparametern

Modell/ Individuum (Nummer- Generation)	Dicke der Außenwand- dämmung [cm]	Dicke der Dachdämmung [cm]	U-Wert der Fenster [W/(m ² · K)]	CV (Fitness)
Basis-Modell	6	5	2,57	-
Ziel-Modell	12	20	1,60	-
1. Test; GA-Parameter: POPS: 64,; NGEN: 10; CXPB: 0,5; MUTPB: 0,7; INDPB: 0,4				
2-10	15,7	10,2	0,66	0,047
8-10	11,9	16,1	0,77	0,064
2. Test; GA-Parameter: POPS: 100; NGEN: 50; CXPB: 0,5; MUTPB: 0,5; INDPB: 0,4				
1-50	13,2	18,1	0,68	0,042

Anmerkung: POPS: Größe der Population, NGEN: Anzahl Generationen, CXPB: Wahrscheinlichkeit für Crossover, MUTPB: Wahrsch. für Mutation eines Individuums, INDPB: Wahrsch. für Mutation einzelner Gene

der Fenster deutlich zu günstig angenommen wurden. Dies ist ein Aspekt der sich vermutlich durch einen Abgleich mit der Bauteiltypologie positiv beeinflussen lassen könnte. Mittels eines solchen Abgleichs könnte sichergestellt werden, dass nur diskrete Werte, auf Basis tatsächlich gebräuchlicher Konstruktionen, zugewiesen werden. Es fällt weiterhin auf, dass das „fitteste“ Individuum, im Sinne des niedrigsten CV-Werts, nicht zwangsläufig auch am besten die Charakteristik des Ziel-Szenarios abbildet. So hat das Individuum 2-10 zwar den günstigsten CV-Wert erreicht, jedoch spiegelt das Genom des Individuums 8-10 die Charakteristik der Ziel-Parameter besser wieder. Abbildung 4 verdeutlicht weiterhin das recht schnelle Konvergieren der Individuen. Nach ca. 20 Generationen erfolgt keine wesentliche Anpassung mehr.

5 Diskussion und Ausblick

Im vorliegenden Beitrag wurde ein Konzept für die Generierung energetischer Gebäudemodelle auf Basis verschiedener Datenquellen und reduzierter Nutzereingaben vorgestellt. Weiterhin wurde ein Ansatz für einen automatischen Kalibrierungsprozess beschrieben, mit welchem typologische Default-Annahmen eines solchen Modells auf Basis von Smart Meter Daten angepasst werden könnten, um die realen Gebäudeeigenschaften besser abzubilden. Die Arbeiten hierzu stehen derzeit noch am Anfang, weshalb die hier präsentierten ersten Testuntersuchungen und die daraus erzielbaren Schlüsse entsprechenden Einschränkungen unterliegen. Die wesentlichste Einschränkung ist das Fehlen realer Messdaten. Dass das verwendete „Basis-Modell“ lediglich in 3 Parametern vom „Ziel-Modell“ abweicht, stellt eine starke Vereinfachung dar. Bei der Untersuchung eines realen Gebäudes und der Verwendung echter Verbrauchsmesswerte ist zu erwarten, dass die simulierten Verbrauchswerte noch weitaus deutlicher und unsystematischer von den realen Messwerten abweichen. Einen wesentlichen Einfluss stellt dabei das Nutzerverhalten dar, dessen Modellierung und Kalibrierung eine besondere Herausforderung darstellt. Dementsprechend liegt der Fokus der weiteren Arbeiten nun auf der Implementierung weiterer Measures und der Inklusion realer Beispieldaten. Weiterhin ist der hier präsentierte Algorithmus sehr rechenintensiv, da im Rahmen der Fitness-Evaluierung für jedes Individuum eine vollständige EnergyPlus-Simulation durchlaufen werden muss. Dieser Prozess ist jedoch auch hochgradig parallelisierbar, in dem Sinne, dass die Fitness-Evaluation aller Individuen

einer Generation parallel ablaufen könnte. Dies wird somit ebenfalls Gegenstand zukünftiger Arbeiten sein.

Literatur

- Destatis – Statistisches Bundesamt (2015). *Zeitverwendungserhebung – Aktivitäten in Stunden und Minuten für ausgewählte Personengruppen 2012/2013*. Wiesbaden.
- Franz, S. und Rüppel, U. (2015). „Supporting forensic fire cause analysis with next generation mobile devices by assembling a BIM based on multiple integrated sensor data“. In: *22nd EG-ICE International Workshop*. (Eindhoven, Netherland, 13.–15. Juli 2015).
- Gröger, G., Kolbe, T., Nagel, C. und Häfele, K. (2012). *OGC City Geography Markup Language (CityGML) Encoding Standard*. OGC Doc. No. OGC 12-019, Version 2.0.0.
- Haupt, R. und Haupt, S. (2004). *Practical Genetic Algorithms*. 2. Aufl. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Irmeler, R., Eller, C. und Rüppel, U. (2016). „A Next Generation Platform for Building Energy Modeling Based on Smart Metering Data, Point Cloud Scans, GIS and Building Typology“. In: *23rd EG-ICE International Workshop*. (Kraków, Polen, 29. Juni–1. Juli 2016).
- Kalogirou, S., Neocleous, C. und Schizas, C. (1997). „Heating load estimation using artificial neural networks“. In: *Proc. of CLIMA 2000 Conf.* (Brussels (Belgium)).
- Loga, T., Stein, B., Diefenbach, N. und Born, R. (2015). *Deutsche Wohngebäudetypologie*. Darmstadt.
- Maile, T., Fischer, M. und Bazjanac, V. (2007). „Building energy performance simulation tools-a life-cycle and interoperable perspective“. In: *CIFE Working Paper 107*. Stanford University.
- Mitchell, M. (1999). *An Introduction to Genetic Algorithms*. 2. Aufl. MIT Press.
- Nouvel, R., Schulte, C., Eicker, U., Pietruschka, D. und Coors, V. (2013). „CityGML-based 3D city model for energy diagnostics and urban energy policy support“. In: *Proceedings of the 13th Conference of International Building Performance Simulation Association*. (Chambéry, France, Aug. 2013), S. 218–225.
- Parekh, A. (2005). „Development of archetypes of building characteristics libraries for simplified energy use evaluation of houses“. In: *IBPSA, Ninth International Conference*. (Montreal, Canada, Aug. 2005), S. 921–928.
- Prognos (2015). *Hintergrundpapier zur Energieeffizienzstrategie Gebäude*. Berlin, Heidelberg, Darmstadt.
- Rhodes, J. D., Gorman, W. H., Upshaw, C. R. und Webber, M. E. (2015). „Using BEopt (EnergyPlus) with energy audits and surveys to predict actual residential energy usage“. In: *Energy and Buildings* 86, S. 808–816.
- Zentrum für Umweltbewusstes Bauen e.V. (ZUB) (2009). *Katalog regionaltypischer Materialien im Gebäudebestand mit Bezug auf die Baualtersklasse und Ableitung typischer Bauteilaufbauten*. 2. Version, 20.10.2009.

A Modular Routing Graph Generation Method for Pedestrian Simulation

Quirin Aumann¹ and Peter Kielar²

*Chair for Computational Modeling and Simulation, Technische Universität München,
Arcisstr. 21, 80333 München, Germany*

¹*E-Mail: quirin.aumann@tum.de,* ²*E-Mail: peter.kielar@tum.de*

Abstract Simulation of pedestrian movement is vital in many applications ranging from modeling pedestrian traffic movement in the vicinity of big events to predicting evacuation scenarios (Galea et al., 2008). These simulations are conducted based on different models that describe pedestrian behavior. Pedestrian navigation behavior is often simulated using graph-based approaches, so called routing graphs. Among the numerous models and strategies to generate and optimize graphs, only certain methods can be used to create a routing graph for pedestrian dynamics. However, an applicable routing graph can often only be generated by a combination of different graph manipulating algorithms. In the following, we will present a flexible framework to generate graphs suitable for pedestrian simulations based on the geometry of a given scenario. For this approach, we identified and separated parts of the graph generation methods and arranged them into standalone procedures, which can be combined in a modular way. Our approach includes four main operations for manipulating a graph: (1) creating vertices, (2) creating edges, (3) deleting vertices, and (4) deleting edges. As our strategy uncouples all independent steps of the graph generation, they can be combined in any logical order defined by a user. Furthermore, we provide predefined graph generation strategies that comprise a sequence of operations for well-known graph generation algorithms. Following our approach, it is possible to generate a graph, which can be adjusted to the given simulation requirements. Due to the flexible and modular structure of our approach, it is also possible to add new methods for graph manipulation and introduce them into already available strategies. Thus, the application of our graph generation framework helps researchers and practitioners to find the most suitable routing graph for different pedestrian simulation applications.

Keywords: Pedestrian Dynamics, Pedestrian Navigation, Routing Graph, Graph Generation

1 Introduction

Many mathematical problems and relations can be described using an abstract graph. Graph theory is a big field in mathematics. A rather intuitive field of application for graphs is the routing paths of people be they in a vehicle or on foot. A famous mathematical proof using graph theory is the problem of the seven bridges of Königsberg dating back to 18th century (Euler, 1741). The problem questions if it is possible to cross all seven bridges in the city of Königsberg in one walk while crossing each bridge exactly once.

The proof uses a graph which could be described as a routing graph. It features edges and vertices depicting accessible paths. Such a routing graph has also been used in pedestrian simulations, see Henderson (1974) or Hillier et al. (1993) for examples. As graph generation and processing are vast fields in mathematics, numerous methods and procedures exist to generate graph networks (Trudeau, 1993). Not all of them are suitable for routing graphs, as they must meet a few specific criteria. The approach proposed in the scope of this paper can be used to generate routing graphs for pedestrian simulations. Thus, we do not focus on pedestrian simulations models, but on graphs that are used in pedestrian behavior approaches.

The remainder of this paper is structured as follows. First, we will present different approaches to generate graphs which can be used for pedestrian simulation and a framework that allows to combine different approaches to generate a custom tailored routing graph to fit most simulation scenarios. Then, the implementation of the framework is addressed along with predefined graph generation strategies picked from the literature. We will conclude with a discussion why the proposed approach offers a high flexibility and simplifies the process of routing graph generation.

2 Related work

There are numerous ways to model pedestrian dynamics and movements to predict the behavior of large crowds. A popular approach is based on a three-tier architecture as proposed by Hoogendoorn and Bovy (2004) and Reynolds (1999). In this concept, three levels of pedestrian movement are defined: the strategic level, the tactical level, and the operational level (locomotion).

The strategic level describes how pedestrians find a next destination to visit. The tactical layer is used to describe way-finding of pedestrians; thus finding a route from the current position to the destination. Finally, the operational level depicts the actual movement of the pedestrian walking behavior (e.g. overtaking, accelerating, or stopping).

2.1 Graph-based navigation

The human navigation behavior can be described with the help of a routing or navigation graph containing the path a pedestrian takes between start point and destination (Höcker et al., 2010). For a larger scale city navigation, for example, this graph can be generated using the street network, as the movement of the pedestrians is limited to distinct routes using the available streets. For smaller scale scenarios where the directions of the movement are not limited, for example, in the open field or a bigger square in the city, such a graph can not be found intuitively.

The base for pedestrian routing are cognitive maps of the environment and spatial cognition, which described the spatial information processing of humans (Wolbers and Hegarty, 2010). These cognitive maps can be well represented by a navigation graph. However, a routing graph should depict such a cognitive map correctly and with an appropriate level of detail to gain reasonable results from the simulation (Andresen et al., 2016). If navigation graphs are used, the routing behavior of pedestrians can be predicted either iteratively, based on a sequence of decisions with only little knowledge of the overall system, or directly, where the simulated pedestrians are aware of their large scale surroundings (Kielar et al., 2015).

The direct methods typically use algorithms to compute the shortest path through the system using, for example, the well-known shortest path algorithm introduced by Dijkstra (1959). Dijkstra-based approaches assume that all pedestrians have a profound knowledge of the street network, so that pedestrians can find the shortest path from start to destination.

In contrary, way-finding models using iterative routing methods do not need to provide overall information about the system, they model routing behavior where the street network is not known to the pedestrians (Kielar et al., 2015). Such an algorithm finds the next navigation node based on local optima. This could be, for example, the angle of the adjacent streets based on the concept that pedestrians tend to walk along straight streets. Another approach can model the spatial knowledge of the system the pedestrians gained throughout the simulation and use this for the routing (Andresen et al., 2016).

For those navigation strategies a well-defined routing graph is of great importance. The graph is the basic part of pedestrian movement as it represents the cognitive map of the pedestrians and, therefore, the ability of the pedestrians to navigate their environment. Resting upon the routing graphs, different navigation patterns can be established but they all depend on the quality of the underlying graph.

2.2 Graph generation

Different approaches for routing graph generation based on the scenario geometry have been proposed for example by Geraerts and Overmars (2006), Kneidl et al. (2012), and Höcker et al. (2010). All those procedures generate suitable routing graphs, which can be used in pedestrian simulations. Those procedures can be reduced to a combination of four basic types of graph manipulation operations, i.e. edge creation and deletion as well as vertex creation and deletion. These operations can be used independently from each other, thus, are exchangeable and can also be combined to generate new graph generation strategies.

Before taking a close look at the graph generation processes, some mandatory conditions for routing graphs have to be elaborated. A routing graph consists out of vertices v_i , which are connected by edges e_{ij} . The graph is not directed, so it is possible for the pedestrians to traverse an edge in both directions, it follows $e_{ij} = e_{ji}$. All entities of the graph are created in the geometrical domain denoted as *scenario* in the following. The scenario holds information about a geometrical situation, for example a floor plan, and describes areas accessible by pedestrians. Non-accessible areas such as walls are called *obstacles*. Vertices cannot be created within those obstacles and the edges must not intersect any obstacle.

3 Modular Routing Graph Generation

In the following, we will present a highly flexible framework for routing graph generation that makes use of the interchangeability of the graph manipulation procedures. As stated above, a routing graph comprises vertices and edges, so we need strategies for the creation and manipulation of these entities. Thus, there are four main categories of graph manipulation used in the framework: vertex and edge creation as well as deletion of vertices and edges.

To make it possible to adapt the different algorithms to various geometrical situations, all strategies have parametrized conditions which affect their outcome. These parameters can easily be changed by the user to fit a specific simulation. In the following sections the main concept behind those strategies is outlined and examples are also provided.

3.1 Vertex creation

The first and also one of the crucial steps in generating a graph is finding suitable spots for vertices. The positions of the vertices determine where edges will be connected, so the vertex will ultimately serve as a corner or an intersection in the routing graph. Although it seems intuitively easy to find well-placed positions for the vertices in, e.g., a floor plan. However, it is not trivial to numerically generate vertices suitable to build up a routing graph, which can also depict realistic walking behavior of pedestrians (Brogan and Johnson, 2003).

The algorithms proposed below use two different approaches to find vertex positions in a given geometric domain. The first strategy, presented in Höcker et al. (2010) and Kneidl (2013), generates vertices at convex corners of the geometric boundaries. The second approach generates the medial axis of the geometric domain with the help of a Voronoi diagram to find suitable positions for vertices to be generated (Geraerts and Overmars, 2006).

Corner method

The corner method proposed in Höcker et al. (2010) scans the obstacle geometries to find suitable locations for vertices and creates them at the corners of the obstacle geometries. The user can define the distance between the created vertex and the corner of the obstacle. This parameter should be large enough that a congestion around the corner can be avoided. Figure 1 illustrates how the algorithm treats different geometric situations.

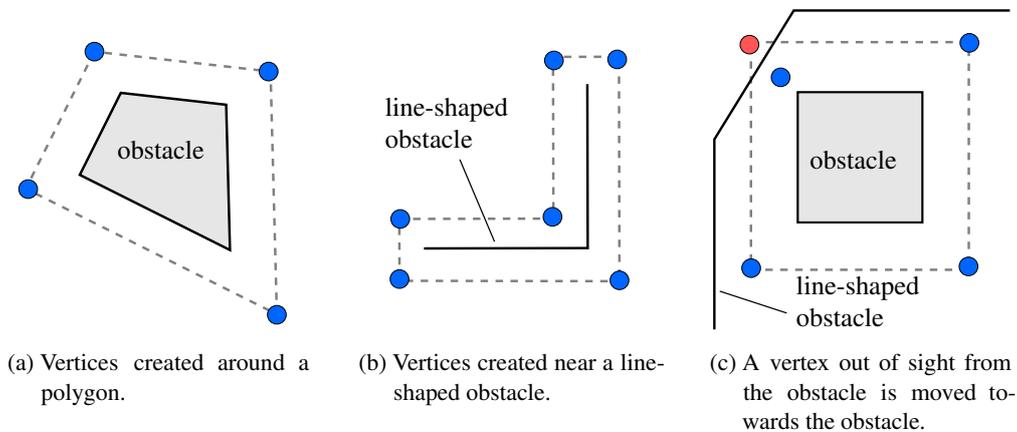


Fig. 1: The corner method illustrated. Actually generated vertices shown in blue, user-defined offset around the obstacle as dashed line.

Depending on the shape of the obstacle, the algorithm creates either one or two vertices at each corner. In the vicinity of polygon shaped obstacles, one vertex is created at the user-specified offset (see figure 1a). Near a line-shaped obstacle, two vertices are placed as shown in figure 1b. Before creating a vertex at a new location, the algorithm checks, if the vertex is visible from the current corner and if not, the vertex is moved towards the obstacle (figure 1c).

This approach creates vertices near the obstacle boundaries, depending on the offset distance specified by the user. Therefore, the method is especially useful for narrow geometric situations like hallways or offices. On the other hand, the algorithm does not populate a large open space (e.g. a square) with vertices.

Voronoi method

This approach finds spots suitable for vertex creation with the help of a Voronoi diagram generated from the underlying geometric domain. This kind of diagram comprises a set of sites p_i in a space M , which divide M into Voronoi cells R_i . Inside R_i the distance of every point to p_i is less or equal to any other site p_j . Aurenhammer et al. (2013) provides a thorough insight in the theory behind Voronoi diagrams. Figure 2 shows how the procedure extracts the locations for vertices from the Voronoi diagram.

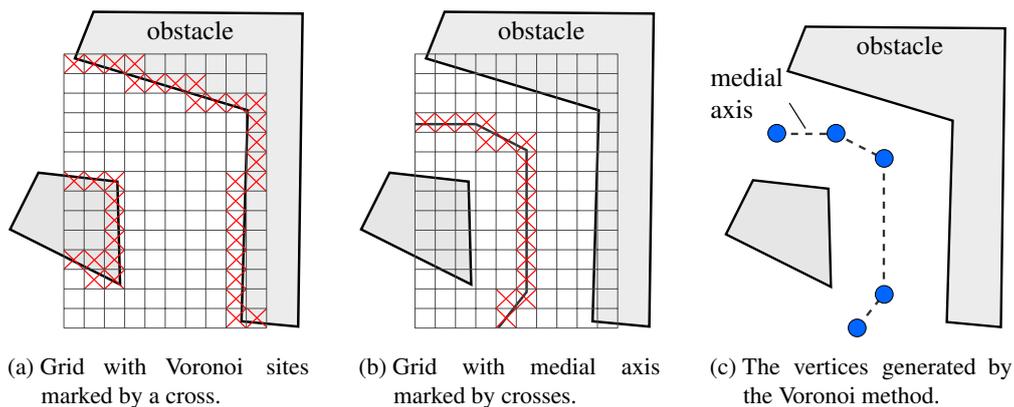


Fig. 2: The Voronoi method illustrated. Vertices shown in blue, marked grid cells in red.

Initially, the approach lays a grid over the geometric domain and marks all cells on the grid intersected by geometric boundaries as shown in figure 2a. The grid size is a parameter that has to be specified by the user and should be adapted to the specific geometric situation. The marked grid cells now serve as the sites p_i used in the Voronoi algorithm and the Voronoi polygons are created. Then, all edges of Voronoi polygons intersecting or lying inside obstacles are deleted so that the medial axis of the geometric domain is now represented by the remaining Voronoi edges. Once again, they are intersected with the grid and all intersected grid cells are marked (figure 2b). After applying thinning algorithms on the marked grid cells, the vertex positions can be extracted by selecting all grid cells with more than two marked neighbors. A vertex will be generated at this very spot in the geometrical domain as those cells represent the corners and intersections of the medial axis. Figure 2c shows the result of this algorithm.

Concluding, the Voronoi method produces vertices as far away as possible from the geometric boundaries. This is especially useful if big squares or open spaces shall be simulated and populated with people because the pedestrians will also walk in the middle of those spaces.

3.2 Edge creation

When at least a few vertices are present in the domain, the edges connecting the vertices can be generated. Before an edge is finally created, the provisional edge is intersected with every obstacle and geometric boundary. If an intersection is detected, this edge is not created, as the edge creation algorithms are based on the pairwise visibility of the vertices. As the edges only connect the given vertices, the positions of these vertices play crucial role in how realistic the routing graph can simulate pedestrian motion.

Visibility method

This procedure generates a full visibility graph which means that every vertex v_i is connected to all vertices v_j if the connecting edge e_{ij} does not intersect any obstacle or geometric boundary. This algorithm loops over all vertices and tries to connect it to every vertex present in the graph. A temporary edge is created between those vertices and a procedure is executed to determine whether the new edge is intersecting any obstacles. If yes, the temporary edge is discarded, otherwise it now connects the two vertices in the graph.

This method produces as much edges as possible that often results in a graph which is too large for efficient computations. It should, therefore, mainly be considered as a starting point for edge deletion algorithms.

Cone method

This approach is presented in Kneidl (2013) and avoids the generation of edges lying in direct vicinity to each other. After generating a new edge, the algorithm defines a cone-shaped area around this edge and does not create any more edges in this area. This avoids the generation of multiple edges that point in the same direction.

The algorithm starts by picking a vertex v_i from which a nearest-neighbor search is performed in order to find the closest in-sight vertex v_j . A new edge e_{ij} connecting these vertices is created and all vertices lying in a cone-shaped area around the new edge are excluded from further searches (see figure 3). The angle α of the cone is parametrized and has to be specified by the user. The next edge from vertex v_i is created connecting the next nearest neighbor which is not located inside the cone and a new cone is created around the new edge. The cones can overlap each other but inside every cone only one edge can be present.

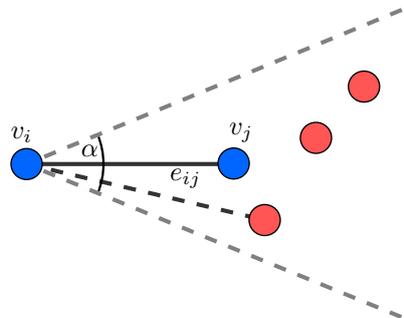


Fig. 3: The cone method illustrated. Vertices shown in red are excluded from the search after edge e_{ij} has been created.

The size of the user parameter α determines how many edges will be created in the system. A small α will create many edges while a large α generates less. This algorithm is efficient and easy to use, as it can be applied directly after a vertex creation step, and at the same time generates a reasonable amount of edges, if the parameter α is chosen properly.

Useful edges

Starting point for the algorithm is a connected but not fully-connected graph. A good foundation for this algorithm is, for example, a minimum spanning tree. So called useful edges are created following the procedure proposed by Geraerts and Overmars (2006) and added to the graph. A useful edge $e_{ij,u}$ is defined as an edge that directly connects two vertices v_i and v_j and is by a factor of K shorter than the path between v_i and v_j through the graph without the useful edge $e_{ij,u}$. The factor K has to be specified by the user and determines how many new edges are created. A high value of K results in less new edges being generated and vice versa.

The algorithm first checks if vertex v_j is visible from v_i under a certain tolerance specified by the user. If there is no obstacle between the two vertices, the length of the path through the graph from v_i to v_j is computed using a variant of the shortest path algorithm by Dijkstra (1959). Now let $L(i, j)$ be the distance between v_i to v_j through the graph and $D(i, j)$ the length of the direct connection. If $KD(i, j) < L(i, j)$ holds, the direct connection between v_i and v_j is considered being useful and the edge e_{ij} is created.

This procedure allows the creation of detours in the graph and lets the user precisely control how long the paths between specific vertices should be. A downside is, that the algorithm needs an already connected graph which often includes some additional computation steps in advance. However, the method can be used at the end of the overall graph creation process to reach a desired level of connectedness.

3.3 Vertex deletion

The vertex generation algorithms presented above are likely produce many vertices in direct vicinity to each other, especially near complex obstacles. Those vertices are sometimes not needed for a proper routing graph and should therefore be pruned. This reduces the computational effort for most of the edge creation algorithms as well as the memory consumption of the graph.

Simple vertex prune method

The algorithm loops over all vertices and checks their distance to all other vertices. The vertices v_j , which lie in a radius less than the user defined merge distance around the current vertex v_i will be grouped. A new vertex v_n is created at the centroid of the grouped vertices v_j . After the new vertex is created, the grouped vertices are deleted. This algorithm can only be used if no edges are present in the graph yet, as it does not consider how the connections of the vertices.

The simple vertex prune method is best to be used directly after vertex creation methods to clean their result and reduce the number of vertices in the geometric domain. When dealing with complex geometries, certain algorithms, like, for example, the Voronoi based vertex creator, often produce many vertices lying close to each other that can be pruned efficiently. The user has to keep in mind that the algorithm might produce sets of vertices that can not be connected on a visibility basis.

Extended vertex prune method

If the graph is already connected, a more sophisticated approach to merge the vertices can be used. This algorithm considers edges and the edges between the vertices and it assures, that the graph is still connected after vertex pruning.

The algorithm loops over all vertices v_i and checks the distance to the neighboring vertices v_j ; but in contrary to the simple method the extended algorithm tries to delete the vertices in direct vicinity rather than creating a new one. If the distance to a neighbor v_j is less than the user defined merge distance, the neighbor is considered for deletion. As the vertex to be deleted has already at least one adjacent edge e_{jk} , those edges have to be reconnected to the core vertex v_i . The reconnected edge e_{ik} must not intersect any geometric boundary or obstacle, so if e_{ik} would intersect an obstacle, the old edge will be kept and the vertex v_j will not be merged into v_i . Figure 4 illustrates the concept of the algorithm.

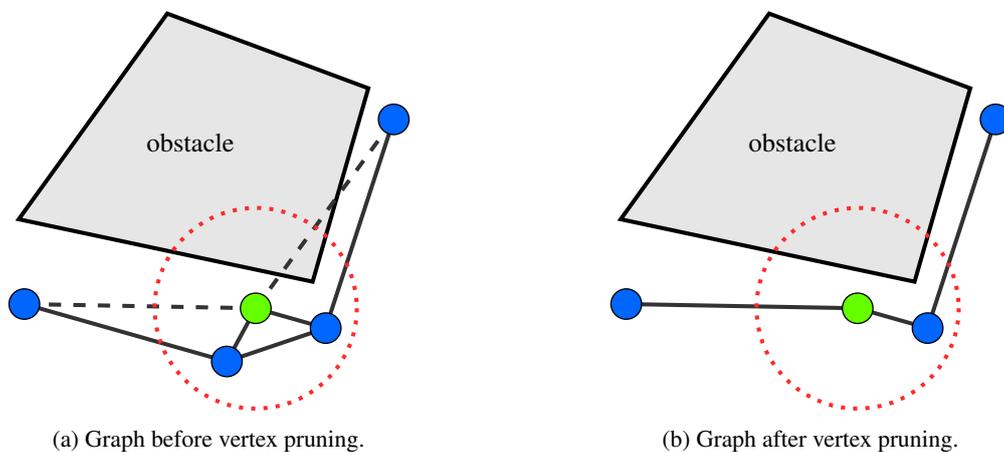


Fig. 4: The red dots illustrate the merge distance around the merge core (green). One of the vertices is not merged into the merge core as otherwise the connection in sight could not be maintained.

The extended vertex prune method can be used at any point in the graph generation process. If no edges are present in the graph the part of checking the visibility is simply ignored. The fact that the algorithm also takes care about the connectivity of the graph makes it a safe tool for pruning the graph in a later period of graph generation.

3.4 Edge deletion

Especially a fully connected visibility graph can have many edges which are not needed for a proper pedestrian simulation. Additionally, such a graph results in a high memory consumption and long computation times for the simulation (Kneidl et al., 2012). In the following, algorithms are presented that delete redundant edges. The deletion of edges can also include graph conversions into subgraphs like a minimum spanning trees.

Delete unreachable edges

After the initial graph generation, it is reasonable to delete all parts of the graph which can not be reached by a pedestrian during the simulation. Those unreachable sections of the graph lie, for example, outside the geometric boundary or in parts of the domain which are surrounded by obstacles. They are created because the vertex and edge creation algorithms only consider, if pedestrians can walk in an area but not, if the specific area can be accessed. This algorithm needs information about vertices that are entries and exits (e.g. doorways) in a pedestrian scenario.

As the unreachable parts of the graph are not connected to the entries or exits, the algorithm traverses along the graph edges beginning at the entry vertex and marks all visited graph entities. As soon as the traverse algorithm can no longer reach any unmarked graph entity, the traverse algorithm terminates. Now, all unmarked graph parts can be deleted as they cannot be reached during the simulation using the given entry.

This procedure does not take any user defined parameters and is most efficiently be used after the first edge creation step. Also the computational effort for following graph manipulation steps is reduced after the non-reachable graph parts have been deleted as they are no longer considered.

Minimum spanning tree

A minimum spanning tree (MST) is a sub-graph of a weighted graph that connects all vertices of the main graph with a minimal total edge weight (Graham and Hell, 1985). The MST itself cannot be used as a proper routing graph but rather as a starting point for further graph manipulations. For example the useful edges method presented in chapter 3.2 can be used on a minimum spanning tree.

The conversion of the fully connected visibility graph to a MST follows the algorithm of Kruskal (1956). This algorithm takes a fully connected visibility-graph as input and successively marks all unmarked edges with the minimum weight of all edges. Edges which would form a circle of marked edges are not marked. If no more edges can be marked as they would form a circle, the marked edges represent the minimum spanning tree and all others are deleted.

4 Implementation

4.1 The MomenTUMv2 framework

MomenTUMv2 is a computational framework for pedestrian behavior simulations (Kielar et al., 2016). In the framework, pedestrian behavior and non-behavior related models are conceptualized as implementation of theories via mathematical operations. This approach enabled developers to extend MomenTUMv2 with new theories simply by implementing given interfaces. This core feature is the basis of the graph generation framework. The graph operations are implemented as non-behavior models that provide operations during the pre-processing phase of simulations. During a simulation, the graph generation concept can be used in case the scenario changes, but may put additional burden on the processing time.

4.2 Predefined Graph Generation Strategies

In the following chapters, two pre-defined routing graph generation strategies implemented in the MomenTUMv2 framework will be described. According to these strategies, the framework applies the vertex and edge manipulation operations as described above. The strategies for routing graph generation presented in the following are taken from Geraerts and Overmars (2006) and Kneidl (2013). The methods are applied on the same example geometry and the results are displayed in figure 5.

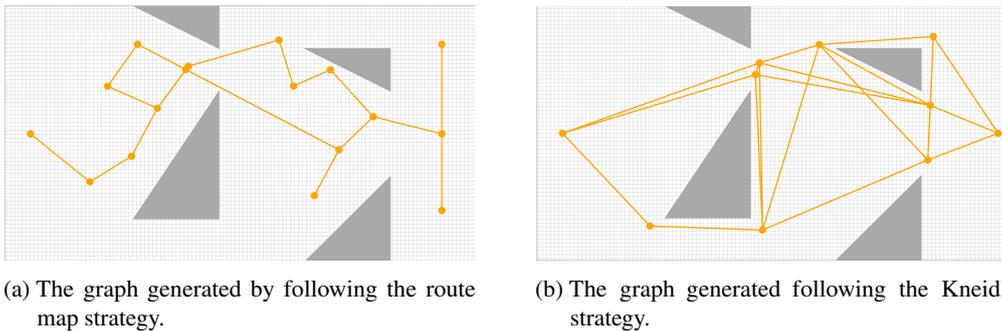


Fig. 5: The routing graphs generated by the two proposed generation strategies.

Routemap method

The following procedure is taken from Geraerts and Overmars (2006) and focuses on generating routing graphs that provide a high clearance to the geometric boundaries. In our implementation this is achieved using Voronoi diagrams. The procedure uses the following sequence of graph manipulation strategies and the result is visualized in figure 5a.

- (1) *Voronoi method*: Vertices are created with the Voronoi method along the medial axis of the geometric domain.
- (2) *Simple vertex prune method*: The vertices are pruned with the simple method as the Voronoi method is likely to produce many vertices in direct vicinity to each other near complex geometric features.
- (3) *Visibility method*: A full visibility graph is created.
- (4) *Minimum spanning tree*: The graph is reduced to a minimum spanning tree for further manipulations.
- (5) *Delete unreachable edges*: All unreachable parts of the graph are deleted.
- (6) *Useful edges*: Useful edges are created to transform the MST to a proper routing graph.
- (7) *Useful vertices*: This method creates vertices at intersections of edges. This adds additional paths through the graph and improves the quality of the simulation results.

Kneidl method

Kneidl (2013) describes a procedure to generate a routing graph especially suitable for narrow environment such as the interior of an office building. The result of the method is shown in figure 5b and the steps to achieve this result are presented in the following.

- (1) *Corner method*: Vertices are generated at the corners of the geometric boundaries according to the user-defined distance parameter.
- (2) *Cone method*: Edges are generated with the cone method algorithm.
- (3) *Extended vertex pruner*: The vertices are pruned based on the specified merge distance and a visibility tolerance while no vertex is merged if a visibility line would be cut.
- (4) *Delete unreachable edges*: All unreachable parts of the graph are deleted.

5 Discussion

The presented modular graph generation approach fits nicely into the highly modular formulation of the MomenTUMv2 pedestrian simulator. The two predefined strategies for graph manipulation already included in the framework allow to create graphs, which fit diverse simulation situations. However, those two strategies for corner-based and route-map graphs might in distinct cases not yield the most suitable graphs, even if the user defined parameters were adjusted iteratively to fit the specific geometric situation.

Nevertheless, due to the intuitive structure of the graph operation methods in the framework, all parameters can be changed quickly and even rearranging the sequence of the graph manipulation methods can be done without much effort. This flexibility allows efficient routing graph generation even if the initial generated graph could not meet the requirements by the simulation scenario. Nonetheless, in future research we will develop evaluation criteria that helps to identify the best fitting routing graph for different pedestrian simulation scenarios.

Future work will also focus on including more graph manipulation approaches to offer techniques which make it possible to generate a proper routing graph for most geometric situations. This is especially important for dynamic scenarios, e.g., closing or opening doors. Due to the fact that the basic framework is already implemented, new graph manipulation strategies and operations can be implemented quickly, as long as the overall structure is maintained.

6 Conclusion

Generating a suitable routing graph is crucial for valid results from a pedestrian simulation. Many different methods for distinct geometric situations exist in literature. For a flexible simulation setup, a framework in which the most suitable method can easily be chosen is most desirable. We presented such a highly flexible framework able to generate custom tailored routing graphs. Our framework structures the graph generation process in different operations, which are categorized as the four major tools for graph manipulation i.e. vertex creation and deletion as well as edge creation and deletion. Using combinations of the proposed graph manipulation methods, it is possible to generate reliable routing graphs for different geometric

simulation scenarios. The user-defined parameters for the manipulation methods allow an accurate adjustment of the resulting graph to complex geometric situations as well as simulation prerequisites such as memory consumption or connectivity of the routing graph.

The framework allows the user to use either completely predefined graph generation strategies, which result directly in a routing graph suitable for many pedestrian simulations. If the graph is to be adjusted to specific conditions, the provided graph manipulation operations can be reordered and their parameters can be chosen in a way that is most suitable for the simulation. Additionally, maintaining the strict division of the four main graph manipulation strategies, new methods can intuitively be added by the users to fit their specific needs in routing graph generation. This allows an even higher flexibility if the default manipulation strategies do not yield the desired result.

The presented graph generation framework was included in the pedestrian simulation software MomenTUMv2. This enables the MomenTUMv2 framework to provide profound methods of graph generations which are used for pedestrian navigation models within MomenTUMv2. Given the high flexibility of all concepts used in MomenTUMv2, it is possible to properly simulate a wide variety of interesting crowd or pedestrian situations.

References

- Andresen, E., Haensel, D., Chraibi, M., and Seyfried, A. (2016). „Wayfinding and cognitive maps for pedestrian models“. In: *arXiv preprint arXiv:1602.01971*.
- Aurenhammer, F., Klein, R., Lee, D.-T., and Klein, R. (2013). *Voronoi diagrams and Delaunay triangulations*. Vol. 8. World Scientific.
- Brogan, D. C. and Johnson, N. L. (2003). „Realistic human walking paths“. In: *Computer Animation and Social Agents, 2003. 16th International Conference on*, pp. 94–101. DOI: 10.1109/CASA.2003.1199309.
- Dijkstra, E. W. (1959). „A note on two problems in connexion with graphs“. In: *Numerische mathematik* 1.1, pp. 269–271.
- Euler, L. (1741). „Solutio problematis ad geometriam situs pertinentis“. In: *Commentarii academiae scientiarum Petropolitanae* 8, pp. 128–140.
- Galea, E. R., Wang, Z., Veeraswamy, A., Jia, F., Lawrence, P. J., and Ewer, J. (2008). „Coupled fire/evacuation analysis of the Station Nightclub fire“. In: *Fire Safety Science* 9, pp. 465–476.
- Geraerts, R. and Overmars, M. H. (2006). „Creating high-quality roadmaps for motion planning in virtual environments“. In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, pp. 4355–4361.
- Graham, R. L. and Hell, P. (1985). „On the History of the Minimum Spanning Tree Problem“. In: *Annals of the History of Computing* 7.1, pp. 43–57. ISSN: 0164-1239. DOI: 10.1109/MAHC.1985.10011.
- Henderson, L. (1974). „On the fluid mechanics of human crowd motion“. In: *Transportation research* 8.6, pp. 509–515.
- Hillier, B., Penn, A., Hanson, J., Grajewski, T., and Xu, J. (1993). „Natural movement: or, configuration and attraction in urban pedestrian movement“. In: *Environment and Planning B: planning and design* 20.1, pp. 29–66.
- Höcker, M., Berkhahn, V., Kneidl, A., Borrmann, A., and Klein, W. (2010). „Graph-based approaches for simulating pedestrian dynamics in building models“. In: *eWork and eBusiness in Architecture, Engineering and Construction*, pp. 389–394.

- Hoogendoorn, S. and Bovy, P. (2004). „Pedestrian route-choice and activity scheduling theory and models“. In: *Transportation Research Part B: Methodological* 38.2, pp. 169–190. ISSN: 0191-2615. DOI: [http://dx.doi.org/10.1016/S0191-2615\(03\)00007-9](http://dx.doi.org/10.1016/S0191-2615(03)00007-9).
- Kielar, P. M., Biedermann, D. H., and Borrmann, A. (2016). *MomentUMv2: a modular, extensible, and generic agent-based pedestrian behavior simulation framework*. Tech. rep. Technische Universität München.
- Kielar, P. M., Biedermann, D. H., Kneidl, A., and Borrmann, A. (2015). „A Unified Pedestrian Routing Model Coupling Multiple Graph-Based Navigation Methods“. In: *Traffic and Granular Flow'15*, p. 16.
- Kneidl, A. (2013). „Methoden zur Abbildung menschlichen Navigationsverhaltens bei der Modellierung von Fußgängerströmen“. PhD thesis. Technische Universität München.
- Kneidl, A., Borrmann, A., and Hartmann, D. (2012). „Generation and use of sparse navigation graphs for microscopic pedestrian simulation models“. In: *Advanced Engineering Informatics* 26.4, pp. 669–680.
- Kruskal, J. B. (1956). „On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem“. In: *Proceedings of the American Mathematical Society* 7.1, pp. 48–50. ISSN: 00029939, 10886826.
- Reynolds, C. W. (1999). „Steering behaviors for autonomous characters“. In: *Game developers conference*. Vol. 1999, pp. 763–782.
- Trudeau, R. J. (1993). *Introduction to Graph Theory (Corrected, enlarged republication. ed.)*
- Wolbers, T. and Hegarty, M. (2010). „What determines our navigational abilities?“ In: *Trends in cognitive sciences* 14.3, pp. 138–146.

Construction Site Pedestrian Simulation with Moving Obstacles

Giovanni Filomeno, Ingrid I. Romero, Ricardo L. Vázquez¹, Daniel H. Biedermann,
Maximilian Bügler

*Lehrstuhl für Computergestützte Modellierung und Simulation,
Technical University Munich,*

Arcisstraße 21, 80333 München, Germany

¹*E-Mail: ricardo.vasquez@tum.de*

Abstract The simulation of pedestrian movement in a dynamic environment is an important challenge in engineering safety simulation. One application is pedestrian dynamics for construction sites. The responsible management must ensure the safety of their workers on the construction site. In recent years, the use of computer simulations has become an important tool for designing safe pedestrian patterns. In this paper we describe a simulation of workers in a construction environment with moving obstacles. The simulation models the movements of the workers and obstacles in a 2D plane, with all the moving obstacles such as bulldozers, excavators and other heavy machines having a safety area around them. This safety area approximates the tendency of humans to avoid moving objects more so than static ones. Our method is based on a combination of repelling and attractive forces. The sum of all the forces shapes the path that a pedestrian will take.

Keywords: Pedestrian dynamics simulations, moving obstacles avoidance, potential fields

1 Introduction

The safety of workers on large construction sites is a complex and important topic in the field of civil engineering. To prevent worker injury, the construction management must have information about the movements of their workers on the construction site. One way to obtain this information is through pedestrian dynamics simulation. Pedestrian dynamics simulations are normally modeled with the help of three independent and interacting layers (Hoogendoorn and Bovy, 2004): strategic, tactical and operational.

The strategic layer determines which destination a pedestrian is heading to next. For every day purposes, this decision process can be carried out by cognitive decision models (Kielar and Borrmann, 2016). However, in the case of construction workers, the different places where they have to work is predefined by the daily work schedule. Thus our simulation uses this schedule to define the sequential order in which a worker goes from one place to another (see Section 3.3).

The tactical layer provides the route along which the pedestrian will move in order to reach the final point. The literature covers a range of tactical models, from the easiest “shortest path”, e.g. the algorithm A* (Kadry et al., 2012), to more complex approaches taking into account psychological knowledge (Kielar et al., 2016). In our simulation model, the tactical layer consists of an attractive force which acts all over the construction site (see Section 3.3) and determine the destination of the worker.

The operational layer models actual movement. For example, in our case the operational layer ensures that the pedestrian will not collide with other pedestrians or obstacles. Moreover the operational layer describes the repulsive forces which act on obstacles, either fixed or moving.

Models of Pedestrian dynamics can be simulated on different space-dependent scales: the three most common are macroscopic, mesoscopic and microscopic. The difference between the macroscopic and non-macroscopic approaches is that macroscopic models only use cumulative parameters (e.g. pedestrian densities), whereas non-macroscopic models consider pedestrians as individual and discrete agents. The most common macroscopic model is the LWR-model (Colombo and Rosini, 2005; Lighthill and Whitham, 1955; Richards, 1956). There are non-macroscopic models that can simulate individual pedestrians: microscopic models, such as social force models (Helbing and Molnar, 1995), and mesoscopic cellular automata models (Blue and Adler, 1998). Additionally, hybrid models can combine models from different spatial scales to lower the overall computational cost (Biedermann et al., 2016; Biedermann et al., 2014; Ijaz et al., 2015).

The content of this paper is divided as follows. The second section describes the background for our solution and discusses the advantages and disadvantages of other approaches. In the third section there is a detailed explanation of the potential field algorithm applied in our solution. In Section 4, we describe the implementation and the main characteristics of the application developed for this project. The fifth section presents the conclusions and future steps for this project.

2 Related work

For this work, defining and describing pedestrians in a dynamic environment, the construction site was described using social force model (Helbing and Molnar, 1995) and obstacle avoidance algorithms. All obstacles are modeled by their shape and define an area that should be avoided by pedestrians. This approach allows us to define interest points for the pedestrians, and create the construction site as a group of zones where the pedestrians can move safely. For the three algorithms that were studied, the following assumptions hold:

- The obstacles can have arbitrary shapes.
- The working space is considered in 2D.
- Every single pedestrian has to reach a final point.

These algorithms are briefly described in the next section. The “Visibility graph” algorithm (Huang and Chung, 2004) defines the obstacles as polygons. The goal is to find the shortest path to the final point. The "Visibility graph" is an indirect method which consist of connecting the two main points, in our case this means starting and final points, with all the corners of the obstacles with straight lines (See Figure 1).

A requirement of the line is that it must not pass through the internal area of the polygon. Considering this, a network that connects all the corners of the obstacles and the final and initial points is created. Finding and choosing the shortest path is now only a problem of comparison. After that, the pedestrian can move following the line of decided path. Figure 1 demonstrates how the algorithm works. The pedestrian moves from the starting point to the corners and then continue from the corners to the end point.

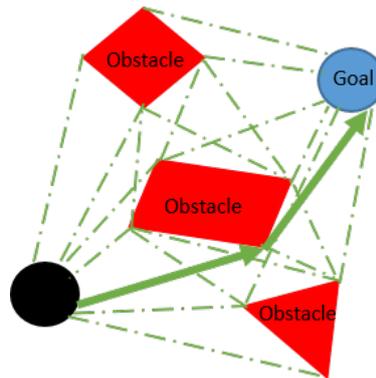


Fig. 1: Visibility Graph example

The disadvantages of this method are that all objects have to be described by polygons, that makes it more difficult to work with continuous shapes. To find the shortest path, all the possible paths must be compared, which leads to a higher computational effort compared with other methods.

Another approach is the so called bug algorithm (Yufka and Parlaktuna, 2009), which mimics the behavior of insects, such as ants. It can be divided into two steps:

- 1- Draw the shortest line that connect the starting point and the final one.
- 2- The pedestrian must move along the line until one of the following two case is met :
 - 2.1- If the pedestrian reaches the final point then stop.
 - 2.2- If the pedestrian touches an obstacle, he stops there, follows the shape of the obstacles until he reaches the shortest line again. Return to step 2 (See Figure 2).

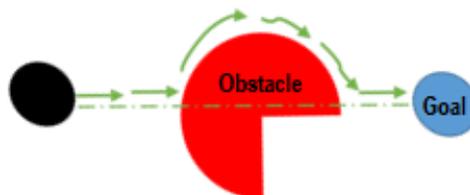


Fig. 2: Bug Algorithm example

This approach works with any shape of the obstacles, however, it requires that the pedestrian gets close to the obstacle and the path is not smooth.

During our research for the best algorithm to implement we discarded the visibility graph and bug algorithm because of the inconveniences of each one:

- In the bug algorithm, the pedestrians have to reach the obstacles in order to avoid them, but this is unacceptable since they should start avoiding the obstacles before reaching them.
- In the case of the visibility graph method, the path may be computed avoiding the edges of the obstacles, however, this method is not very efficient to work with dynamic updating of the elements in the simulation.

3 Simulation model

3.1 Strategic layer

The strategic layer is created because all the workers have a work schedule. To model this, we create arrays for each pedestrian that contain their destinations one needs to reach during the simulation. Each destination represents a place where the worker has to do a task e.g. building a wall, collecting material, etc. Thus, the worker waits for some time at each of these destinations. This represents that the worker is performing some task in that place.

Since the workflow of the worker is linear, there is always only one goal for each point in time. When the current goal is accomplished the next one is adapted as the new goal. This process continues until the pedestrian gets to all the destinations in the array.

3.2 Tactical and operational layer

In this approach the tactical and operational layer are composed of the attractive and repulsive forces of the next destination a pedestrian is heading to.

3.3 Algorithm approach

As an alternative to the approaches described in Section 2, one of our requirements is an algorithm which can avoid obstacles, for example, the movements of vehicles on a construction site.

The approach relies on a model of the construction site having two artificial fields: attraction and repulsion. Our approach thus follows the idea of social force (Helbing and Molnar, 1995). Our approach is a representation of physical phenomena, where the target points can be modeled as sinks and the obstacles as surfaces with high elevation. The pedestrian can then be modeled as a free object that will move around according to the sum of the attractive and repulsive forces (see Figure 3c).

As described above, the destination is the source of the attractive field, acting all over the field with a constant value (see Figure 3a). When the pedestrian is far away or very close to the goal, the strength of the field will decrease. In contrast, all the obstacles will create a repulsive field around them with in a limited range. When the pedestrian is inside the effective area of the obstacle's field, the force will increase as the pedestrian gets closer (see Figure 3b).

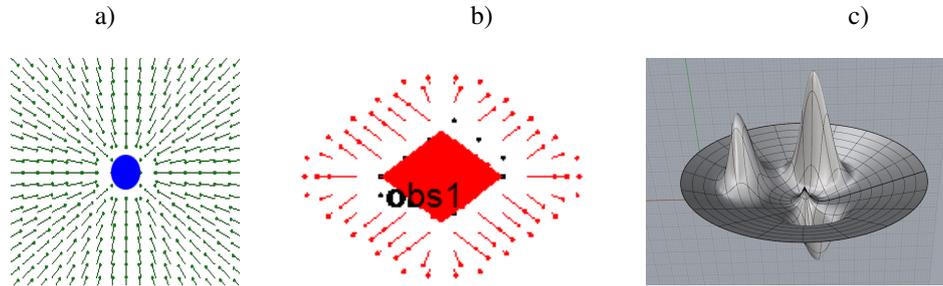


Fig. 3: a) Representation of attractive force, b) Representation of repulsive force, c) The 3d representation of repulsive (peak values) and attractive (dip values) forces

The governing equations for the attractive and repulsive fields are as follows:

Equations for the attraction force

$$U = P_g - P_p \quad (1)$$

$$\hat{U} = \frac{U}{\|U\|} \hat{=} \text{a vector of unit length from the pedestrian to the goal} \quad (2)$$

If the pedestrian is inside the goal zone:

$$F_A = \alpha U \quad (3)$$

If the pedestrian is far from the goal zone:

$$F_A = \beta \hat{U} \quad (4)$$

where:

$P_g \hat{=}$ Position of the Goal

$P_p \hat{=}$ Position of the Pedestrian

$F_A \hat{=}$ Attraction Force

$r \hat{=}$ Radius of the goal zone

$\beta \hat{=}$ Magnitude of the force for all the field

$\alpha = \frac{\beta}{r} \hat{=}$ Constant to decrease the force inside the goal zone

Equations for the repulsive force

$$V = P_p - P_o \quad (5)$$

$$\hat{V} = \frac{V}{\|V\|} \text{ vector of unit length from the obstacle to the pedestrian} \quad (6)$$

If the pedestrian is inside of the restricted area of the obstacle:

$$F_R = S \hat{V} \quad (7)$$

If the pedestrian is inside of the effective area ρ of the obstacle:

$$F_R = \gamma(r_o - \|V\|)\hat{U} \quad (8)$$

where :

- $r_o \hat{=}$ Effect area of obstacle
- $S \hat{=}$ Speed of the obstacle
- $P_0 \hat{=}$ Position of the Obstacles
- $P_P \hat{=}$ Position of Pedestrian
- $F_R \hat{=}$ Repulsive Force
- $\gamma \hat{=}$ Magnitude of the repulsive force

To apply the previous equations, it is necessary to know whether the pedestrian is inside the effective area of an obstacle, which, for polygons is calculated by the following method: the obstacles are defined by a set of points given in counter clockwise direction. For each edge of the obstacle the equation of the line is defined from two points as follows:

$$P_1 = (x_1, y_1) \quad (9)$$

$$P_2 = (x_2, y_2) \quad (10)$$

$$Ax + By + C = 0 \quad (11)$$

where

$$A = y_2 - y_1 \quad (12)$$

$$B = x_1 - x_2 \quad (13)$$

$$C = y_1x_2 - x_1y_2 \quad (14)$$

$$(15)$$

To verify if the pedestrian is in a position in the view side of one edge, the following condition has to be fulfilled (see Figure 4):

$$D_1 + D_2 = L \quad (16)$$

where:

$D_1 \hat{=}$ Distance to the P1 normal line

$D_2 \hat{=}$ Distance to the P2 normal line

$L \hat{=}$ Length of the edge

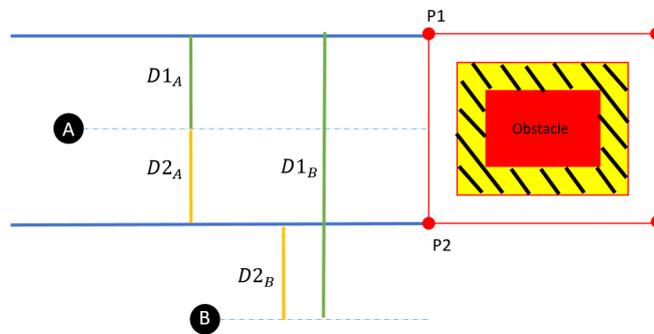


Fig. 4: Method for verifying the presence of pedestrians near an obstacle

Once the pedestrian is confirmed to be in the range of one edge, then the distance to the edge, D , has to be checked to verify that it is smaller than the effective area for that obstacle:

$$D = \frac{Ax + By + C}{L} \quad (17)$$

By combining these two fields at each point on the grid, there will be a unique force equivalent to the sum of gradients from nearby obstacles and the target point.

In Figure 5 we can visualize the sum of both fields in the total area of the scenario, creating, in some point, “pure attractive fields” (A) where the obstacle has no effect. Elsewhere, the sum of both effects, attraction and repulsion, can push together in the direction of the target point (B). A special case is when the attractive and repulsive forces are opposed, and the sum of the forces from the obstacle and the target may become zero (C).

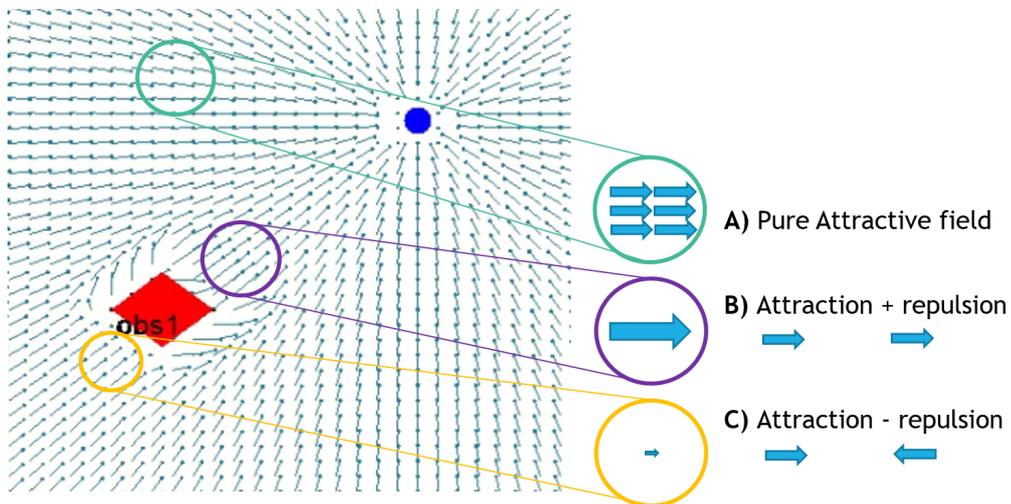


Fig. 5: Potential field behaviour on the construction site

The main force action on the pedestrians is called the action vector. It will give the direction to pedestrian, and the magnitude of the action vector corresponds roughly to the movement of the pedestrian (see 6).

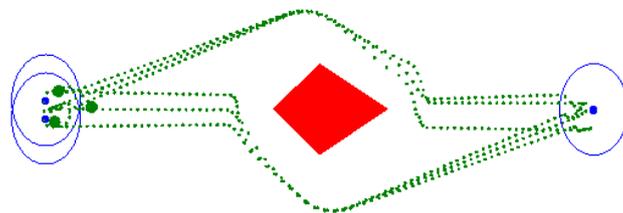


Fig. 6: History of the trajectory followed by the pedestrians

3.4 Local Zero

One important problem is the local zero or “local minima” (Koren and Borenstein, 1991), where the attracting and repulsing fields have the same values but in opposite directions. In these cases the pedestrian will get stuck in that point. To avoid the creation of convergence points out of the desired target, two additional potential fields may be added to the previous formulation to take the pedestrian out of the local minima:

- *Rotational field around the obstacles*: this option will help us to avoid the local minimum, breaks the symmetry and guides the pedestrian around groups of obstacles when they are in movement (see Figure 7a).
- *Random field*: the creation of small random forces around the obstacle gets the pedestrian unstuck and avoids some local minimum (see Figure 7b).

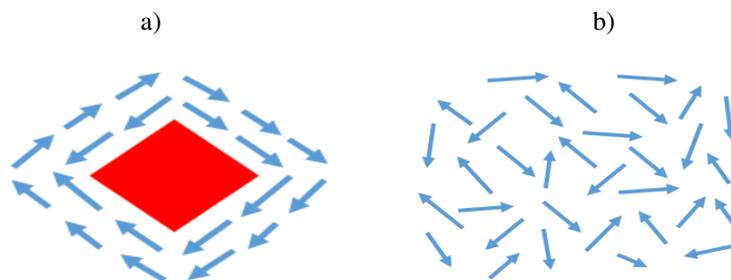


Fig. 7: a) Rotational field example, b) Random field example

3.5 Moving Obstacles

In this section, the representation of all the obstacles in the simulation is described in detail. As mentioned before, the main goal is to obtain results from a dynamic simulation, which means that the environment consists of moving obstacles with different shapes that should be avoided by the pedestrians. The moving obstacles represent the machines in the construction site, such as excavators, bulldozers and other heavy machines. The different types of obstacles available in the simulation are:

- Obstacles of arbitrary shapes with fixed path and a waiting time at some points
- Moving obstacles that avoid each other
- Fixed obstacles
- Pedestrians are also obstacles that are avoided by other pedestrians

The methodology described in Section 3.4 is applied to fixed obstacles. However, since the goal of this work is to create dynamic simulations with moving objects, the following considerations were applied:

- The range of the potential field is larger in the movement directions of the obstacle, See Figure 8b.

- The rotation field is always pointing in the opposite relative direction to the movement of the obstacle. Moreover it is positioned in the obstacle center (see Figure 8b).

$$s = \hat{V} * \hat{F} \quad (18)$$

where:

$$\begin{aligned} \hat{V} &= \text{Vector of the velocity} \\ \hat{F} &= \text{Potential field created by the obstacle} \end{aligned}$$

If $s > 0$ then the potential field created by the obstacle is increased by 50 percent.

$$r = \hat{V} * \hat{R} \quad (19)$$

where:

$$\hat{R} \hat{=} \text{Rotational field created by the obstacle}$$

If $r > 0$ then the rotational field is inverted

$$\hat{R} = -\hat{R} \quad (20)$$

The sum of these two effects will help pedestrians avoid the moving obstacles by creating paths in the opposite direction of the movement of the obstacles, which leads to safer paths across the simulated environment. Figure 8 visualizes the difference between fixed and moving obstacles.

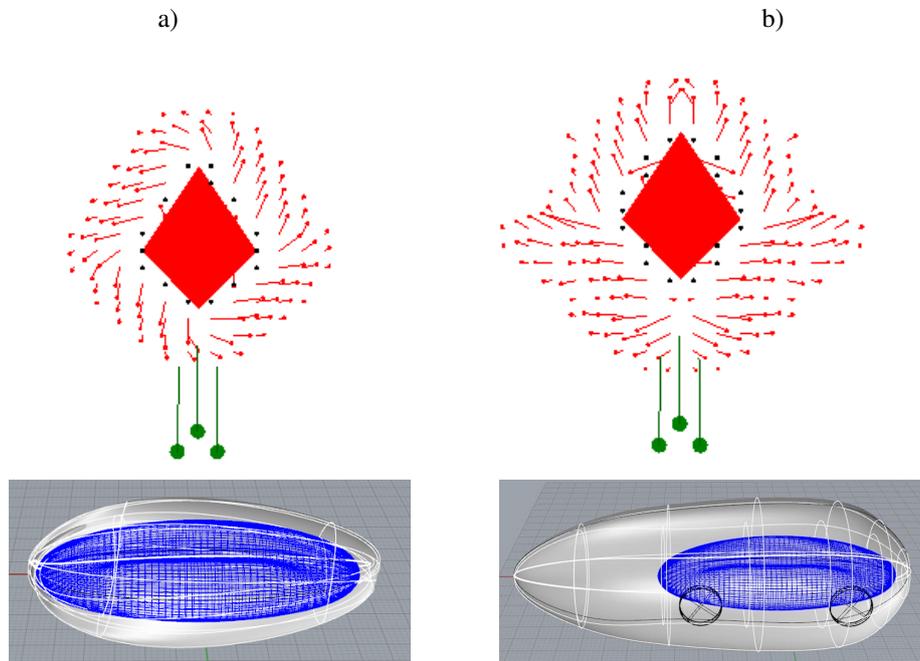


Fig. 8: Visualization in 2D & 3D of potential fields a) Fixed obstacle b) Moving obstacle

4 Flow-Chart of the simulation model

This section explains the considerations applied in the simulation and the way it was implemented. All the pedestrians and objects are created with a finite number of points of interest representing goals. As it is described in Figure 9, to reach a goal, each pedestrian computes the distance to the goal, thereby calculation the attractive field. The repulsive field for the pedestrians is computed by measuring the distance to all the obstacles and other pedestrians and then summing up the effects of all of them to create the total repulsive field. After that, the norm of the force is switched according to the value of the distance. If it is lower than the safety distance, then it reaches the maximum value, and, if not, the norm is proportional to the distance.

The repulsive force for the moving obstacles is computed in a similar way, with the difference that the influence of the pedestrians is not considered, (i.e. changing direction, decreasing velocity, etc.). Moreover, the computation of the potential field is done only considering the current state of all the elements in the simulation. The pedestrians do not make any predictions about the obstacles and vice versa.

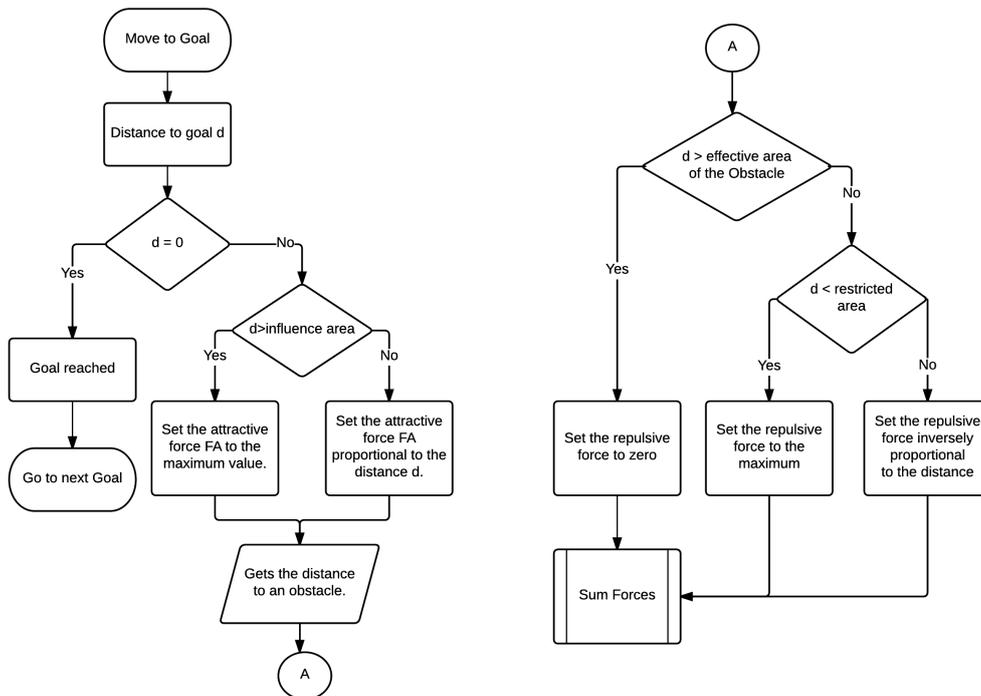


Fig. 9: Flow Diagram of Potential Fields

5 Conclusion

In this paper, we present an algorithm allowing pedestrians to avoid obstacles in a dynamic environment. Our approach for avoiding obstacles is successful representing the movement of pedestrians in a dynamic environment. The working stations of the workers and the routes of the machines were used as input to set up the environment and simulate the interaction between

them. This was done bearing in mind that the pedestrians will avoid both other pedestrians and obstacles. Regarding the machines, or moving obstacles, they will only avoid other obstacles, but the presence of pedestrians does not have an effect on their trajectories.

The developed application outputs the history of the points that describe the trajectory. This can be visualized and may be used to define both the safe routes in a construction site where the workers can move with low risk, and the high risk areas that the pedestrians should avoid. The algorithm used in this project is not limited to simulation of construction sites. Its applications can be extended and used to create any scenario where the user wants to find safe paths or zones for pedestrians in dynamic environments. An aspect that must be considered for future work is the pedestrians' effect on the moving obstacles. Most of the time the machine is not aware of the pedestrians and will not try to avoid them. However, there may be situations where the driver of the machine sees the pedestrians and may try to avoid them.

References

- Biedermann, D. H., Torchiani, C., Kielar, P. M., Willems, D., Handel, O., Ruzika, S., and Borrmann, A. (2016). „A hybrid and multiscale approach to model and simulate mobility in the context of public event“. In: *Proc. of the mobil.TUM conference 2016*.
- Biedermann, D. H., Kielar, P. M., Handel, O., and Borrmann, A. (2014). „Towards TransiTUM: A generic framework for multiscale coupling of pedestrian simulation models based on transition zones“. In: *Transportation Research Procedia 2*, pp. 495–500.
- Blue, V. J. and Adler, J. (1998). „Emergent fundamental pedestrian flows from cellular automata microsimulation“. In: *Transportation Research Record 1644*, pp. 29–36.
- Colombo, R. M. and Rosini, M. D. (2005). „Pedestrian flows and non-classical shocks“. In: *Mathematical Methods in the Applied Sciences 28.13*, pp. 1553–1567.
- Helbing, D. and Molnar, P. (1995). „Social force model for pedestrian dynamics“. In: *Physical review E 51.5*.
- Hoogendoorn, S. P. and Bovy, P. H. (2004). „Pedestrian route-choice and activity scheduling theory and models“. In: *Transportation Research Part B: Methodological 38.2*, pp. 169–190.
- Huang, H.-P. and Chung, S.-Y. (2004). „Dynamic visibility graph for path planning“. In: *Intelligent Robots and Systems*. Vol. 3, pp. 2813–2818.
- Ijaz, K., Sohail, S., and Hashish, S. (2015). „A survey of latest approaches for crowd simulation and modeling using hybrid techniques“. In: *17th UKSIMAMSS International Conference on Modelling and Simulation*, pp. 111–116.
- Kadry, S., Abdallah, A., and Chibli, J. (2012). „On The Optimization of Dijkstra's Algorithm“. In: *Informatics in Control, Automation and Robotics 133*, pp. 393–397.
- Kielar, P. M., Biedermann, D. H., Kneidl, A., and Borrmann, A. (2016). „A Unified Pedestrian Routing Model Combining Multiple Graph-Based Navigation Methods“. In: *Proceedings of the 11th Conference on Traffic and Granular Flow*.
- Kielar, P. M. and Borrmann, A. (2016). „Modeling pedestrians' interest in locations: A concept to improve simulations of pedestrian destination choice“. In: *Simulation Modelling Practice and Theory*, pp. 47–62.
- Koren, Y. and Borenstein, J. (1991). „Potential field methods and their inherent limitations for mobile robot navigation“. In: *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 1398–1404.

Construction Site Pedestrian Simulation with Moving Obstacles

- Lighthill, M. J. and Whitham, G. B. (1955). „On kinematic waves. II. A theory of traffic flow on long crowded roads“. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 229. 1178, pp. 317–345.
- Richards, P. I. (1956). „Shock waves on the highway“. In: *Operations research* 4.1, pp. 42–51.
- Yufka, A. and Parlaktuna, O. (2009). „Performance Comparison of BUG Algorithms for mobile Robots“. In: *5th International Advanced Technologies Symposium*.

Is macroscopic resolution enough? – A study about the use of different spatial scales in pedestrian dynamics

Daniel H. Biedermann¹ and Sebastian Esser

*Chair of Computational Modeling and Simulation, Technische Universität München, Arcisstr.
21, Germany*

¹*E-Mail: daniel.biedermann@tum.de*

Abstract Secure evacuation of buildings or crowded areas is an important topic in civil engineering. Pedestrian dynamics simulations help to detect dangerous parts of an evacuation scenario. Unfortunately, simulations of complex scenarios result in high computational costs. The costs depend mainly on the used spatial scale. Models from the macroscopic scale (network flow models) have a low and models from the microscopic scale (spatial continuous models) a high spatial resolution. The spatial resolution of mesoscopic models (cellular automata) is located in between. High spatial resolution results in high computational costs. But low spatial resolution may lead to insufficient simulation results. Consequently, it is difficult to decide for one scale in a general way. The scenario's infrastructural characteristics and desired application fields define which spatial scale should be used. We performed field studies to develop a generic method to determine the optimal scale based on the given evacuation scenario.

Keywords: pedestrian dynamics, experiment, spatial scales, classification procedure

1 Introduction

Pedestrian dynamics simulations have to combine two contrary requirements: they have to model a realistic pedestrian behavior but should to be computational efficient at the same time. The accuracy of a pedestrian dynamics model depends to a large extent on its spatial resolution. Unfortunately, the spatial resolution influences the needed computational effort. Thus, different types of pedestrian dynamics models have individual attributes according to spatial resolution and computational effort. In pedestrian dynamics, three different types of spatial resolution are mainly used: macroscopic, mesoscopic and microscopic.

Macroscopic simulation models have the lowest computational costs and lowest spatial resolution. These models reduce the simulation scenario to a simple network graph and calculate with cumulated parameters (e.g. pedestrian densities) instead of individual pedestrian agents. They are mainly suitable for scenarios with directed pedestrian flows. An one dimensional scenario structure supports such walking behavior. A simulation model type much suitable for such scenarios are the LWR-models from Lighthill and Whitham (1955) and Richards (1956). Originally designed for vehicle traffic on highways, they were adopted to pedestrian dynamics

Is macroscopic resolution enough? – A study about the use of different spatial scales in pedestrian dynamics

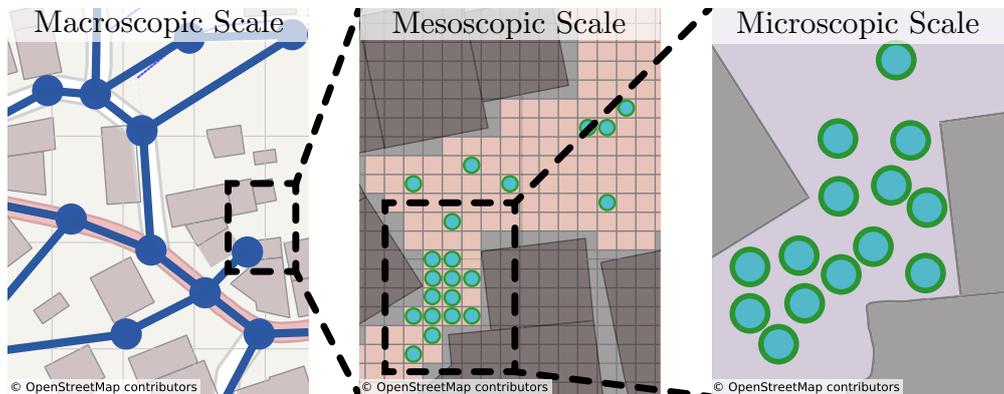


Fig. 1: From left to right: examples of a macroscopic, mesoscopic and microscopic simulation models.

by Colombo and Rosini (2005). These models use the continuity equation to describe flow of human crowds.

Models of the mesoscopic scale have an average spatial resolution and an average computational effort. These cellular automata simulate individual pedestrian agents on a cellular grid, whereas each cell can be free, occupied by an obstacle or occupied by exactly one pedestrian. The assignment of pedestrians to regular grid cells saves computational time, but artificially restricts the maximal spatial resolution to the size of one unit cell. Mesoscopic models are most suitable for open areas without bottle necks and with medium high densities (Biedermann et al., 2016). A well-known representative is the cellular automaton from Blue and Adler (2001), which is able to model bi-directional movement of individual pedestrians.

In the case of high densities or small scale obstacles (e.g. bottlenecks at doors), microscopic models have to be used due to their high spatial resolution. Unfortunately, microscopic models are computational heavy compared to mesoscopic or even macroscopic approaches. These models simulate individual pedestrians on a continuous space. This means that the spatial resolution is not limited by artificial restrictions. The most wide-spread type of microscopic modeling are the social force models (Helbing and Molnar, 1995). These models assign a repulsive force to each simulation object (e.g. obstacles, pedestrians) and an attractive one to the next intermediate destination of a pedestrian. The superposition of all forces describes the movement of this pedestrian.

Figure 1 shows an example for the holistic use of all three scales: the macroscopic approach is used for the transportation along a street network, the open area is simulated by a mesoscopic model and at the very dense region a microscopic approach is used. The boxes with dashed lines describe the transition between individual scales. Such hybrid approaches are helpful to save computational time, since only regions which are potentially dangerous (e.g. bottlenecks, local high densities) are simulated in a high spatial resolution, while the remaining parts are calculated by less costly models (Biedermann et al., 2016). A sufficient overview about current hybrid approaches are presented by Ijaz et al. (2015). For example, Anh et al. (2011) combine a macroscopic LWR-model with an agent-based leader-follower approach. Rabiaa and Foudil (n.d.) propose a hybrid model simulating in real-time, by combining a the overall movement of a crowd with individual pedestrian behavior. An approach for a generic coupling of mesoscopic and microscopic simulation models is described by Biedermann et al. (2014).

All these hybrid approaches have in common that they simulate different regions of the

simulation scenario with pedestrian dynamics models from different scales. Therefore, it is necessary to know which region of the simulation scenario should be simulated on which spatial resolution. In this work, we analyze the characteristics and attributes from macroscopic, mesoscopic and microscopic models. Additionally, we present new methods which are able to determine the most suitable scale for a given simulation scenario.

2 Characteristics of different spatial scales

A scenario with characteristics from a specific spatial scale should be simulated with a simulation model from the same scale. If models from other scales are used for the simulation process, various disadvantages can occur. Table 1 gives an overview about this issue: the simulation results can suffer from inaccuracy, if a too low spatial scale is used for the simulation. If a too high spatial scales is used, the simulation will be inefficient since too much computational effort is spent for the simulation process. Thus, a clear definition of spatial scale characteristics is necessary to assign the correct resolution to a given scenario.

Table 1: Relationship of the suitability between multi-scale scenarios and simulations.

	Microscopic Scenario	Mesoscopic Scenario	Macroscopic Scenario
Microscopic Simulation Model	suitable	inefficient	very inefficient
Mesoscopic Simulation Model	not accurate	suitable	inefficient
Macroscopic Simulation Model	not accurate	not accurate	suitable

Macroscopic pedestrian dynamics scenarios are characterized by their one-dimensional like structure, which means that pedestrians on the macroscopic scale share the same walking direction. Such behavior is mainly recognized on directed transportation facilities like street networks or walking paths. But it also appears on open space scenarios if most of the pedestrians are heading towards similar directions. Pedestrians of a crowd move into similar directions if their points of interest are the same. Destinations at the entrance areas of a public event are examples for such shared points of interest: every visitor is heading to one of the entrances and this results in directed flows towards these areas. Figure 2a shows an example of a macroscopic scenario. People in this picture walk towards the same destination and therefore a directed flow is clearly visible. Section 3 describes mathematical methods to recognize such macroscopic behavior in a scenario.

Pedestrian dynamics scenarios on the mesoscopic scale are characterized by a diverse and therefore two-dimensional walking behavior. This is observed on open scenarios in which pedestrians of a crowd have no or only few shared points of interest. Therefore, they are heading towards individual destinations and no directed flow occurs. An example for this scenario type is a weekly market in a small town. All pedestrians have their own points of interest (e.g. different vendor stands, people they know) and no directed flows can occur. Figure 2b represents a typical mesoscopic scenario on a schoolyard. The pupils are walking around without having shared destinations, everyone walks towards the direction he or she likes.

Scenarios on the microscopic spatial scale are characterized by their high local densities or the existence of movement space restricting small scale obstacles (e.g. bottlenecks). In

Is macroscopic resolution enough? – A study about the use of different spatial scales in pedestrian dynamics

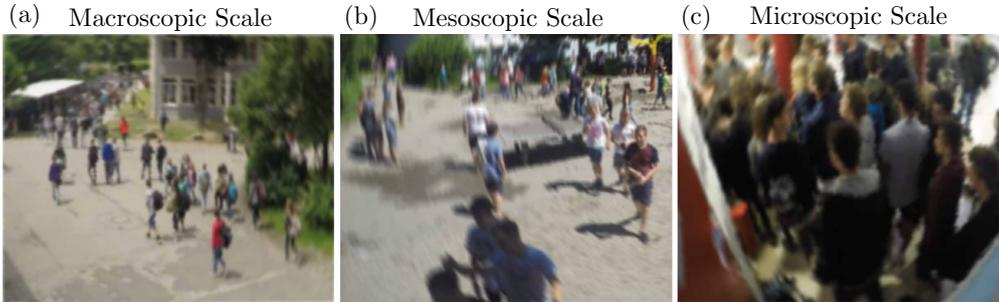


Fig. 2: From left to right: examples of a typical macroscopic, mesoscopic and microscopic scenario.

the case of high local densities the pedestrians' velocity is rather small (Weidmann, 1992). Mesoscopic simulation models are not able to simulate microscopic scenarios sufficiently since high densities and critical small scale objects are difficult to model in mesoscopic simulation approaches. Figure 2c shows a typical microscopic scenario: a dense crowd is visible in an environment with movement restricting obstacles (e.g. fire extinguisher, columns).

Table 2: Necessary geometries and maximal pedestrian densities for mesoscopic and microscopic models. Size values of the torsos are based on Weidmann, 1992.

<i>Mesoscopic</i>	Minimal Cell Size	Maximal Density
Triangular Cell Shape	$a_t = 2\sqrt{3}r \approx 0,80 \text{ m}$	$\frac{1}{A_t} = \frac{4\sqrt{3}}{3a_t^2} \approx 3,64 \text{ ped/m}^2$
Quadratic Cell Shape	$a_q = 2r = 0,46 \text{ m}$	$\frac{1}{A_q} = \frac{1}{a_q^2} \approx 4,73 \text{ ped/m}^2$
Hexagonal Cell Shape	$a_h = \frac{2\sqrt{3}}{3}r \approx 0,27 \text{ m}$	$\frac{1}{A_h} = \frac{2\sqrt{3}}{9a_h^2} \approx 5,46 \text{ ped/m}^2$
<i>Microscopic</i>	Torso Size	Maximal Density
Circular Torso	$r = 0,23 \text{ m}$	$\frac{1}{A_c} = \frac{\eta_c}{r^2\pi} \approx 5,46 \text{ ped/m}^2$
Elliptical Torso	$a = r$ $b = \frac{1}{2}r$	$\frac{1}{A_e} = \frac{\eta_e}{a \cdot b \pi} \approx 10,47 \text{ ped/m}^2$

The decision if a scenario is mesoscopic or microscopic depends on two aspects: the maximal density in this scenario and on the occurrence of critical small scale obstacles. The maximal density which can be represented by mesoscopic models is limited by one pedestrian per cell, which results in the maximal possible densities described in Table 2. In simulation models from the microscopic scale pedestrians are typically represented as circular or elliptical shaped torsos. According to Weidmann, 1992, the space requirements of a pedestrian can be represented by either a circle with a radius of $r = 0,23 \text{ m}$ or as an ellipse¹ with half-axes of $a = 0,23 \text{ m}$ and $b = 0,115 \text{ m}$. Since a grid cell has to contain the whole torso of a pedestrian, a necessary minimum cell size can be calculated based on radius r respectively half-axis a . The half-axes of this ellipse result in an aspect ratio of $\frac{a}{b} = 2$. In theory, the maximal density would be $6,02 \text{ ped/m}^2$ for circular shaped torsos and $12,03 \text{ ped/m}^2$ for elliptical shaped ones. However, contrary to regular triangles, rectangles and hexagons, it is not possible to fill a two dimensional

¹Actually, a circle can be also seen as an ellipse with aspect ratio $\frac{a}{b} = 1$.

plane completely with circles or ellipses. Circles have a maximal packing density of $\eta_c = \frac{\pi}{2\sqrt{3}} \approx 0.91$ (Chang and Wang, 2010) and ellipses with an aspect ratio of $\frac{a}{b} = 2$ have a maximal packing density of $\eta_e \approx 0.87$ (Guises et al., 2009). The packing density η_c (η_e) describes the maximal percentage share of an area which can be filled by packing only circles (ellipses). Thus, the theoretical pedestrian density of microscopic models has to be reduced according to these factors to obtain the real density limit. Table 2 gives an overview about the maximal densities and the needed minimal cell size to cover a circular or elliptical torso of a pedestrian. The maximal density of hexagonal shaped cells is identical to the maximal density of circles. This is intuitively clear, since the optimal way to pack circles in a plane is a hexagonal lattice (Chang and Wang, 2010). Furthermore, this overview shows clearly that critical pedestrian densities over $6,0 \text{ ped/m}^2$ (Oberhagemann, 2012) can be only modeled by elliptical shaped pedestrian agents.

The occurrence of movement space restricting small scale obstacles is another characteristic of microscopic scenarios. Mesoscopic models have problems with such objects since their spatial resolution is limited by the size of their grid cells. A grid cell is either occupied or free. So even if only a small part of an obstacle crosses a grid cell, the cell has to be assigned as occupied. Otherwise, a pedestrian on such a cell would enter unaccessible areas which would lead to unrealistic simulation results. Unfortunately, assigning the total cell as occupied unrealistically restricts the total movement space of the scenario (Biedermann et al., 2016). A possibility to lower the number of unrealistic closed cells is the use of a half-filling method (Mayer et al., 2014). In this approach, a thin obstacle in a cell only cuts the connection to neighboring cells, but leaves the cell accessible. In some cases, this helps to prevent the unrealistic restriction behavior of mesoscopic scenarios. However, if too many small scale obstacles occur in a scenario, a microscopic simulation model has to be used. Section 4 describes methods how it is possible to classify scenarios as either mesoscopic or microscopic.

3 Classification of macroscopic scenarios

Macroscopic models are most suitable for scenarios which support one dimensional movement. Such scenarios are mainly networks of streets or pathways, which enforce one-directional or bi-directional (in the case of counter flows) movement. One-dimensional movement means, that the movement of observed pedestrians is strongly biased towards one direction. Based on our definition that a macroscopic model is most suitable for the description of such movement behavior, we use weighting methods to recognize macroscopic simulation models.

To detect macroscopic movement behavior, we have to recognize patterns of preferred directions. Thus, we have to study the walking directions from pedestrians of the observed scenario. A common way to receive this kind of information is the use of video observations (Biedermann et al., 2015). Based on these data samples, we are able to determine the position of pedestrians at different times. Therefore, a pedestrian's current direction can be calculated by comparing the pedestrian's position at two different times:

$$\vec{d}_i(t') = \vec{o}_i(t_2) - \vec{o}_i(t_1) \quad t_1 \leq t' \leq t_2 \quad (1)$$

The comparison of the direction vector $\vec{d}_i(t')$ from different pedestrians P_i for times t_1 and t_2 can be done by the help of a comparison vector \vec{v} . Vector \vec{v} is heading towards an arbitrary direction and is only needed to normalize the direction of all pedestrians P_1 towards the same

Is macroscopic resolution enough? – A study about the use of different spatial scales in pedestrian dynamics

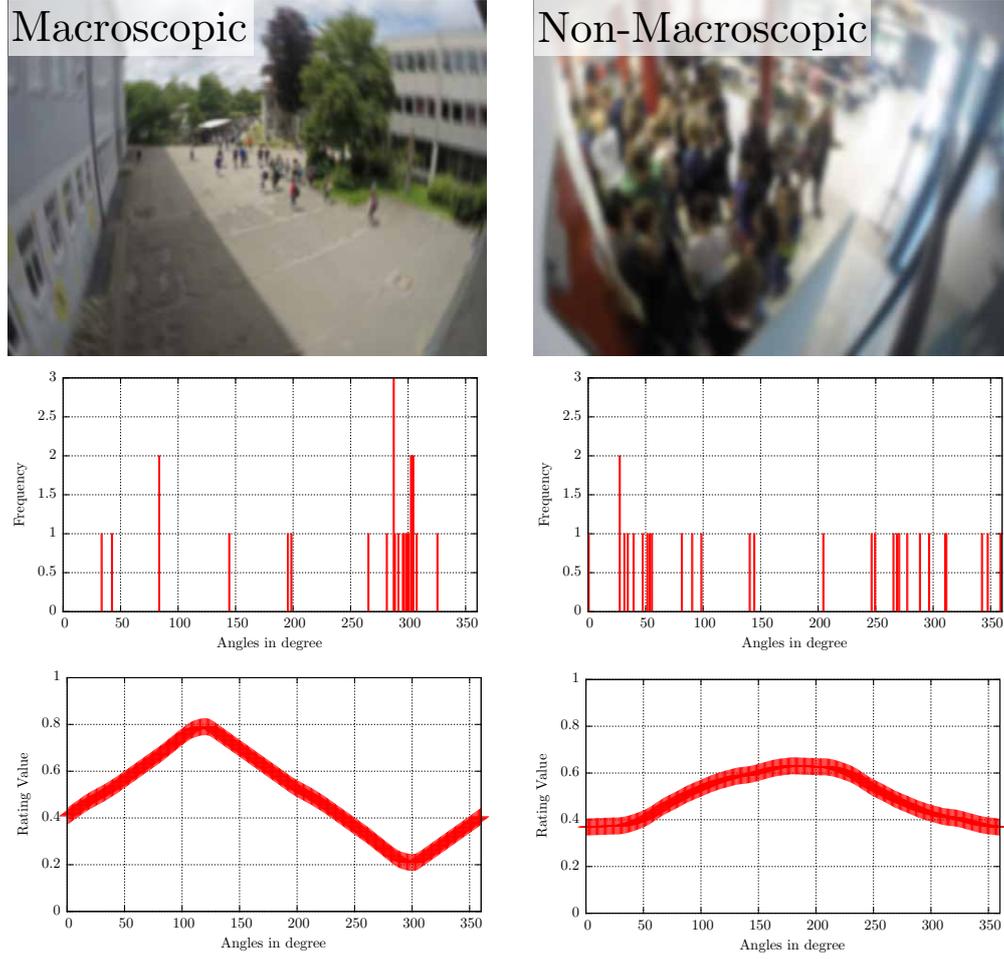


Fig. 3: Macroscopic and non-macroscopic scenario with their associated classic histogram and weighting diagram based on rating values.

direction. Thus, we calculate a direction angle $\alpha_i(t')$ for each pedestrian P_i :

$$\alpha_i(t') = \arccos \left(\frac{\vec{d}_i(t') \circ \vec{v}}{|\vec{d}_i(t')| \cdot |\vec{v}|} \right) \quad (2)$$

Based on angle $\alpha_i(t')$, we determine if preferred walking directions exist and if the movement is biased enough to define the scenario as macroscopic. An often used visualization method to recognize a preference in data samples are classic histograms. For this method, we divide the possible direction space from $\Omega = [0, 2\pi]$ into constant intervals and calculate how many data points are located in each interval. This method comes with two major issues: If we use small intervals a large amount of data samples are necessary to detect a preferred direction with this method. Secondly, the discretization of our interval strongly influences the localization of our data samples. Thus, we introduce a rating histogram method, which is able to emphasize peaks and dips with a low amount of given data points.

In a first step, we define M constant intervals $I = [I_{\downarrow}, I_{\uparrow}] \in \Omega$ with $n \cdot I = 2\pi$ for our angular space $\Omega = [0, 2\pi]$. The size of one interval for our rating histogram approach can be

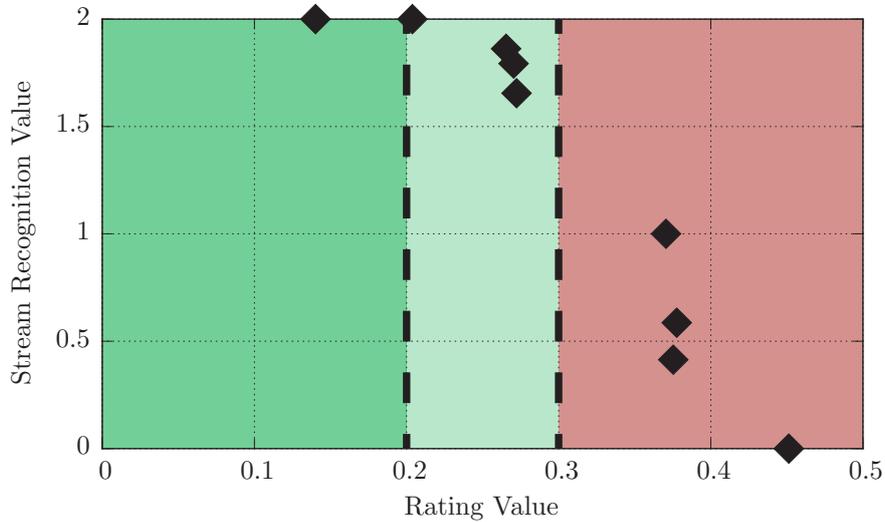


Fig. 4: Relationship between the stream recognition value and the rating value.

arbitrary small, since the location of data samples is independent from size and discretization of our intervals. For each interval, we rate how strong the data points are biased towards the intervals mean value $\bar{I} = \frac{1}{2}(I_{\uparrow} - I_{\downarrow})$. This rating value R_I is influenced how close all N data points α_i are located towards the mean value \bar{I} of this interval I :

$$R_I = \frac{1}{\pi} \sum_i^N \min(|\bar{I} - \alpha_i|, 2\pi - |\bar{I} - \alpha_i|) \quad (3)$$

A low rating value R_I stands for a strong bias towards this angle interval I . The value range equals $R_I = [0, 1]$. Figure 3 shows the difference between a classical histogram and the weighting diagram approach. Solely based on the classic histogram it is difficult to recognize if the tested scenario is macroscopic or not: both classic histograms show a small clustering around some angles, but no clear tendency can be seen. Contrary to the classic histogram, the weighting diagram shows a clear preference towards the angle interval $I = [300^\circ, 301^\circ]$: The minimal value of the weighting diagram determines how one-dimensional like a scenario is. In this example, the macroscopic scenario shows a minimal rating value of 20%, whereas the non-macroscopic scenario has a rating value from more than 40%. Thus, the rating value of the non-macroscopic scenario is twice as high as the macroscopic one. The range of the minimal rating value for a scenario is always in the interval $[0, 0.5]$: the minimal rating value equals zero if all vectors are heading towards the same direction and 0.5 if all vectors are evenly distributed towards all directions.

The next step is to find a sufficient threshold for the rating value R_T : a scenario has to be considered macroscopic, if its rating value is lower than R_T . We made an experimental study to define upper and lower bounds of these threshold parameters. For this study we used one of the most flexible pattern recognition machines, the human mind (Tarr, 2000). In our study, we showed the 29 participants videos from 9 different pedestrian dynamics scenarios. Afterwards, the participants had to mention if they did or did not recognize a pedestrian stream on the video. A scenario was given a point if a stream was visible for the participant, zero points if the participant was unsure and minus one point if the participant did not see any pedestrian

streams. Some time later, the same videos were shown to the participants but in a different order. Therefore, each participant was able to rate the stream recognition value of each video between $[-2, 2]$. We calculated the minimal rating value for each scenario and associated these values with the average stream recognition value. The rating values of the scenarios were determined by calculating the direction vectors of moving pedestrians in three exemplary short video scenes for each scenario. In one scenario, which had two clearly distinguishable and orthogonal streams, we only considered the main stream, since we were interested in the one dimensional structure only. For all other scenarios all moving pedestrians were considered for the calculation. The relationship between the average stream recognition value and the minimal rating value for each scenario is shown in Figure 4. Two scenarios received from all participants the maximal possible rating value of two, which means that all participants recognized a pedestrian stream in these scenarios. Furthermore, these two scenarios had the best rating values ($\lesssim 0.2$) compared to all other tested scenarios. Consequently, we propose that scenarios with a minimal rating value $R_T < 0.2$ can be seen as a macroscopic scenario. Scenarios with a minimal rating value between $[0.2, 0.3]$ were rated quite high (≥ 1.5), but not all participants recognized pedestrian streams in these scenarios. However, we propose that these scenarios have a tendency to be macroscopic. Scenarios with a minimal rating value higher than 0.3 had a stream recognition value of ≤ 1.0 . Thus, we propose that these scenarios are non-macroscopic.

The presented values are sufficient to check if a scenario has a one-dimensional structure. If counter-flows occur, the method is insufficient, since the contrary flows cancel out each other. Thus, the rating value would not indicate a macroscopic scenario. Thus, if a macroscopic scenario with counter-flow should be detected, the direction vectors $\vec{d}_i(t')$ have to be mapped towards the same half sphere of the unit circle. This can be done with the help of the Heaviside-function:

$$\vec{d}_i^*(t') = -2\vec{d}_i^{\bar{*}}(t') \cdot \mathcal{H}(\alpha_i(t') - \pi) + \vec{d}_i^{\bar{*}}(t') \quad (4)$$

To get the rating value R_I^* for counter-flows, an additional reduction of the direction space Ω from 2π to π is necessary:

$$R_I^* = \frac{1}{0.5\pi} \sum_i^N \min(|\bar{I} - \alpha_i|, \pi - |\bar{I} - \alpha_i|) \quad (5)$$

4 Classification between mesoscopic and microscopic scenarios

Mesoscopic and microscopic scenarios can be distinguished by two aspects: the local pedestrian density of a scenario and the amount of movement space restricting obstacles.

If the maximal pedestrian density of a scenario extends the limit of the used mesoscopic model (see Table 2), an appropriate microscopic model has to be used to obtain realistic simulation results. To classify the local density of a scenario, different methods exist in literature (Duives et al., 2015). A classic approach is the grid based method, which divides the scenario by a cellular grid and counts the number of pedestrians per cell to determine the local pedestrian density. The main issue with this approach is the discontinuity of the measured density. Steffen and Seyfried, 2010 propose the Voronoi-method as an alternative for a more continuous measurement of pedestrian density. This method does not count the pedestrians per area; instead it calculates for each pedestrian his or her accessible area: all fractions of an area, which are closest to a given pedestrian are assigned to this person. An issue of this method is the inclusion of obstacles and walls into the calculation of accessible area fractions. Edie, 1963 defined another method, the x-t approach, to calculate pedestrian densities (Duives et al., 2015; Leonard and Ni, 2006). The x-t

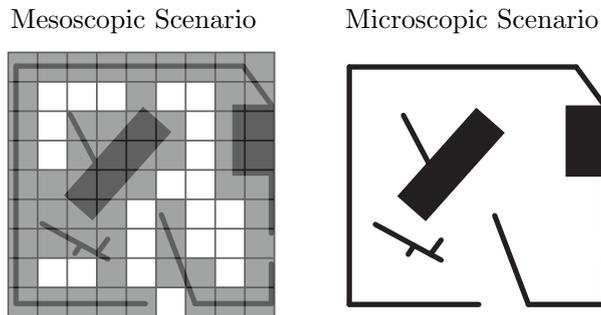


Fig. 5: Comparison of the restricted areas for mesoscopic and microscopic models.

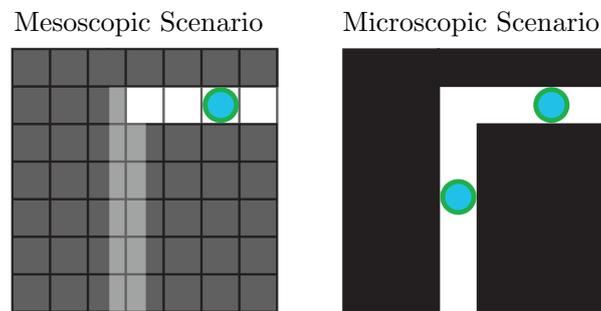


Fig. 6: For a mesoscopic pedestrian, the corridor is closed, though this bottleneck is accessible for a microscopic pedestrian.

method is a dynamic extension of the classic approach. It divides the scenario into a cellular grid and counts how long pedestrians stay in each cell. This enables a more continuous measurement of density. We recommend the Voronoi-approach or the x-t method for the calculation of density for a pedestrian dynamics scenario. Based on the calculated densities, it is possible to easily decide according to Table 2 if a scenario is either mesoscopic or microscopic.

But even if a scenario has a low density, a microscopic treatment can be necessary if small scale obstacles restrict the movement space of pedestrians. Since the spatial resolution of mesoscopic models is limited by the size of the unit cell, unrealistic spatial restrictions can occur: if we consider a bottleneck, e.g. an open door in a wall, its size has to be larger than the pedestrian's body radius (smaller half-axis) in a circular (elliptical) microscopic model. Otherwise, the bottleneck's space is too small for a pedestrian to walk through this opening in the wall. But such large bottlenecks can be still too small for pedestrians simulated on the mesoscopic scale. Figure 5 shows an example for the different restricted areas of mesoscopic and microscopic models for the same scenario. Grey colored cells represent cells occupied by obstacles, whereas white cells are free and therefore accessible for pedestrians. This example shows that mesoscopic grid cells can restrict the available movement space of a scenario in an unrealistic manner.

A possibility to distinguish between mesoscopic and microscopic scenarios is the ratio R_A between their restricted areas A_{meso} and A_{micro} :

$$R_A = \frac{A_{\text{micro}}}{A_{\text{meso}}} \leq 1 \quad (6)$$

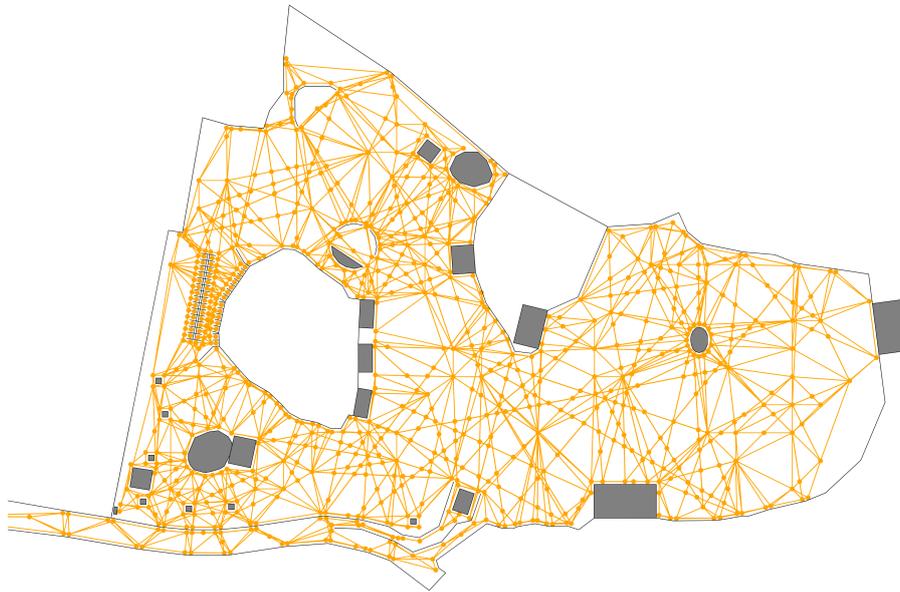


Fig. 7: Visibility graph for an exemplary scenario²⁷.

A ratio R_A close to zero is most likely a microscopic scenario, since the mesoscopic model restricts much more movement space than the microscopic one. Contrary, a ratio R_A close to one often can be simulated realistically by mesoscopic models. However, a mesoscopic model can lead to unrealistic movement restrictions, even if the total restricted area has a similar size. Figure 6 shows such an example. The mesoscopic model restricts the movement of the whole corridor, though the size of the restricted area is quite similar compared to the microscopic model. Consequently, the comparison of restricted areas A_{micro} and A_{meso} is no absolute certain way to distinguish between mesoscopic and microscopic scenarios. The most important aspect is the question if all relevant paths are accessible for pedestrians from both models. A way to determine important paths on a scenario are visibility graphs (Kneidl et al., 2012). Such graph structures show all possible routes pedestrians can take to walk from their origins to their destinations by avoiding obstacles. Figure 7 shows a visibility graph for such an exemplary scenario. The movement space is restricted by the mesoscopic model if the visibility graph of the microscopic model contains more edges and therefore more walking paths. This means, that the mesoscopic model has less accessible pathways than the microscopic model. Therefore, all areas which are not accessible in the mesoscopic model have to be simulated with a microscopic approach. For such combined approaches, hybrid models should be used (Ijaz et al., 2015). However, mesoscopic areas with the same amount of edges like the microscopic visibility graph can be simulated by mesoscopic models.

5 Conclusion and Outlook

In this paper, we studied the use of different spatial scales in the context of pedestrian dynamics simulations. We showed a classification for macroscopic, mesoscopic and microscopic scenarios: a distinction between macroscopic and non-macroscopic is possible by considering

²⁷This Figure originates from the MomenTUMv2 simulator (Kielar et al., 2016).

the one-dimensionality of a scenario, whereas mesoscopic and microscopic scenarios can be distinguished by local densities and movement restricting obstacles.

In the future we will improve the presented methods, since they currently have some shortcomings. For example, a more detailed investigation about microscopic pedestrian body shapes is necessary. Currently, we studied circular and elliptical shapes and neglected the influence of pedestrians' legs on the geometry of a microscopic pedestrian. Some pedestrian simulators exist, which use the influence of legs on the pedestrians' space requirements, e.g. Chraïbi et al., 2010. Furthermore, body shapes with more complex geometries exist, e.g. spheropolygon shaped torsos (Alonso-Marroquín et al., 2014). We will focus on these issues in future publications. Another issue is that our study was executed only by a small number of participants. A larger survey with more test cases and several hundreds of participants would be useful to verify the results of this experimental study.

Acknowledgments

We would like to thank all members of the MultikOSi project for their helpful discussions according to this topic. Furthermore, we would like to thank all participants, who took part at our experiment. This work is supported by the Federal Ministry for Education and Research (Bundesministerium für Bildung und Forschung, BMBF), project MultikOSi, under grant FKZ 13N12823.

References

- Alonso-Marroquín, F., Busch, J., Chiew, C., Lozano, C., and Ramírez-Gómez, Á. (2014). „Simulation of counterflow pedestrian dynamics using spheropolygons“. In: *Physical Review E* 90.6.
- Anh, N. T. N., Daniel, Z. J., Du, N. H., Drogoul, A., and An, V. D. (2011). „A hybrid macro-micro pedestrians evacuation model to speed up simulation in road networks“. In: *Advanced Agent Technology*. Springer, pp. 371–383.
- Biedermann, D. H., Torchiani, C., Kielar, P. M., Willems, D., Handel, O., Ruzika, S., and Borrmann, A. (2016). „A hybrid and multiscale approach to model and simulate mobility in the context of public event“. In: *Proc. of the mobil.TUM conference 2016*.
- Biedermann, D. H., Dietrich, F., Handel, O., Kielar, P. M., and Seitz, M. (2015). *Using Raspberry Pi for scientific video observation of pedestrians during a music festival*. Technical Report.
- Biedermann, D. H., Kielar, P. M., Handel, O., and Borrmann, A. (2014). „Towards TransiTUM: A generic framework for multiscale coupling of pedestrian simulation models based on transition zones“. In: *Transportation Research Procedia* 2, pp. 495–500.
- Blue, V. J. and Adler, J. L. (2001). „Cellular automata microsimulation for modeling bi-directional pedestrian walkways“. In: *Transportation Research Part B: Methodological* 35.3, pp. 293–312.
- Chang, H.-C. and Wang, L.-C. (2010). „A Simple Proof of Thue's Theorem on Circle Packing“. In: *arXiv preprint arXiv:1009.4322*.
- Chraïbi, M., Seyfried, A., and Schadschneider, A. (2010). „Generalized centrifugal-force model for pedestrian dynamics“. In: *Physical Review E* 82.4.
- Colombo, R. M. and Rosini, M. D. (2005). „Pedestrian flows and non-classical shocks“. In: *Mathematical Methods in the Applied Sciences* 28.13, pp. 1553–1567.

Is macroscopic resolution enough? – A study about the use of different spatial scales in pedestrian dynamics

- Duives, D. C., Daamen, W., and Hoogendoorn, S. P. (2015). „Quantification of the level of crowdedness for pedestrian movements“. In: *Physica A: Statistical Mechanics and its Applications* 427, pp. 162–180.
- Edie, L. C. (1963). *Discussion of traffic stream measurements and definitions*. Port of New York Authority.
- Guises, R., Xiang, J., Latham, J.-P., and Munjiza, A. (2009). „Granular packing: numerical simulation and the characterisation of the effect of particle shape“. In: *Granular Matter* 11.5, pp. 281–292.
- Helbing, D. and Molnar, P. (1995). „Social force model for pedestrian dynamics“. In: *Physical review E* 51.5.
- Ijaz, K., Sohail, S., and Hashish, S. (2015). „A survey of latest approaches for crowd simulation and modeling using hybrid techniques“. In: *17th UKSIMAMSS International Conference on Modelling and Simulation*, pp. 111–116.
- Kielar, P. M., Biedermann, D. H., and Borrmann, A. (2016). *MomentUMv2: a modular, extensible, and generic agent-based pedestrian behavior simulation framework*. Tech. rep. Technische Universität München.
- Kneidl, A., Borrmann, A., and Hartmann, D. (2012). „Generation and use of sparse navigation graphs for microscopic pedestrian simulation models“. In: *Advanced Engineering Informatics* 26.4, pp. 669–680.
- Leonard, J. D. and Ni, D. (2006). „Direct Methods of Determining Traffic Stream Characteristics by Definition“. In: *Annual Meeting of the Transportation Research Board* 6.241.
- Lighthill, M. J. and Whitham, G. B. (1955). „On kinematic waves. II. A theory of traffic flow on long crowded roads“. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 229. 1178, pp. 317–345.
- Mayer, H., Klein, W., Frey, C., Daum, S., Kielar, P., and Borrmann, A. (2014). „Pedestrian simulation based on BIM data“. In: *Building Simulation Conference*.
- Oberhagemann, D. (2012). *Static and dynamic crowd densities at major public events*. Technical Report.
- Rabiaa, C. and Foudil, C. „Toward a Hybrid Approach for Crowd Simulation“. In: *International Journal of Advanced Computer Science & Applications* 1.7, pp. 52–61.
- Richards, P. I. (1956). „Shock waves on the highway“. In: *Operations research* 4.1, pp. 42–51.
- Steffen, B. and Seyfried, A. (2010). „Methods for measuring pedestrian density, flow, speed and direction with minimal scatter“. In: *Physica A: Statistical mechanics and its applications* 389.9, pp. 1902–1910.
- Tarr, M. J. (2000). *Visual pattern recognition*. American Psychological Association.
- Weidmann, U. (1992). *Transporttechnik der Fussgänger*. Schriftenreihe des IVT.

Vergleich verschiedener Fingerprint-Berechnungsverfahren zur Standortbestimmung innerhalb von Gebäuden

Marcus Dombois

*Institut für Numerische Methoden und Informatik im Bauwesen, Technische Universität
Darmstadt, Franziska-Braun-Str. 7 64287 Darmstadt, Deutschland
E-Mail: dombois@iib.tu-darmstadt.de*

Abstract Die Nutzung von Standortinformationen speziell innerhalb von Gebäuden hat in den letzten Jahren und verstärkt mit der Vorstellung der Bluetooth-basierten Apple iBeacons ein hohes Interesse erfahren. Gerade im Bereich des Facility Managements zeigt sich zum Beispiel bei der Navigationsunterstützung von Handwerkern während der Konstruktions- oder Nutzungsphase des Gebäudes ein großes Potenzial durch die Lokalisierung der Nutzer. Bei der Entwicklung von Systemen zur Standortbestimmung wird dabei häufig auf die bereits vorhandene drahtlose Infrastruktur (Wi-Fi) oder energieeffiziente Bluetooth Low Energy-Beacons zurückgegriffen. Aufgrund ihrer hohen Genauigkeit und Stabilität werden bei diesen signalbasierten Sensoren oft Musterabgleichsverfahren, allen voran das Fingerprinting, für die Berechnung der Position der Nutzer verwendet. In dieser Arbeit werden, basierend auf der Definition eines Fingerprints als mathematischer Vektor, insgesamt elf unterschiedliche Berechnungsverfahren untersucht und miteinander verglichen. Dabei werden sowohl „Nearest Neighbor“-Verfahren betrachtet, welche die Euklidische Distanz zweier Vektoren berechnen, als auch Verfahren, die die Kosinus-Ähnlichkeit und somit den Winkel zwischen zwei Vektoren vergleichen. Darüber hinaus werden weitere Verfahren betrachtet, die andere Ähnlichkeitsmaße verwenden und häufig im Kontext des „Data-Mining“ Anwendung finden. Die verschiedenen Verfahren wurden im Zuge einer umfangreichen Evaluation in einem Bürokomplex bei laufendem Betrieb getestet und verglichen. Dabei wurden zwei Smartphones für die Datensammlung verwendet und die Verfahren in unterschiedlichen Infrastrukturbedingungen analysiert. Die Evaluationsergebnisse zeigen, dass es teilweise erhebliche Unterschiede in der Genauigkeit der Berechnung gibt und veranschaulichen, dass das „Adapted Nearest Neighbor“-Verfahren das größte Potenzial bietet und wie die Berechnungsgenauigkeit noch weiter optimiert werden kann.

Keywords: Standortbestimmung innerhalb von Gebäuden, Fingerprinting, Euklidische Distanz, Kosinus Ähnlichkeit

Einführung

Die Verwendung von Standortinformationen für z.B. Navigationsapplikationen oder ortsbasierten Empfehlungen auf Smartphones hat in den letzten Jahren deutlich an Bedeutung gewonnen. Das Forschungsinteresse liegt durch die vorhandenen Limitierungen der Positionierung über

GPS und Funknetz-Zellen dabei hauptsächlich bei der Ortsbestimmung innerhalb von Gebäuden. Auch wenn die Anwendungen vielfältig sind, birgt die Nutzung von Standortdaten für das Facility Management mit das größte Potenzial. So können durch die Personenlokalisierung beispielsweise intelligente Raumbuchungskonzepte zur Anwendung kommen. Auch sind mithilfe des Standorts innovative Zutrittssysteme denkbar. Des Weiteren sind ortsbasierte Empfehlungen auf Smartphones oder Wearables bzw. Navigationsanwendung für das Facility Management interessante Anwendungsgebiete. Speziellen Navigationsszenarien, wie zum Beispiel die Navigation von Rettungskräften im Brandfall oder der Navigation von Handwerkern in der Konstruktionsphase kommt dabei eine besondere Bedeutung zu.

Aus den genannten Beispielen wird ersichtlich, dass die Lokalisierung von Personen speziell in Gebäuden zu einer Optimierung verschiedener Prozesse, insbesondere im Bereich der Facility Services (alle Dienstleistungen zur Unterstützung der Hauptaktivitäten einer Organisation) führen kann. Für die Standortbestimmung sind vor allem Smartphones interessant, da sie mittlerweile über eine Vielzahl von Sensoren verfügen und von den meisten Personen stetig mitgeführt werden.

Eine der in bisherigen Forschungsarbeiten gängigsten Methoden zur Ermittlung des Standortes in Gebäuden ist das Fingerprinting. Dabei handelt es sich um ein Musterabgleichverfahren, das die dem Nutzer nächste Position aus einer Menge bekannter Positionen berechnet. In diesem Paper werden verschiedene Fingerprint-Berechnungsverfahren untersucht und miteinander verglichen. Dabei werden sowohl der häufig eingesetzte Nearest Neighbor-Algorithmus und die Kosinus-Ähnlichkeit betrachtet als auch Algorithmen aus dem Bereich des Data-Mining. Die Unterschiede in der Qualität der Berechnungsverfahren werden in einer ausführlichen Evaluation ermittelt.

Der verbleibende Teil der Arbeit untergliedert sich in vier Abschnitte: im Abschnitt „Berechnungsverfahren zur Standortbestimmung“ wird zunächst eine Einführung in die verschiedenen Berechnungsverfahren gegeben und ihr Einsatz anhand von vergleichbaren Forschungsarbeiten aufgezeigt. Im darauffolgenden Abschnitt wird der Schwerpunkt auf das Fingerprinting gesetzt und die einzelnen Algorithmen gegenübergestellt und erläutert. Der vierte Abschnitt beschreibt den Aufbau und die Durchführung der Evaluation und vergleicht anschaulich die Ergebnisse der verschiedenen Berechnungsverfahren. Im letzten Abschnitt werden schließlich die Ergebnisse zusammengefasst und ein Fazit formuliert. Weiterhin wird ein Ausblick in weitergehende Forschungsfragen gegeben.

Berechnungsverfahren zur Standortbestimmung

Die Berechnung der Standorte der Nutzer kann mit verschiedenen Verfahren erfolgen. Die Basis für alle Verfahren bilden die gewonnenen Sensordaten. Für die Standortbestimmung innerhalb von Gebäuden mithilfe von Smartphones werden typischerweise Wi-Fi und Bluetooth bzw. Bluetooth Low Energy sowie die Inertialsensoren (Beschleunigungssensor, Magnetometer und Gyroskop) für die Berechnung herangezogen. Teilweise kommen auch weitere Sensoren, wie Schrittzähler oder Drucksensoren zum Einsatz. Zu den Berechnungsverfahren zählen: die Nachbarschaftsdetektion, distanzbasierte Verfahren, winkelbasierte Verfahren und der Musterabgleich. Nachfolgend werden die einzelnen Verfahren beschrieben und ihr Einsatz in vergleichbaren Forschungsarbeiten aufgezeigt. Der Vollständigkeit halber sei als weitere Berechnungsmöglichkeit auch das Kopplungsverfahren genannt. Die Position wird bei diesem

Verfahren nicht über Funktechnologien bestimmt, sondern über die im mobilen Endgerät vorhandenen Bewegungssensoren. Kopplungsverfahren werden häufig in Kombination mit einem der zuvor genannten eingesetzt.

Nachbarschaftsdetektion / Proximity Sensing

Die Nachbarschaftsdetektion ist eines der einfachsten Verfahren zur Positionsbestimmung. Sie kommt schon länger bei der Lokalisierung über das „Global System for Mobile Communications“ (GSM) zum Einsatz („Cell of Origin“-Prinzip), wird jedoch auch im stark wachsenden Markt der Bluetooth Beacons bzw. iBeacons häufig eingesetzt. Während die Nachbarschaftsdetektion beim GSM vor allem im Außenbereich stattfindet, wird sie bei Bluetooth Beacons bzw. iBeacons im Gebäude angewendet.

Bei der Nachbarschaftsdetektion mit Wi-Fi und Bluetooth Low Energy wird die Position des Nutzers mit der bekannten Position des Beacons bzw. des Access Points gleichgesetzt zu dem die beste Signalstärke gemessen wird (bestenfalls RSSI-Wert) (Henniges, 2012). La Delfa und Catania (2014) verwenden das Verfahren in ihrer Arbeit zur Detektion des Raumes in dem sich der Nutzer befindet. Dabei verwenden sie Bluetooth Low Energy Beacons die in jedem Raum platziert sind. Die Autoren argumentieren, dass die Raumgenauigkeit für die meisten Anwendungsszenarien ausreichend ist und empfehlen den Einsatz von optischen Markierungen, sofern eine genauere Navigation gefordert sein sollte.

Distanzbasierte Verfahren

Distanzbasierte Verfahren arbeiten typischerweise in zwei Schritten. Sie ermitteln zunächst die Distanz zu mehreren Sendern mit bekannten Standorten über zeitbasierte (z.B. „Time of Arrival“) oder signalstärkenbasierte Verfahren und berechnen anschließend aus diesen Entfernungen ihre eigene Position über mathematische Verfahren wie die Trilateration, Multilateration, Methode der kleinsten Quadrate oder Multidimensionale Skalierung. Bei der Trilateration wird mit Hilfe der Radien zu bekannten Punkten und ihren Koordinaten ein lineares Gleichungssystem aufgestellt und durch dessen Lösung die Koordinaten des aktuellen Standpunktes des Benutzers berechnet. Damit das Gleichungssystem eindeutig lösbar ist, werden für einen zweidimensionalen Punkt drei bekannte Punkte benötigt, für einen dreidimensionalen Punkt dementsprechend vier. Sollen mehr Punkte in die Berechnung einfließen (Multilateration), so kann das Ergebnis u.a. durch Iteration bzw. Mittelwertbildung von mehreren Trilaterationen bestimmt werden. Aufgrund von Ungenauigkeiten in der Bestimmung der Radien kann oft nur eine Region und kein eindeutiger Punkt ermittelt werden (Galvan-Tejada et al., 2012; Henniges, 2012).

Die Berechnung der Distanz aus der Signalstärke, wie sie auch in der Arbeit von Galvan-Tejada et al. (2012) mit den Technologien Wi-Fi und Bluetooth umgesetzt wird, nutzt das Phänomen des Ausbreitungsverlustes zur Entfernungsberechnung. Dieses geht davon aus, dass sich die Signalstärke mit zunehmender Distanz verringert. Das Phänomen ist allerdings nur für die Idealbedingung einer störungsfreien Ausbreitung („line-of-sight“) ohne Einschränkung gültig. Durch stationäre Objekte (z.B. Wände o.ä.) oder durch dynamische Objekte (wie bspw. Menschen) kommt es in der Praxis daher zu teils starken Abweichungen der Signalstärke beim Empfänger (Henniges, 2012; Mahalik, 2007).

Winkelbasierte Verfahren

Bei der Triangulation oder auch „Angle-of-Arrival“-Methode wird der Standort des Empfängers über die Winkel, die der Empfänger mit den umgebenen Sendern aufspannt, berechnet. Die Ermittlung der Position ist vergleichbar mit der „Time Difference of Arrival“-Methode. Die Berechnung des Winkels zwischen Sender und Empfänger erfolgt nach dem folgenden Ablauf: Der Sender sendet ein Signal aus, welches auf den Empfänger trifft und von diesem zum Sender zurückgesendet wird. Das zurückgesendete Signal wird vom Sender über eine spezielle Antenne empfangen. Diese Antenne ist in der Lage zu detektieren an welcher Stelle das Signal zu welcher Zeit auftritt. Mit diesen Informationen kann der Sender dann die Richtung bzw. den Winkel zum Empfänger berechnen. Die Antenne ist somit in der Lage, die Phasenverschiebung des Signals zu detektieren. Durch die spezielle Hardware ist die Positionsbestimmung üblicherweise nur auf Seiten des Senders möglich, da die meisten mobilen Empfänger (insbesondere Smartphones) nicht über die entsprechenden Voraussetzungen verfügen. Das System „ArrayTrack“, das in der Arbeit von Xiong und Jamieson (2013) beschrieben ist, verwendet das winkelbasierte Verfahren mit Wi-Fi Access Points zur Bestimmung der Standorte der Nutzer. Die Autoren argumentieren, dass es einen allgemeinen Trend zum Einsatz von multiplen Antennen bzw. solchen, die die Phasenverschiebung messen können gibt, um die Kapazität und die Erreichbarkeit der Netze zu erhöhen (Henniges, 2012; Xiong und Jamieson, 2013).

Musterabgleich

Zu den Verfahren des Musterabgleichs zählen alle diejenigen Verfahren, die in dem Bereich, in dem eine Positionsbestimmung erfolgen soll, eine Menge von Referenzpunkten definieren, mit denen die aktuelle Position verglichen wird. Dazu zählen visuelle Muster wie zum Beispiel Barcodes oder QR-Codes (La Delfa und Catania, 2014) und Messmuster; das am häufigsten eingesetzte Verfahren mit Messmustern ist das sog. „Fingerprinting“. Beim Fingerprinting wird die Signalstärke zu allen im Bereich befindlichen Sendern gemessen und als „Fingerabdruck“/„Fingerprint“ der entsprechenden Position in einer Datenbank gespeichert. Dabei stellt ein Fingerprint einen Vektor mit RSSI-Werten zu jedem Sender in Reichweite dar. Dieser Vektor kann abhängig von den Systembedingungen technologieübergreifend sein (zum Beispiel von Wi-Fi Access Points und Bluetooth LE Beacons) oder technologiegetrennt aufgenommen werden. Musterabgleichs-Verfahren arbeiten zweistufig: In der Offline-Phase (bzw. Kalibrierungsphase oder Trainingsphase) wird das Muster erstellt; in der Online-Phase wird ein Abgleich der aktuell gemessenen Daten mit dem Muster vorgenommen (Henniges, 2012; Mahalik, 2007).

In der Offline-Phase werden Fingerprints an allen vordefinierten Standorten aufgenommen und in einer Datenbank gespeichert. Dabei muss zwar der entsprechende Referenzpunkt eindeutig definiert sein, Kenntnisse über den Standort der Sender im entsprechenden Bereich sind jedoch nicht erforderlich (Henniges, 2012; Mahalik, 2007).

In der Online-Phase werden dann, an einem zunächst unbekanntem Punkt, die relevanten Sensordaten aufgezeichnet und wiederum in einen Vektor konvertiert. Danach wird dieser Vektor mit allen vorab definierten Fingerprints in einem Musterabgleich verglichen und der, nach Maßgabe einer Abstandsfunktion, beste Fingerprint bzw. der entsprechende Referenzpunkt als aktueller Standort des Empfängers angenommen. Dabei ist die Genauigkeit des Verfahrens abhängig von der Dichte der Referenzpunkte und dem eingesetzten Berechnungsverfahren für

den Musterabgleich. Beispiele speziell für das Fingerprinting lassen sich in vergleichbaren Forschungsarbeiten viele finden. Beispielhafte Systeme, die Wi-Fi Fingerprinting in Kombination mit Kopplungsverfahren (vgl. nächster Abschnitt) verwenden sind z.B. in den Arbeiten von Hilsenbeck et al. (2014) und Radu und Marina (2013).

Analyse verschiedener Fingerprinting Verfahren

Von den im letzten Abschnitt beschriebenen Verfahren werden besonders häufig Musterabgleichverfahren speziell das Fingerprinting in Systemen zur Standortberechnung verwendet. Die Vorteile des Fingerprintings liegen in seiner hohen Genauigkeit und Stabilität, insbesondere für sich wenig verändernde Umgebungen. Bei einer hohen Dynamik im Detektionsbereich empfiehlt es sich, in wiederkehrenden Abständen Trainingsphasen durchzuführen. Ein weiterer Vorteil ist, dass vorhandene Hardware (Wi-Fi Infrastruktur, Smartphones) ohne Änderungen verwendet werden kann und weder die Standorte der Sender noch deren Sendestärke bekannt sein müssen. Darüber hinaus kann die Berechnung sowohl Client- als auch Serverseitig erfolgen und somit Bedenken bei der Privatsphäre eingedämmt werden. Größter Nachteil des Musterabgleichs ist die aufwendige Trainingsphase.

Im Verlauf der Arbeit wurden insgesamt 11 Fingerprint-Verfahren entwickelt bzw. aus der Literatur entnommen und im System implementiert. Um eindeutige Berechnungsergebnisse zu ermöglichen, setzen alle Verfahren mindestens drei Scanergebnisse voraus, um eine Position zu berechnen. Wenn diese Voraussetzung nicht erfüllt ist, wird der Scan verworfen und es kann keine Position bestimmt werden.

Mathematisch gesehen handelt es sich bei einem Fingerprint um ein Element eines m -dimensionalen Vektorraumes, wobei jede Dimension einen Access Point (im Falle von Wi-Fi) bzw. ein Beacon (im Falle von Bluetooth) repräsentiert. Die Werte der einzelnen Dimensionen entsprechen dem gemessenen RSSI-Wert des entsprechenden Access Points bzw. Beacons. Nicht im Scan vorhandene Access Points oder Beacons erhalten den Wert „0“ zugewiesen. Um zwei RSS-Vektoren bzw. Fingerprints im Vektorraum miteinander zu vergleichen werden in der Literatur häufig die Methoden der linearen Algebra verwendet. Eine Vergleichsmöglichkeit ist die Untersuchung der Länge des Verbindungsvektors zwischen den zwei Elementen, eine andere die Betrachtung des, durch die zwei RSS-Vektoren aufgespannten, Raumwinkels.

In Tabelle 1 ist eine Übersicht der in der Arbeit verwendeten Berechnungsverfahren gegeben. In der ersten Spalte der Tabelle findet sich der Name des jeweiligen Algorithmus, in der zweiten Spalte ist die jeweilige Berechnungsformel dargestellt. Die Formelnotationen ist dabei für alle Formeln identisch: $r_j^{(i)}$ bezeichnet den RSSI-Wert zum j -ten Access Point bzw. Beacon (von insgesamt m Access Points bzw. Beacons) am i -ten Fingerprint der Fingerprint-Datenbank (mit insgesamt n Fingerprints). s_j beschreibt den gescannten RSSI-Wert zum j -ten Access Point bzw. Beacon. Der match count beschreibt die Anzahl an Access Points bzw. Beacons, die sowohl in einem Fingerprint aus der Datenbank als auch im gescannten RSS-Vektor vorkommen. Die Algorithmen sind jeweils so dargestellt, dass der kleinste berechnete Wert stets das beste Ergebnis repräsentiert.

Der Ablauf innerhalb der Positionsberechnung ist, abgesehen von der Anwendung der unterschiedlichen Abstandsmaße, immer gleich: zunächst wird überprüft, ob die Liste mit den verarbeiteten Scanergebnissen eine Größe von mindestens drei hat. Ist dies der Fall, wird in einer Schleife jeder Referenz-Fingerprint aus der Datenbank mit dem gescannten Fingerprint verglichen und das Abstandsmaß berechnet. Nach der Prüfung, ob mindestens drei gleiche

Vergleich verschiedener Fingerprint-Berechnungsverfahren zur Standortbestimmung innerhalb von Gebäuden

Access Points bzw. Beacons sowohl im Fingerprint-Vektor der Datenbank als auch in dem Scannergebnis vorliegen, wird das Ergebnis in einer Liste gespeichert. Ist der eben beschriebene „Match Count“ nicht erfüllt, wird das Ergebnis mit dem Faktor 100 verschlechtert. Diese Gewichtung führt dazu, dass Ergebnisse mit hoher Unsicherheit künstlich verschlechtert und somit gute bzw. Resultate mit hoher Sicherheit indirekt unterstützt werden. Alternativ könnten die Werte, die den „Match Count“ nicht erfüllen auch aus Ergebnisliste entfernt werden. Am Ende der Berechnung liegt somit eine nach dem Abstandsmaß geordnete Liste mit den möglichen Positionen vor.

Tabelle 1: Übersicht der Fingerprint-Berechnungsverfahren

Algorithm	Equation
Nearest Neighbor	$\sqrt{\sum_{j=1}^m (r_j^{(i)} - s_j)^2}$
Weighted Nearest Neighbor	$weight * \sqrt{\sum_{j=1}^m (r_j^{(i)} - s_j)^2}$ $weight = \frac{m}{m + match\ count}$
Adapted Nearest Neighbor	$\sum_{j=1}^m c_j^{(i)\frac{1}{3}}$ $c_j^{(i)} = \begin{cases} r_j^{(i)} - s_j & \text{if } r_j^{(i)} - s_j \leq 20 \\ 20 & \text{if } r_j^{(i)} - s_j > 20 \end{cases}$
Cosine Similarity	$\frac{\sqrt{\sum_{j=1}^m r_j^{(i)} * \sum_{j=1}^m s_j}}{\sum_{j=1}^m (s_j * r_j^{(i)})}$
Weighted Cosine Similarity	$weight * \frac{\sqrt{\sum_{j=1}^m r_j^{(i)} * \sum_{j=1}^m s_j}}{\sum_{j=1}^m (s_j * r_j^{(i)})}$ $weight = \frac{m}{m + match\ count}$
Combined Cosine Similarity and Nearest Neighbor	$\ln \left(\sqrt{\sum_{j=1}^m (r_j^{(i)} - s_j)^2} \right) * \left(1 - \frac{\sum_{j=1}^m (s_j * r_j^{(i)})}{\sqrt{\sum_{j=1}^m r_j^{(i)} * \sum_{j=1}^m s_j}} \right)$
Combined Cosine Similarity and Adapted Nearest Neighbor	$\ln \left(\sum_{j=1}^m c_j^{(i)\frac{1}{3}} \right) * \left(1 - \frac{\sum_{j=1}^m (s_j * r_j^{(i)})}{\sqrt{\sum_{j=1}^m r_j^{(i)} * \sum_{j=1}^m s_j}} \right)$ $c_j^{(i)} = \begin{cases} r_j^{(i)} - s_j & \text{if } r_j^{(i)} - s_j \leq 20 \\ 20 & \text{if } r_j^{(i)} - s_j > 20 \end{cases}$
Pseudo Cosine Similarity	$\frac{\sum_{j=1}^m r_j^{(i)} * \sum_{j=1}^m s_j}{\sum_{j=1}^m (s_j * r_j^{(i)})}$
Dice Similarity	$\frac{\sum_{j=1}^m r_j^{(i)} + \sum_{j=1}^m s_j }{2 * \sum_{j=1}^m (s_j * r_j^{(i)})}$
Overlap Similarity	$\frac{\min \left(\sum_{j=1}^m s_j , \sum_{j=1}^m r_j^{(i)} \right)}{\sum_{j=1}^m \left(\min(s_j , r_j^{(i)}) \right)}$
Jaccard Similarity	$\frac{\sum_{j=1}^m (s_j * r_j^{(i)})}{\sum_{j=1}^m r_j^{(i)} + \sum_{j=1}^m s_j - \sum_{j=1}^m (s_j * r_j^{(i)})}$

Differenz der Vektorlängen („Nearest Neighbor“-Algorithmen)

Der in der existierenden Forschung zur Standortbestimmung am häufigsten Anwendung findende Algorithmus ist der „Nearest Neighbor“-Algorithmus. Er berechnet die Euklidische Distanz der beiden Vektoren. Darauf aufbauend wurde der „Weighted Nearest Neighbor“-Algorithmus

entwickelt, der ebenfalls die Euklidische Distanz berechnet, das Ergebnis allerdings abhängig von der Anzahl von übereinstimmend vorkommenden Access Points bzw. Beacons, künstlich verbessert. Der „Adapted Nearest Neighbor“-Algorithmus ist aufgrund der Untersuchungen von Lee et al. (2013) als Verfahren aufgenommen worden und basiert ebenfalls auf dem „Nearest Neighbor“-Algorithmus. Ausgehend von der Feststellung, dass bei unterschiedlicher Handhaltung, Ausrichtung oder Höhe des Smartphones sehr starke Schwankungen im RSSI-Wert auftreten, versucht der „Adapted Nearest Neighbor“-Algorithmus große Differenzen zwischen gemessenem Wert und Datenbankwert zu verringern. Dafür werden zwei Mechanismen verwendet. Zum einen wird eine maximale Differenz definiert, die alle größeren Differenzen gleichwertig behandelt. Zum anderen wird die Potenz, im Gegensatz zu Euklid, auf einen Wert kleiner eins verringert. Beide Mechanismen sollen den Einfluss schlechter RSSI-Werte verringern und geringe Differenzen unterstützen. Dadurch soll ein insgesamt besseres Ergebnis erzielt werden. Alle drei Verfahren berechnen jeweils die Längendifferenzen zwischen dem Fingerprint-Vektor aus der Datenbank und dem vom Sensor gemessenen Fingerprint-Vektor.

Vergleich der Raumwinkel (Kosinus-Ähnlichkeit)

Bei den Berechnungsverfahren „Cosine Similarity“ und „Weighted Cosine Similarity“ handelt es sich um Verfahren, die die Kosinus-Ähnlichkeit zweier Fingerprint-Vektoren berechnen. Han et al. (2015) argumentieren, dass ein Schwachpunkt des „Nearest Neighbor“-Verfahrens die Fehleranfälligkeit beim Einsatz von verschiedenen Geräten ist. Sie merken an, dass sich die auftretenden RSSI-Variationen gleichermaßen auf alle Access Points bzw. Beacons auswirken und es dadurch zu erheblichen Unterschieden in den Euklidischen Distanzen kommt. Durch die Berechnung der Kosinus-Ähnlichkeit werden Verschiebungen durch den Einsatz unterschiedlicher Geräte kompensiert, da sich der Winkel der Fingerprint-Vektoren zueinander nicht ändert. Gerade bei einem Crowd-Sourcing Ansatz ist der Einsatz unterschiedlicher Geräte unvermeidbar und die Implementierung von Verfahren zur Berechnung über den Kosinus sinnvoll. Für die „Weighted Cosine Similarity“-Strategie wird äquivalent zum „Weighted Nearest Neighbor“ eine Gewichtung auf die Ergebnisse angewendet, die die Anzahl von übereinstimmend vorkommenden Access Points bzw. Beacons mit in die Bewertung einbezieht.

Kombination von Vektorlänge und Raumwinkel

Mit dem „Combined Cosine Similarity and Nearest Neighbor“ und dem „Combined Cosine Similarity and Adapted Nearest Neighbor“ wurden zwei Verfahren entwickelt, die die beiden zuvor beschriebenen Möglichkeiten, Vektoren zu vergleichen, kombinieren. Für die Kombination werden sowohl die Distanz als auch der Kosinus nach den oben beschriebenen Verfahren berechnet. Im Anschluss wird der natürliche Logarithmus der berechneten Distanz multipliziert mit der Differenz aus eins und dem berechneten Kosinus-Wert. Dabei wird vorausgesetzt, dass die Kosinus-Ähnlichkeit einen Wert zwischen 0 und 1 annimmt. Da der berechnete Distanzwert theoretisch jede Zahl größer 0 annehmen kann, werden durch die Verwendung des Logarithmus die beiden Teilergebnisse vergleichbarer gemacht.

Weitere Ähnlichkeitsmaße

Neben der Kosinus-Ähnlichkeit gibt es viele weitere Ähnlichkeitsmaße, die häufig im Kontext des „Data-Mining“ verwendet werden. Ferber (2003) stellt neben der Kosinus-Ähnlichkeit auch die Pseudo-Kosinus-Ähnlichkeit, die Dice-Ähnlichkeit, die Overlap-Ähnlichkeit und die Jaccard-Ähnlichkeit vor. Alle Verfahren sind theoretisch und praktisch evaluiert und werden zum Beispiel bei der Dokumentensuche eingesetzt. Bei der Untersuchung der verschiedenen Verfahren zur Positionsbestimmung, sollten sie deshalb nicht außen vor gelassen werden, da sie, wie schon die zuvor genannten Verfahren, auf dem Vektorraummodell-Ansatz aufbauen.

Evaluation

Um eine gute Analyse über einzelne Verfahren zur Standortbestimmung durchführen zu können, ist es wichtig, dass Evaluationen in realistischen Szenarien durchgeführt werden. Um diesem Anspruch gerecht zu werden, wurde die in dieser Arbeit umgesetzte Evaluation in einem großen Bürokomplex während der normalen Betriebszeiten durchgeführt. Dies führte dazu, dass es im Evaluationsbereich viele dynamische Störquellen gab, die von den verschiedenen Fingerprinting-Verfahren möglichst kompensiert werden mussten. Im folgenden Abschnitt werden sowohl der Aufbau als auch die Durchführung der Evaluation genauer beschrieben und im Anschluss die Ergebnisse aufgeführt. Dabei werden die einzelnen Fingerprinting-Verfahren miteinander verglichen und aufgetretene Ergebnisse diskutiert.

Die Evaluation fand über eine Periode von zwei Wochen statt. Das untersuchte Areal (ca. 2600 m^2) war gekennzeichnet durch ein offenes Bürokonzept mit vielen abgetrennten Bereichen, aber wenigen völlig abgeschlossenen Räumen wie in Abb. 1 anhand des Flurplans zu sehen. Neben großen Besprechungsräumen waren verschiedene kleinere Bereiche durch schwere Vorhänge und Stellwände von den Arbeitsbereichen getrennt. Büroräume, gekennzeichnet durch Tischgruppen von vier bis sechs Tischen, wurden mit Raumteilern, Pflanzen und Glaswänden voneinander getrennt. Im gesamten Areal war der Fußboden mit Teppich ausgelegt.

Für die Evaluation wurde ein virtuelles $2 \times 2\text{m}$ -Gitternetz definiert und über die gesamte Fläche gelegt. Damit wurden 4m^2 große Zellen erzeugt, in denen während der Evaluation die Referenzpunkte definiert wurden. Durch die Entfernung von evaluierter und berechneter Zelle wurde ein Vergleich der Verfahren ermöglicht. Für die praktische Anwendung ist ein virtuelles Gitternetz jedoch nicht erforderlich, da mit dem Fingerprinting-Verfahren auch zuvor erzeugte symbolische Standorte verwendet werden können und das Verfahren nicht an Koordinaten gebunden ist. Die Größe der Zellen wurde definiert über die zu erwartende Genauigkeit der Verfahren (2-4m). Im Evaluationsareal befanden sich 30 fest installierte Wi-Fi Access Points (2,4 und 5 GHz). Zusätzlich zu den vorhandenen Wi-Fi Access Points wurden 72 Bluetooth Low Energy Beacons platziert. Dies diente dem Zweck, den Vergleich der Fingerprinting-Verfahren auf Basis mehrerer Technologien zu prüfen und zusätzlich einen Vergleich zwischen den Technologien zu erreichen. Um speziell das Bluetooth Fingerprinting weitergehend zu untersuchen, wurde die Evaluationsfläche in zwei Bereiche aufgeteilt, die jeweils eine unterschiedlich hohe Dichte an Bluetooth Beacons aufwies (vgl. Abb. 1). Im ersten Bereich (grüne Umrandung) wurden Beacons alle vier bis fünf Meter platziert. Dabei wurde darauf geachtet, dass sich mindestens ein Beacon in jedem Büro und Besprechungsraum befand. Zusätzliche Beacons wurden an charakteristischen Wegpunkten wie zum Beispiel Garderoben und Treppenhauseingängen

platziert. Innerhalb der Büros und Besprechungsräume wurden die Beacons stets auf Tischen befestigt. Dabei sollte angemerkt werden, dass speziell mit den auf Tischen befestigten Beacons starke Signalschwankungen möglich waren, da sich die Tische in ihrer Höhe verstellen ließen. Im zweiten Teilbereich (blaue Umrandung) betrug die Beacon-Dichte acht bis zehn Meter. Die Beacons befanden sich ausschließlich in Fluren und auf Raumteilern und wurden nicht in spezifische Räume gelegt. Der Abstand zwischen den Beacons wurde in allen Bereichen so gewählt, dass eine gute Abdeckung und Überlappung der Signale gegeben war.

Fingerprints wurden an 108 Standorten im Evaluationsbereich gesammelt. Zusätzliche Fingerprints wurden darüber hinaus außerhalb des Areal aufgezichnet, um gezielt falsche Berechnungen möglich zu machen. Die Verteilung der so gesammelten Referenzpunkte hatte nicht zum Ziel das gesamte Gitternetz abzudecken sondern sollte vielmehr typische Wege von Personen innerhalb des Bereiches darstellen. Bei der Aufnahme der Referenzpunkte wurde der Evaluationsbereich ebenfalls in zwei Teilbereiche untergliedert und die Dichte der Referenzpunkte in den verschiedenen Teilbereichen variiert. Dies ermöglichte die Darstellung von zwei Stadien der Implementierung eines Systems zur Standortbestimmung. In Szenario 2 (vgl. Abb. 1 schwarze Umrandung) befindet sich das System noch im Aufbau und es existieren nur wenige Referenzpunkte. Im Gegensatz dazu liegt in Szenario 1 (rote Umrandung) ein vollständig einsatzbereites System vor und eine hohe Zahl an Referenzpunkten ist gegeben. An jedem Referenzpunkt (gelb markierte Zellen) wurden Fingerprints in vier Himmelsrichtungen aufgezeichnet. Dabei wurde das Gerät auf Hüfthöhe vor dem Körper gehalten. Die Aufnahme der Referenzpunkte erfolgte mit zwei Geräten, einem Sony Z3 Compact (Modell: D5803; Betriebssystem: Android 5.0) und einem Samsung S5 (Modell: SM-G900F; Betriebssystem: Android 5.0). Nach der Sammlung der Fingerprint-Daten konnten die Evaluationsdaten (rot gestrichelte Zellen) gesammelt werden; diese wurden aufgezeichnet indem ein Scan in allen Sensoren in einer Zelle im Gitternetz initiiert wurde. Dabei wurde nicht auf die Orientierung geachtet um eine möglichst reale Situation darzustellen. Während im Bereich von Szenario 1 Evaluationsdaten ausschließlich an Referenzpunkten gesammelt wurden, wurden in Szenario 2 auch an „unbekannten“ Standorten zwischen zwei Referenzpunkten Evaluationsdaten aufgezeichnet. Damit sollte das Berechnungsverhalten der einzelnen Verfahren untersucht werden, wenn sich der Nutzer nicht an einem bekannten Punkt befindet. Die Sammlung der Evaluationsdaten erfolgte ebenfalls mit den zuvor beschriebenen Geräten. Für die Evaluation standen am Ende drei Datenreihen für die Auswertung zur Verfügung:

1. „Z3 Compact“ Referenzdaten kombiniert mit „Z3 Compact“ Evaluationsdaten
2. „S5“ Referenzdaten kombiniert mit „S5“ Evaluationsdaten
3. „S5“ Referenzdaten kombiniert mit „Z3 Compact“ Evaluationsdaten

Um eine qualitative Aussage über die einzelnen Verfahren und Technologien treffen zu können und die Verfahren miteinander zu vergleichen, wurde für die Berechnung der Standorte nur ein RSSI-Filter angewendet. Dieser überprüft die Rohdaten eines Sensors und entfernt Einträge mit einer zu geringen Signalstärke. Dadurch wird vermieden, dass Berechnungsergebnisse zu starken Schwankungen unterliegen und das Berechnungsergebnis verfälscht wird. Der akzeptierte RSSI-Schwellwert liegt bei -90 für Wi-Fi Daten und bei -85 für Bluetooth Low Energy Daten. Weitere Filter oder Glättungsverfahren kamen nicht zum Einsatz.

Um die Güte der Berechnungsverfahren beurteilen zu können, wurde neben dem mittleren Fehler als weiteres Kriterium die Orientierungsentfernung eingeführt. Dabei handelt es sich um die bei der Berechnung maximal akzeptierte Entfernung, bei der davon ausgegangen werden

Vergleich verschiedener Fingerprint-Berechnungsverfahren zur Standortbestimmung innerhalb von Gebäuden

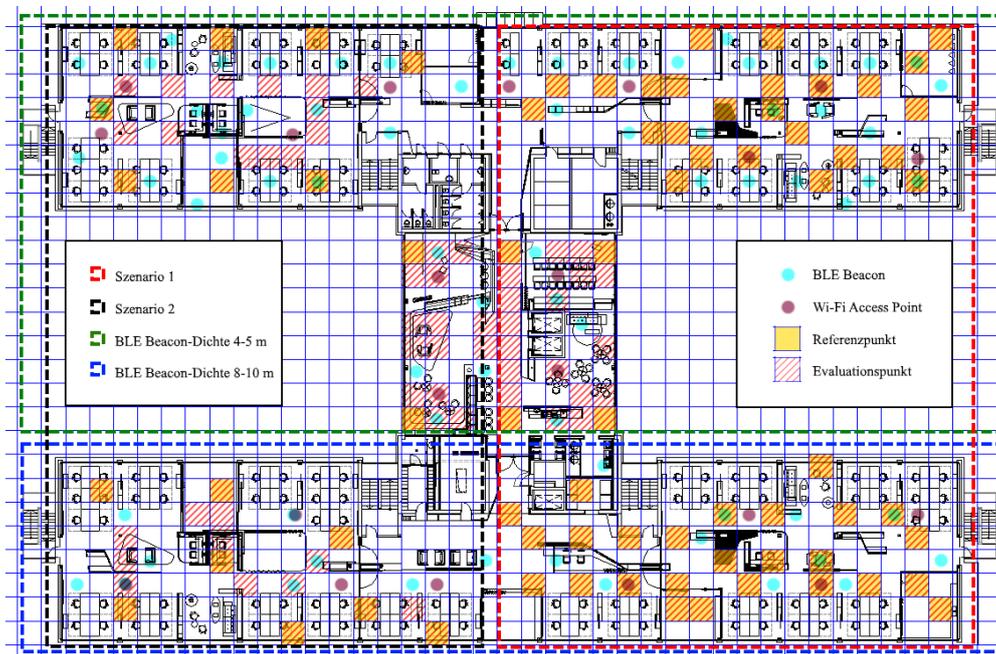


Abb. 1: Evaluationsbereich mit Infrastrukturen, Referenz- und Evaluationspunkten

kann, dass ein Nutzer sich orientieren und dementsprechend seinen wahren Standort ermitteln kann. Aus eigenen Erfahrungen und den Diskussionen mit potentiellen Nutzern wurde die Orientierungsentfernung auf 5m festgelegt.

Wie in Tabelle 2 anhand der Sony Z3 Compact Wi-Fi-Evaluationsdaten von Szenario 1 (erste Datenreihe) zu sehen, weist das „Adapted Nearest Neighbor“-Verfahren im Vergleich zu allen anderen die höchste Güte auf. Mit einem mittleren Fehler von 2,533m und 81,897% der Daten innerhalb der akzeptierten Entfernung, ist es das mit Abstand genaueste Verfahren. Die beiden Basisverfahren „Nearest Neighbor“ und „Cosine Similarity“ liegen mit ihren Ergebnissen im Mittelfeld. Da sie sich in ihren Ergebnissen nur minimal unterscheiden, kann im direkten Vergleich keine Präferenz zwischen distanzbasierten und winkelbasierten Verfahren identifiziert werden. Die „Overlap Similarity“, die bei den Wi-Fi-Evaluationsdaten das drittbeste Ergebnis erzielt, ist bei der Betrachtung der Bluetooth-Daten weit abgeschlagen mit dem zweitschlechtesten Ergebnis. Sowohl bei den Wi-Fi- als auch bei den Bluetooth-Daten erzielt die „Pseudo Cosine Similarity“ die schlechtesten Ergebnisse und somit die niedrigste Güte.

Wie zu erwarten, sind die Wi-Fi-Ergebnisse für Szenario 2 geringer als die zuvor beschriebenen (vgl. Tabelle 3). Dies lässt sich mit der deutlich geringeren Dichte an Referenzpunkten erklären. Auch hier bietet das „Adapted Nearest Neighbor“-Verfahren mit einem mittleren Fehler von 3,580m und 70,732% innerhalb der akzeptierten Entfernung die größte Genauigkeit. Das Verhältnis zu den anderen Verfahren ist wie im bereits betrachteten Szenario 1. Vergleicht man die Daten zwischen den beiden Technologien Wi-Fi und Bluetooth Low Energy, so stellt man fest, dass Bluetooth im Schnitt 15-20% schlechtere Ergebnisse erzielt. Die Differenz kann in Teilen über die ungünstige Position der Beacons im Vergleich zu den in der Decke angebrachten Access Points erklärt werden, deutet jedoch auch auf eine höhere Schwankung in den Bluetooth Scanergebnissen hin.

Tabelle 2: Wi-Fi-Evaluationsdaten für Szenario 1 (Sony Z3 Compact)

	Mittelwert [m]	Median [m]	Standard-abweichung [m]	≤ 5 m [%]
Adapted Nearest Neighbor	2,533	0,000	3,295	81,897
Nearest Neighbor	3,598	4,000	3,676	79,310
Weighted Nearest Neighbor	3,632	4,000	3,661	79,310
Cosine Similarity	3,573	4,000	3,821	77,586
Weighted Cosine Similarity	3,550	4,000	3,810	77,586
Combined Cosine Similarity and Nearest Neighbor	3,593	3,414	3,900	76,724
Combined Cosine Similarity and Adapted Nearest Neighbor	3,213	2,828	3,619	81,034
Pseudo Cosine Similarity	5,679	4,472	5,434	52,586
Dice Similarity	4,195	4,000	4,081	71,552
Overlap Similarity	3,425	2,828	3,602	74,138
Jaccard Similarity	4,195	4,000	4,081	71,552

Bezüglich der beiden Geräte konnte festgestellt werden, dass das Z3 Compact etwas bessere Ergebnisse für die Wi-Fi-Daten (ca. 3% besser) erzielen konnte, bei den Bluetooth-Daten jedoch minimal schlechter abschnitt (vgl. Tabelle 3) Es wird vermutet, dass die Differenz durch die unterschiedlichen Hardware-Sensoren und ihrer Position im Gerät zustande kommt. Daher kann davon ausgegangen werden, dass im praktischen Einsatz die Güte der Standortbestimmung bei der Verwendung einer Vielzahl unterschiedlicher Geräte eher abnimmt. Eine signifikante Abweichung konnte festgestellt werden wenn die Referenzdaten des Samsung S5 mit den Evaluationsdaten des Z3 Compact kombiniert wurden (dritte Datenreihe). Die Berechnung auf Basis von „fremden“ Referenzdaten führte stets zu deutlich verschlechterten Ergebnissen und sollte im praktischen Einsatz vermieden werden. An dieser Stelle konnten auch die Verfahren die mit der Kosinus-Ähnlichkeit arbeiten keinen deutlichen Mehrwert liefern. Das beste Verfahren war auch hier das „Adapted Nearest Neighbor“-Verfahren. Eine genauere Analyse zu dieser Thematik bleibt jedoch Thema von zukünftigen Arbeiten.

Die Betrachtung aller Berechnungsdaten zeigte, dass das erste Ergebnis in der sortierten Liste aller möglichen Ergebnisse eines Verfahrens nicht immer die beste Approximation des Standortes repräsentierte. Vielmehr konnte festgestellt werden, dass bei der Betrachtung mehrerer Einträge der Ergebnisliste, das Ergebnis, welches am häufigsten auftrat, im Mittel auch die bessere Approximation lieferte. Für die Wi-Fi-Daten des Z3 Compact (Szenario 1) konnte so festgestellt werden, dass bei der Betrachtung der ersten drei Berechnungsergebnisse die Güte merkbar steigt (mittlerer Fehler: 2,123m; akzeptierte Entfernung: 86,207%). Dabei wurde ein Ergebnis immer dann als richtig interpretiert, wenn es mehrfach ermittelt wurde oder, bei keinem mehrfach auftretenden Ergebnis, das erste der Liste war.

Vergleich verschiedener Fingerprint-Berechnungsverfahren zur Standortbestimmung innerhalb von Gebäuden

Tabelle 3: Auszug aus den verschiedenen Evaluationsergebnissen der unterschiedlichen Datenreihen

	Datenreihe / Szenario / Sensor	Mittelwert [m]	Median [m]	Standardabweichung [m]	≤ 5 m [%]
Adapted Nearest Neighbor	1 / Szenario 2 / Wi-Fi	3,580	2,828	3,300	70,732
Adapted Nearest Neighbor	1 / Szenario 1 / BLE	4,013	4,000	3,321	61,207
Adapted Nearest Neighbor	1 / Szenario 2 / BLE	5,164	4,000	6,352	58,537
Adapted Nearest Neighbor	2 / Szenario 1 / Wi-Fi	2,686	2,828	3,028	80,769
Adapted Nearest Neighbor	2 / Szenario 1 / BLE	4,071	4,000	3,295	63,265
Adapted Nearest Neighbor	3 / Szenario 1 / Wi-Fi	4,473	4,000	3,992	62,931
Nearest Neighbor	3 / Szenario 1 / Wi-Fi	5,065	4,472	4,364	60,345
Cosine Similarity	3 / Szenario 1 / Wi-Fi	5,060	4,472	4,317	59,483

Fazit und Ausblick

Im Rahmen dieser Arbeit wurden verschiedene Fingerprinting-Berechnungsverfahren zur Bestimmung des Standortes von Personen in Gebäuden untersucht und miteinander verglichen. Dabei wurde zunächst erläutert wie die Positionsbestimmung mittels Musterabgleichsverfahren funktioniert und im Anschluss auf die einzelnen Algorithmen-Gruppen und ihre Besonderheiten eingegangen. In einer großflächig angelegten Evaluation in einem Bürogebäude wurden die einzelnen Verfahren während des laufenden Betriebes getestet und evaluiert. Die Evaluation zeigte, dass das „Adapted Nearest Neighbor“ Verfahren, welches versucht die Einflüsse schlechter Scannergebnisse zu reduzieren, die besten Ergebnisse erzielt. Der Vergleich der beiden Technologien Wi-Fi und Bluetooth Low Energy zeigte weiterhin, dass Bluetooth ohne spezielle Filter und Glättungsverfahren nicht an die Güte von Wi-Fi herankommt. Beim Vergleich unterschiedlicher Endgeräte konnte außerdem festgestellt werden, dass mit zunehmender Anzahl an Geräten die Berechnungsgüte abnimmt. Hier zeigt sich ein großes Potenzial für zukünftige Arbeiten, welche sich mit der Frage der Integration von multiplen Geräten in Systeme zur Standortbestimmung befassen. Auch sollte weitergehend untersucht werden, wie sich die Genauigkeit der Standortberechnung durch Betrachtung der gesamten Ergebnismenge verbessern lässt.

Literatur

- Ferber, R. (2003). *Information Retrieval: Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. dpunkt-Verlag.
- Galvan-Tejada, I., Sandoval, E. I. und Brena, R. (2012). „Wifi bluetooth based combined positioning algorithm“. In: *Procedia Engineering* 35, S. 101–108. DOI: 10.1016/j.proeng.2012.04.170.
- Han, S., Zhao, C., Meng, W. und Li, C. (2015). „Cosine similarity based fingerprinting algorithm in WLAN indoor positioning against device diversity“. In: *2015 IEEE International Conference on Communications (ICC)*. (8.–12. Juni 2015).
- Henniges, R. (2012). „Current approaches of Wi-Fi positioning“. In: *Servicecentric Networking-Seminar WS 2011/2012*. (TU Berlin).
- Hilsenbeck, S., Bobkov, D., Schroth, G., Huitl, R. und Steinbach, E. (2014). „Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning“. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. (Seattle, Washington), S. 147–158.
- La Delfa, G. C. und Catania, V. (2014). „Accurate indoor navigation using smartphone, bluetooth low energy and visual tags“. In: *Proceedings of the 2nd Conference on Mobile and Information Technologies in Medicine*.
- Lee, J.-Y., Yoon, C.-H., Park, H. und So, J. (2013). „Analysis of location estimation algorithms for wifi fingerprint-based indoor localization“. In: *Proceedings of the 2nd International Conference on Software Technology*.
- Mahalik, N. P. (2007). *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications*. Berlin, Heidelberg: Springer.
- Radu, V. und Marina, M. K. (2013). „Himloc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced wifi fingerprinting“. In: *International Conference on Indoor Positioning and Indoor Navigation (IPIN) 2013*. (28.–31. Okt. 2013).
- Xiong, J. und Jamieson, K. (2013). „ArrayTrack: a fine-grained indoor location system“. In: *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*.

Parametrisierte Architekturzeichnungen: Architektonische Entwurfsabsicht durch parametrisierte Koordinatenabhängigkeiten beschreiben und darstellen

Dominik Ganghofer

*AG Mathematical Geometry Processing, Freie Universität Berlin, Arnimallee 6, 14195 Berlin
E-Mail: dominik.ganghofer@fu-berlin.de*

Abstract *Computer-Aided Design-Systeme* sind in der architektonischen Praxis mittlerweile allgegenwärtig. Dabei sind sie jedoch weniger Designhilfe als vielmehr Geometriemodellierung und Dokumentationswerkzeug. Sie bieten nur ungenügende Möglichkeiten, eine Entwurfsabsicht zu beschreiben. Vielmehr werden sie dazu genutzt, Geometrie zu modellieren, die das statische Ergebnis eines Entwurfs ist. Somit herrscht noch immer eine große Diskrepanz zwischen der Art und Weise, wie Gebäudegeometrien erdacht und sprachlich beschrieben werden und dem, was digitale Modelle letztlich abbilden. Die geometrischen Abhängigkeiten, die zu einer bestimmten Gestalt geführt haben, bilden den Kern des architektonischen Entwurfs. Diese Arbeit beschreibt eine Methode, diese Abhängigkeiten durch eine geeignete Parametrisierung im Modell abzubilden. Dabei wird das Modell als Ergebnis der verwendeten Modellierungsoperationen in einem orthonormalen Koordinatensystem beschrieben. Nachträgliche Änderungen der Geometrie werden durch Anpassung des Konstruktionsgraphen realisiert. Anschließend beschreiben wir eine grafische Nutzeroberfläche, die Analyse und Änderungen auch an umfangreichen parametrisierten Modellen gewährleistet.

Bei der folgenden Arbeit handelt es sich um ein Auszug aus dem meiner Bachelorarbeit (Ganghofer, 2016) im Fach Informatik bei Professor Polthier (AG Mathematical Geometry Processing) an der Freien Universität Berlin. Die Arbeit wurde 15. August 2016 eingereicht und am 22. August 2016 öffentlich vorgestellt.

Keywords: Entwurfsabsicht, Design Intent, Parametric Modelling, Computer-Aided Design

1 Einleitung

Problembeschreibung

CAD-Systeme werden in der architektonischen Praxis entgegen ihrer Bezeichnung weniger als Entwurfswerkzeuge sondern in erster Linie zur Geometriemodellierung und Dokumentation verwendet. Es stellt sich die Frage, ob es möglich ist CAD-Systeme so zu gestalten, dass sie den Anforderungen des Entwurfsprozesses besser gerecht werden und diesen aktiv unterstützen können. Ein Aspekt der Gestaltung solcher Systeme ist die Beschreibung der Geometrie im System: Architekten entwickeln Gebäudegeometrien meist durch das sukzessive Hinzufügen

Parametrisierte Architekturzeichnungen: Architektonische Entwurfsabsicht durch parametrisierte Koordinatenabhängigkeiten beschreiben und darstellen

geometrischer Objekte unter Bezugnahme auf bereits bestehende Geometrien und die Definition expliziter Distanzparameter. CAD-Systeme beschreiben die erzeugten Geometrien jedoch meist nur in Form globaler Koordinaten. Das Wissen über die verwendeten Bezugnahmen geht dabei verloren. Unterschiedliche Entwurfsprozesse können die selbe Geometrie erzeugen. Aus der resultierenden Geometrie lassen sich solche Abhängigkeiten nicht mehr ableiten. Diese Arbeit möchte ein System entwickeln, das neben der Gebäudegeometrie auch die Abhängigkeiten und Parameter, die zu dieser geführt haben, abbildet. Die entsprechend erzeugten Modelle erlauben es, neben der Geometrie auch zentrale Aspekte der architektonischen Entwurfsabsicht abzubilden und zwischen verschiedenen Akteuren auszutauschen. Dadurch werden vor allem anschließende Änderungen am Entwurf erleichtert.

Entwurfsabsicht abbilden

Iyer und Mills haben in einer Übersicht (Iyer und Mills, 2006) Arbeiten zur Abbildung der Entwurfsabsicht im Kontext von 2D-CAD-Zeichnungen gegenübergestellt. Sie definieren dabei die Entwurfsabsicht als Wissen, das den Entwurfsprozess dokumentiert, Gründe und Alternativen für Entwurfsentscheidungen abbildet und die Charakteristik und Beziehungen der erstellten geometrischen Objekte beschreibt. Diese Arbeit möchte nun ein Werkzeug entwickeln, das einen Teil der so definierten Entwurfsabsicht abbildet. Dabei beschränken wir uns darauf, die Charakteristik und Beziehungen der geometrischen Objekte sowie deren Konstruktionsreihenfolge zu beschreiben. Auf die Beschreibung von Entwurfsentscheidungen, Gründen und Alternativen wird verzichtet. Der Entwurf ist ein oftmals sehr intuitiver Prozess, so dass es schwer möglich und wenig sinnvoll ist, alle Entscheidungen formal zu begründen.

Beispiel

In Abb. 2 soll die Problematik anhand des Entwurfs eines exemplarischen, orthogonalen Grundrisses veranschaulicht werden. Zunächst beschreibt der Entwerfende ein Grundstück von 10,00 m Breite und 6,00 m Höhe. Entlang der Grundstückskante befinden sich Wände, deren Wandstärke von 0,50 m sich nach innen erstreckt. Nun werden Türen definiert: Deren Breite von 1,00 m ergibt sich aus Baunormen beziehungsweise aus verfügbaren Industrieformaten, ihre Stärke aus der Dimensionierung der Wand. Aus funktionalen Gründen wird sie möglichst in der Ecke des Raums positioniert. Der Abstand von 0,60 m zur Wand ist notwendig um das Öffnen der Tür zu gewährleisten. Abschließend wird der Innenraum in der Mitte in zwei gleich große Partitionen unterteilt. Durch unterschiedlichen Bodenbelag sollen zwei funktionale Zonen definiert werden. Diese Überlegungen, die zur Geometrie des Grundrisses geführt haben, werden in klassischen CAD-Systemen jedoch nicht dargestellt. Soll nun zum Beispiel die Breite des Grundstücks verändert werden, ist nicht definiert, wo die Tür anschließend positioniert werden soll. Muss die Wandstärke erhöht werden, muss erneut abgeklärt werden, in welche Richtung der Wand sich diese ausdehnt. Eine angemessene Darstellung einer architektonischen Entwurfsabsicht sollte deshalb nicht nur die statische Geometrie als Ergebnis des Entwurfs speichern, sondern die Distanzparameter und Abhängigkeiten zwischen den Komponenten abbilden. Somit könnten Änderungen der Geometrie direkt durch die Anpassung der verwendeten Parameter durchgeführt werden und die architektonische Entwurfsabsicht eindeutig kommuniziert werden.

2 State of the Art

Parametric Computer-Aided Design

Unter Parametric CAD versteht man den Versuch, Geometriemodelle nicht nur anhand ihrer globalen Koordinaten zu beschreiben, sondern durch geeignete parametrisierte geometrische Abhängigkeiten.

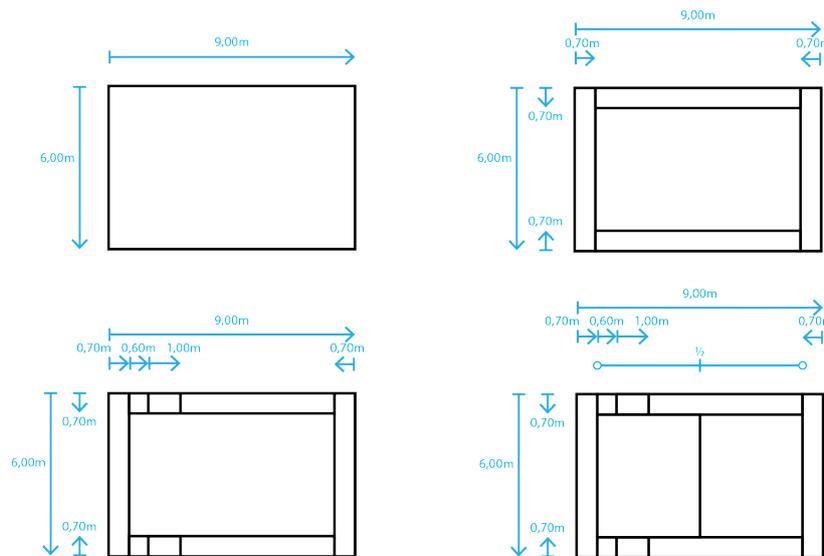


Abb. 1: Beispielgrundriss

Ivan Sutherland entwickelte in seiner Doktorarbeit am MIT *Sketchpad*, das als erstes CAD-System gilt (Sutherland, 1963). Er beschränkte sich dabei nicht auf das digitale Erstellen statischer Zeichnungen. Vielmehr betonte er bereits die Möglichkeit, topologische Abhängigkeiten in das Modell einzugeben und geometrische Constraints wie Parallelität auf der Geometrie zu definieren. Er bezeichnete *Sketchpad* als Man-Machine Graphical Communication System; "... [that can] interpret information drawn directly on a computer display". Schon zu Beginn war es also Ziel von CAD-Systemen, neben der Erstellung von Geometrien auch Anwenderwissen wie eine Entwurfsabsicht zu modellieren. Frühe Arbeiten, die meistens im Kontext des Maschinenbaus entstanden sind, betonten dabei vor allem die Rolle der verwendeten Distanzparameter als Zwangsbedingungen zu Parametrisierung der Geometrie. Anstelle von parametrischem CAD sprach man anfangs von *Dimension Driven Geometry*. In (Roller et al., 1989) stellt Roller verschiedene solcher Ansätze vor. Sie lassen sich dabei in drei Gruppen unterteilen:

1. *Constructive Approach*: Hierbei wird die Geometrie als Ergebnis der verwendeten Konstruktionsschritte modelliert. Alle Modellierungsoperationen werden in einem gerichteten Konstruktionsgraphen gespeichert. Um nachträgliche Änderungen durchzuführen, werden beispielsweise die in einem Konstruktionsschritt verwandten Parameter geändert und das Modell anschließend entlang des Konstruktionsgraphen neu berechnet. (Fitzgerald, 1981; D. C. Gossard et al., 1988; Kondo, 1990; Mun et al., 2003; Roller, 1991; Solano und Brunet, 1994)

Parametrisierte Architekturzeichnungen: Architektonische Entwurfsabsicht durch parametrisierte Koordinatenabhängigkeiten beschreiben und darstellen

2. *Variational Geometry*: Bei diesen Ansätzen spielt die Reihenfolge, in der die Geometrie modelliert wurde keine Rolle. Das Modell besteht aus Geometrie mit bestimmten topologischen Eigenschaften und darauf definierten parametrisierten Zwangsbedingungen (Light et al., 1981; Light und D. Gossard, 1982). Dafür muss der Nutzer zuerst eine Geometrie modellieren und anschließend definieren, durch welche Zwangsbedingungen sein Modell parametrisiert werden soll.
3. *Expert Systems*: Die letzte Gruppe von Ansätzen versucht die Geometrie anhand einer Syntax zu beschreiben, die sich aus der jeweiligen Anwendungsdomäne ableitet. Wie beim *Constructive Approach* werden hierbei die Modellierungsoperationen des Nutzers direkt in Beziehungen zwischen den Geometrieobjekten übersetzt (Sunde, 1986). Anstelle von geometrischen Zwangsbedingungen werden die Abhängigkeiten dabei jedoch in Form von Regeln symbolisch beschrieben (Verroust et al., 1992).

Parametrische Systeme im Bauwesen

Vor allem im Maschinenbau bieten viele Systeme wie *PTC Creo Parametric* die Erstellung umfangreich parametrisierter Modelle an. In der architektonischen Praxis konnten sich entsprechende Systeme bisher jedoch nicht durchsetzen (Monedro, 2000). Dort werden CAD-Systeme noch immer meist als reine Zeichenwerkzeuge verwendet (CADD - computer aided drafting and design). Der Nutzer erstellt aus abstrakten geometrischen Primitiven eine statische Zeichnung der Gebäudegeometrie. Anschließende Änderungen werden aufwendig durch Transformationsoperationen (Translation, Skalierung, Rotation) auf den geometrischen Objekten umgesetzt. Die einzelnen am Bauprozess beteiligten Akteure kommunizieren durch den Austausch statischer Geometrien, die mit zusätzlichen Informationen kommentiert werden. Als BIM (Building Information Modelling) wird der Versuch bezeichnet, solche statischen Abbildungen durch ein "intelligentes" digitales Gebäudemodell zu ersetzen (Bentley und Workman, 2003; Eastman et al., 2003). Dabei wird mit einem Schwerpunkt auf dem Projekt- und Datenmanagement versucht, alle Informationen, die für Planung, Realisierung und Betrieb eines Gebäudes relevant sind in einem digitalen Modell zu bündeln. Viele dieser Informationen lassen sich dabei geometrisch beschreiben. BIM-Systeme wie Revit oder Archicad strukturieren die Gebäudegeometrie dabei durch parametrisierte, semantische Bauteilobjekte. Sie können mit Metadaten wie Kosten oder Typbezeichnung ausgestattet werden. Aus ihrer Semantik wird dabei Domänenwissen abgeleitet und als Zwangsbedingungen modelliert. Geometrische Abhängigkeiten die sich über die Bauteilgrenzen hinweg durch das Gebäudemodell erstrecken, lassen sich jedoch nicht implizit aus der Semantik ableiten. Für die Beschreibung des Entwurfsgedanken sind solche Abhängigkeiten jedoch unerlässlich.

3 Parametrisierte Architekturzeichnungen - Modellierung

Im Folgenden soll ein System entwickelt werden, das bei der Erstellung von Grundrissen, Schnitten und Ansichten die architektonische Entwurfsabsicht durch eine geeignete Parametrisierung abbilden kann. Ausgehend von der vorhergehenden Analyse existierender Systeme wurden dabei die folgenden Gestaltungskriterien für ein geeignetes System ermittelt.

Defizitanalyse und Designkriterien

1. *geringer Mehraufwand durch automatisches Ableiten der Zwangsbedingungen:* Aus verschiedenen Gründen ist im architektonischen Kontext die Motivation zum parametrischen Arbeiten geringer. Die Geometrien sind in der Regel wenig komplex und auch in statischen Systemen gut zu handhaben. Zudem sind Architekten weniger technikaffin als beispielsweise Ingenieure. Die gezeigten Systeme haben es nicht geschafft, den Mehraufwand in parametrischen Systemen entsprechend gering zu halten, um diese auch Architekten zugänglich zu machen. Deshalb soll die Parametrisierung ausschließlich durch die Interpretation der üblichen Modellierungsoperationen erzeugt werden. Die Erzeugung des parametrisierten Modells unterscheidet sich dabei nicht von der Erzeugung statischer Geometrien in den gängigen CAD-Systemen.
2. *Lesbarkeit der geometrischen Abhängigkeiten:* Die beschriebenen Geometrien im architektonischen Kontext sind meist umfangreicher als beispielsweise im Maschinenbau. Die Darstellung der geometrischen Abhängigkeiten auf der Zeichnung, wie es in parametrischen Systemen üblich ist, führt dabei schnell zu einer überladenen Benutzeroberfläche. Die Lesbarkeit der Abhängigkeiten ist dabei nicht mehr gewährleistet. Die größte Hürde zur Nutzung parametrischer Systeme im architektonischen Kontext ist daraus resultierende komplexe Benutzeroberfläche. Der architektonische Entwurfsprozess ist nicht nur von technischen Kriterien bestimmt, sondern vor allem auch von ästhetischen Überlegungen. Aus diesem Grund sollte die grafische Benutzeroberfläche möglichst reduziert gestaltet werden, um das visuelle Denken zu unterstützen. Eine benutzerfreundlichen Benutzeroberfläche ist deshalb zentraler Bestandteil des beschriebenen Systems.
3. *Beschränkung auf Orthonormales Koordinatensystem* Eine uneingeschränkte Übertragung der parametrischen CAD-Systeme aus dem Maschinenbau ist somit nicht möglich. Die Abbildung aller Constraints und Parameter auf der Geometrie führt bei umfangreichen architektonischen Geometrien zu einer schlechten Lesbarkeit und einer überladenen Benutzeroberfläche. Unter anderem durch die Beschränkung auf orthogonale Geometrien soll versucht werden die Komplexität des Systems zu reduzieren und die Lesbarkeit zu erleichtern. Im architektonischen Kontext ist eine solche Einschränkung gerechtfertigt. Rechte Winkel dominieren die gebaute Umwelt. Zwar sind oftmals ganze Gebäudeteile zueinander verdreht; lokal überwiegen jedoch orthogonale Geometrien.
4. *keine Semantik, keine Entwurfsunterstützung* Vor allem im Bauwesen versuchen bestehende Systeme der Parametrisierung der Modelle von semantischen Bauteilobjekten auszugehen. Dies entspricht jedoch nicht der abstrakten Denkweise entwerfender Architekten. Die Semantische Anreicherung der Objekte wird oftmals dazu genutzt um daraus automatisch Zwangsbedingungen abzuleiten. So können beispielsweise Fenster in BIM-Systemen nur innerhalb von Wänden definiert werden. Wir möchten jedoch kein System entwickeln, das den Nutzer daran hindert "invalide" Geometrien zu erzeugen. Der Gestalter soll nicht in seinen Entwurfsentscheidungen unterstützt werden. Vielmehr sollen seine frei getroffenen Entwurfsentscheidungen präzise abgebildet werden.

Syntax

Nun soll untersucht werden, wie sich geometrische Abhängigkeiten in einem orthonormalen Koordinatensystem automatisch aus den verwendeten Zeichenoperationen ableiten lassen. CAD-Systeme zur Erstellung zweidimensionaler Geometrien bieten dem Nutzer in der Regel Maus-

Parametrisierte Architekturzeichnungen: Architektonische Entwurfsabsicht durch parametrisierte Koordinatenabhängigkeiten beschreiben und darstellen

eingaben und die Eingabe expliziter Distanzen zur Erstellung präziser Modelle an. Bei der Erstellung geometrischer Objekte, wie Linien oder Rechtecken, weist der Nutzer diesen Koordinaten im globalen Koordinatensystem des Modells zu. Möchte ein Nutzer ein präzises Modell erzeugen, so gibt er diese Koordinaten selten in globaler Form an. Vielmehr bezieht er sich per Mauseingabe und Objektfang meist auf bereits existierende Geometrien. Zur Vereinfachung soll zunächst nur die Erzeugung von Rechtecken beschrieben werden. In einem orthonormalen zweidimensionalen Koordinatensystem ist ein Rechteck anhand zweier Koordinaten in jeder Dimensionen bestimmt. Die Koordinatenrichtungen können dabei unabhängig voneinander betrachtet werden. In jeder Dimension muss der Nutzer dem Rechteck also zwei Koordinaten zuweisen. Dabei hat er unter Verwendung der gängigen Eingabemethoden folgende drei Möglichkeiten:

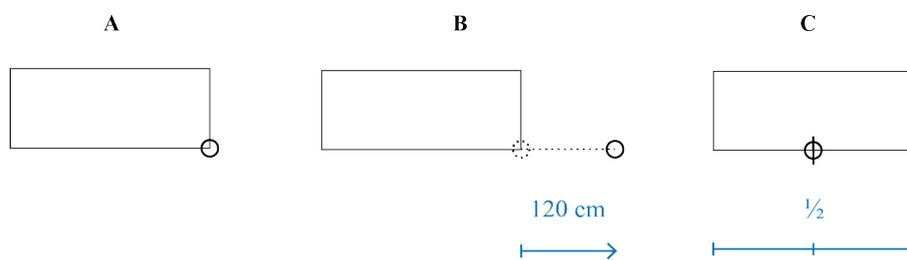


Abb. 2: Position unter Bezug auf existierende Geometrien bestimmen: A Objektfang, B Abstand eingeben, C Objektfang in Mittelpunkt

- A: Koordinate übernehmen** Per Objektfang kann der Nutzer die Koordinate eines Punktes oder einer Kante eines vorhandenen Rechtecks übernehmen.
- B: expliziter Abstand zu existierender Koordinate** Durch die Wahl eines Bezugspunktes und die Eingabe eines expliziten Distanzparameters kann die Koordinate beschrieben werden. Diese ist also durch einen Bezugspunkt und einen Abstand zu diesem definiert.
- C: Relative Position zwischen zwei Koordinaten** Durch einen Objektfang in ein vom System angebotenen relativen Bezugspunkt (z.B. Mittelpunkt) kann eine Koordinate zwischen zwei bestehenden Koordinaten beschrieben werden. Deren Position ist also durch zwei Bezugspunkte und eine baryzentrische Koordinate definiert.

Ausgehend von dieser Analyse der Nutzereingaben während der Geometrieerstellung in klassischen CAD-Systemen wird nun eine Syntax entwickelt, die die daraus abgeleiteten geometrischen Abhängigkeiten geeignet beschreibt. Die Abhängigkeiten des Modells werden dabei nicht zwischen den geometrischen Objekten sondern zwischen entkoppelten Koordinatenobjekten definiert. Dies ergibt sich aus der Analyse der Nutzerinteraktion mit dem Systems. Bei der oben beschriebenen Geometrieerstellung durch relative Bezüge bezieht sich der Nutzer nicht auf explizite Objekte, sondern vielmehr auf deren Position. Daraus ergeben sich somit folgende zwei Koordinatentypen, aus denen sich die Parametrisierung konstruieren lässt. Betrachten wir dabei eine Koordinatenrichtung:

μ -Koordinate: expliziter Abstand zu bestehender Koordinate

Wird wie in eine Koordinate durch einen explizit eingegebenen Abstandsparameter zu einer anderen definiert (Fall B) so wird dabei eine neue μ -Koordinate erzeugt. Diese besitzt eine Bezugsordinate c_{parent} und einen Distanzparameter μ .

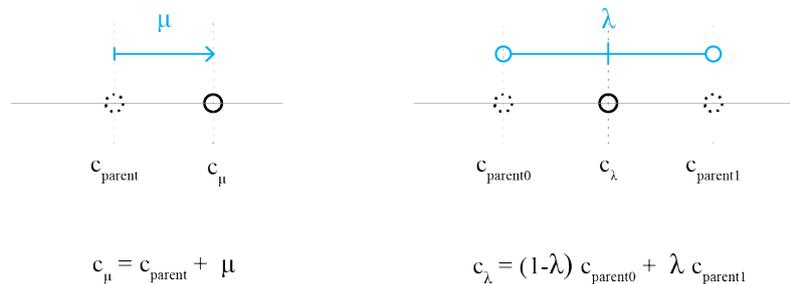


Abb. 3: Die zwei Koordinatentypen

λ -Koordinate: relativer Position zwischen zwei bestehenden Koordinaten

Wird ein Objekt durch einen relativen Bezugspunkt definiert (Fall C), so wird an dieser Stelle ein baryzentrische λ -Koordinate erzeugt. Diese besitzt zwei Bezugskoordinaten, $c_{parent0}$ und $c_{parent1}$, sowie einen Parameter λ , der einen Punkt zwischen ihnen oder darüber hinaus definiert.

Verwendung gleicher Distanzparameter

An dieser Stelle sollen die Eingabemethoden klassischer CAD-Systeme erweitert werden. Im Eingabedialog der die Eingabe eines expliziten μ -Distanzparameter erwartet, soll es möglich sein auf einen bereits existierenden Parameter zu verweisen, anstatt dessen Wert erneut einzugeben. Wird also zum Beispiel eine Wand erzeugt, die die gleiche Stärke wie eine bereits existierende Wand besitzen soll, so ist es möglich deren μ -Parameter zu übernehmen. Während das System die Eingabe einer Distanz erwartet, ist es möglich auf einen existierenden Distanzparameter zu markieren um diesen zu übernehmen. Die Distanzparameter existieren also entkoppelt von den sie verwendenden Koordinatenobjekten. Die Richtung des Distanzparameters wird dabei im Koordinatenobjekt hinterlegt.

Abhängigkeiten im Beispiel

In Abb. 4 wurde das beschriebene Modell an dem anfangs gezeigten Beispielgrundriss angewendet. Der Grundriss besteht aus Rechtecken. Bei der Erzeugung dieser Rechtecke wurden die Nutzereingaben in Abhängigkeiten zwischen den Koordinaten übersetzt. So hängt nun x_5 , die rechte Seite der Tür, von x_4 , der linken Seite der Tür, ab. Diese wiederum ist durch die Position von x_2 und dem Distanzparameter d bestimmt.

Datenstruktur

1. *Die Geometrieebene:* Die Geometrieobjekte sind von der Parametrisierung entkoppelt. Es liegt also eine Liste von Geometrieobjekten vor, in denen eine Referenz auf die von ihnen verwendeten Koordinatenobjekte hinterlegt ist. Ein Rechteck ist so in jeder Koordinatenrichtung durch zwei Koordinaten beschrieben.

Parametrisierte Architekturzeichnungen: Architektonische Entwurfsabsicht durch parametrisierte Koordinatenabhängigkeiten beschreiben und darstellen

2. *Die Koordiantenebene:* Ausgehend von einem Koordinatenursprung werden die Koordinaten einer Koordinatenrichtung durch das sukzessive Erzeugen von μ - und λ - Koordinaten erzeugt. Der entstehende Abhängigkeitsgraph ist gerichtet und besitzt keine Zyklen (directed acyclic graph). μ -Koordinaten besitzen einen Elternknoten, λ - Koordinaten zwei.

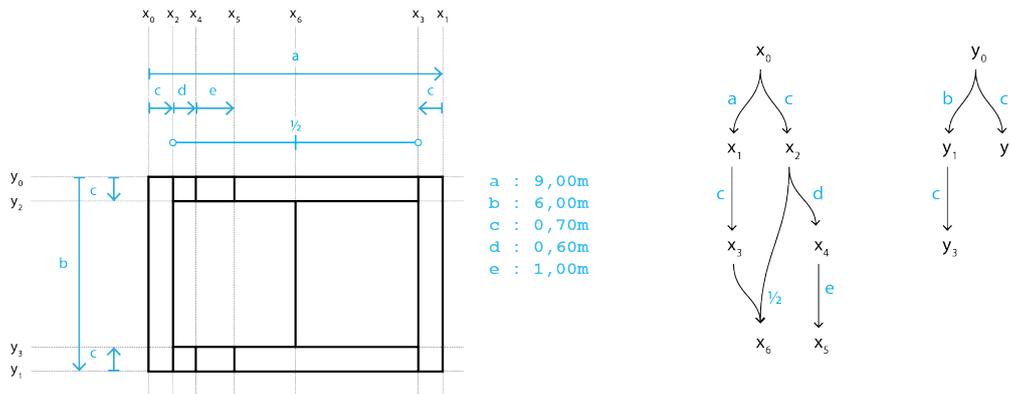


Abb. 4: Beispiel

3. *Die Parameterebene:* Die Distanzparameter der μ -Koordinaten sind von diesen entkoppelt. Die μ -Koordinaten besitzen lediglich eine Referenz auf ein Parameterobjekt der Liste von μ -Distanzparametern. λ -Parametern werden direkt in den λ -Koordinaten hinterlegt, da diese nicht mehrfach verwendet werden sollen.

Die Datenstruktur ist also streng hierarchisch aufgebaut. Änderungen der Parameter wirken sich auf die Koordinatenobjekte aus. Diese wiederum beschreiben die Geometrie.

4 Änderungenoperationen

Im Folgenden sollen die wichtigsten Änderungenoperationen am parametrisierten Modell beschrieben werden.

Änderung der Distanzparameter

Die einfachste Operation zur Anpassung der Koordinaten ist die Veränderung der verwendeten Distanzparameter. Anhand dieser Operationen wird besonders gut deutlich, worin die Vorteile eines parametrisierten Modells liegen. Einem der verwendeten Parameter in der Parameterliste wird ein neuer Wert zugewiesen. Anschließend werden entlang des Konstruktionsgraphen die Positionen der Koordinaten im globalen Koordinatensystem neu berechnet. In Abb.5 wird die Wandstärke verändert. Dies wirkt sich auf alle Koordinaten aus, die mit diesem Distanz parametrisiert worden sind. Es wird also die Stärke aller Wände angepasst. Entsprechende Anpassungen vormals beschriebener Distanzen kommen im Entwurfsprozess regelmäßig vor. In klassischen, statischen CAD-Systemen müssen diese durch Skalierungs- und Translationsoperationen auf allen betroffenen geometrischen Objekten realisiert werden. Der Nutzer muss sich dabei der zuvor getroffenen Entwurfsentscheidungen erneut bewusst werden oder diese vom Autor des Modells erfragen. Dies ist ein sehr arbeitsaufwendiger und fehleranfälliger Prozess.

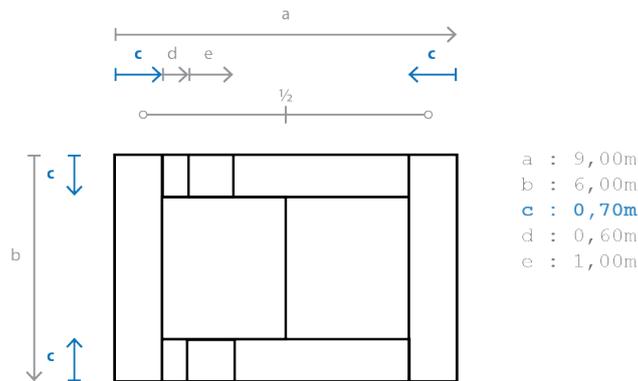


Abb. 5: Parameterwert anpassen

Koordinaten umhängen

Angenommen der Nutzer möchte die Koordinate x_4 die die Position der beiden Türen bestimmt nun nicht mehr durch einen Abstand zu x_2 beschreiben (Abb. 6). Hierfür ist es möglich, diese Koordinate als $x_{markiert}$ auszuwählen und sie an eine andere Stelle im Modell zu hängen. Zunächst wird dabei der Teilgraph aller Koordinaten, die von $x_{markiert}$ abhängen ermittelt. Anschließend wird die ursprüngliche Definition von $x_{markiert}$ als μ -Koordinate mit Abstand d zu x_2 verworfen. Der Nutzer wählt anhand der in Abschnitt 3 beschriebenen Methoden eine neue Koordinate x_{neu} im System durch die $x_{markiert}$ ersetzt werden soll. Um Zyklen zu vermeiden, kann er sich dabei jedoch nicht auf Koordinaten beziehen, die Teil des ermittelten Teilgraphen sind. Anschließend müssen die Zeiger aller Geometrien, die ursprünglich auf die verworfenen Koordinate x_4 gezeigt haben, durch Zeiger auf die neue bestimmte Koordinate ersetzt werden. Der Subgraph, dessen Wurzel $x_{markiert}$ war, wird an diese neue Koordinate gehängt werden.

5 Grafische Benutzeroberfläche - Hybrid aus Graph und Maßketten

Es lässt sich also ohne großen Aufwand eine hierarchische Parametrisierung des Modells erzeugen, die die Entwurfsabsicht angemessen abbildet. Auch Änderungen am Modell können effizient durchgeführt werden. Nun stellt sich jedoch die Frage, wie sich die innere Struktur der Geometrie darstellen lässt. Die Analyse bestehender parametrischer CAD-Systeme, wie sie im Maschinenbau verwendet werden, hat gezeigt, dass die Darstellung aller Zwangsbedingungen auf der Geometrie schnell zu einer schlecht lesbaren Benutzeroberfläche führt. Durch die Beschränkung auf ein orthonormales Koordinatensystem sind jedoch weit weniger Informationen notwendig, um die Geometrie zu beschreiben. Zudem entfallen wenig intuitive Symbole zur Darstellung geometrischer Zwangsbedingungen wie Parallelität. Die Maßketten einer Koordinatenrichtung lassen sich so positionieren, dass deren topologische Struktur ablesbar ist. Es entsteht ein Hybrid aus Graph und Maßkette, der die Vorteile beider Darstellungen vereint. Im Folgenden sollen anhand eines Beispiels (Abb. 7) einige erste Überlegungen zur Gestaltung der Maßketten in der Benutzeroberfläche dargestellt werden.

1. *Positionierung*: Die Maßkette einer Koordinate sollte möglichst an der Geometrie positioniert werden, die durch sie beschrieben ist. Sind mehrere Geometrien auf der selben

Parametrisierte Architekturzeichnungen: Architektonische Entwurfsabsicht durch parametrisierte Koordinatenabhängigkeiten beschreiben und darstellen

Koordinate definiert, so wir die Maßkette an der Geometrie positioniert, die sich am weitesten oben beziehungsweise links befindet. Werden mehrere Maßketten an der selben Position dargestellt, so werden sie entsprechend der Konstruktionshistorie sortiert: Koordinaten mit geringerem Abstand zur Wurzel werden weiter oben beziehungsweise links positioniert und die entfernteren direkt an der Geometrie.

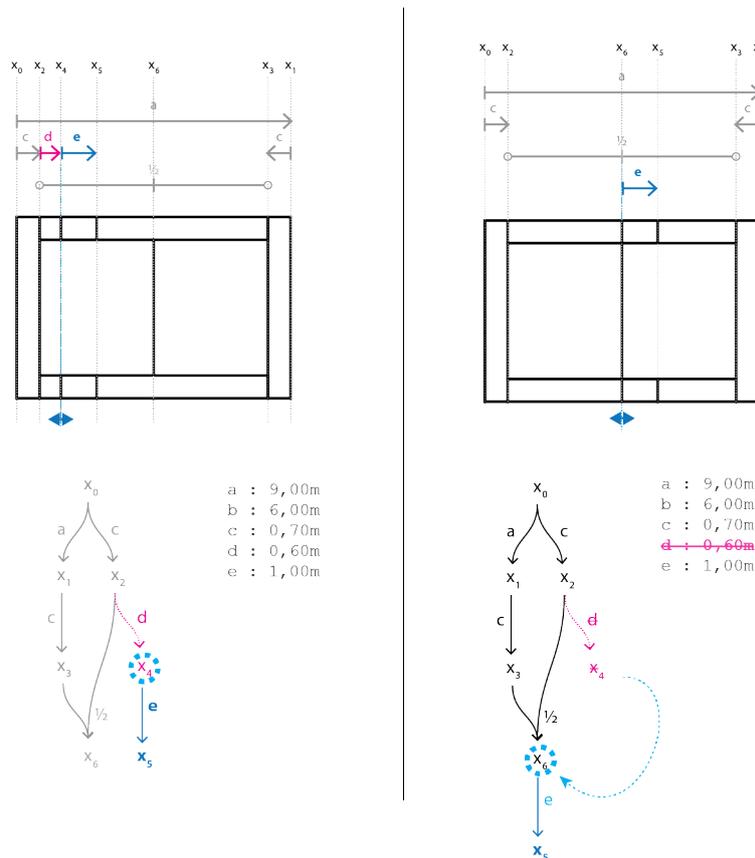


Abb. 6: Parameterwert anpassen

2. *Kapselung*: Es entspricht der Denk- und Arbeitsweise von Architekten bestimmte Koordinatenmuster zu kapseln. Ein solches Muster ist die Unterteilung einer Distanz in n gleiche Teile. Treppenstufen werden beispielsweise durch die gleichmäßige Unterteilung der Treppenlänge definiert. In der Nutzeroberfläche werden auch die Maßketten dieses Musters entsprechend zusammengefasst.
3. *Filterung*: Wie anfangs beschrieben ist es vor allem der große Umfang von Gebäudegeometrien, der die Verwendung parametrischer Modellierungsmethoden erschwert. Trotz der Beschränkung auf orthogonale Geometrien ist es auch im beschriebenen Interface schwierig, alle Abhängigkeiten eines Gebäudemodells gleichzeitig gut lesbar darzustellen. Es wird deshalb vorgeschlagen, die Maßketten der Koordinaten zu filtern und in jeder Zoomstufe nur die relevantesten Koordinaten zu visualisieren. So soll beispielsweise bei der Ansicht des gesamten Gebäudes primär die Geometrie makroskopischer Elemente wie Räume und Kerne erklärt werden [Abb.7, links].

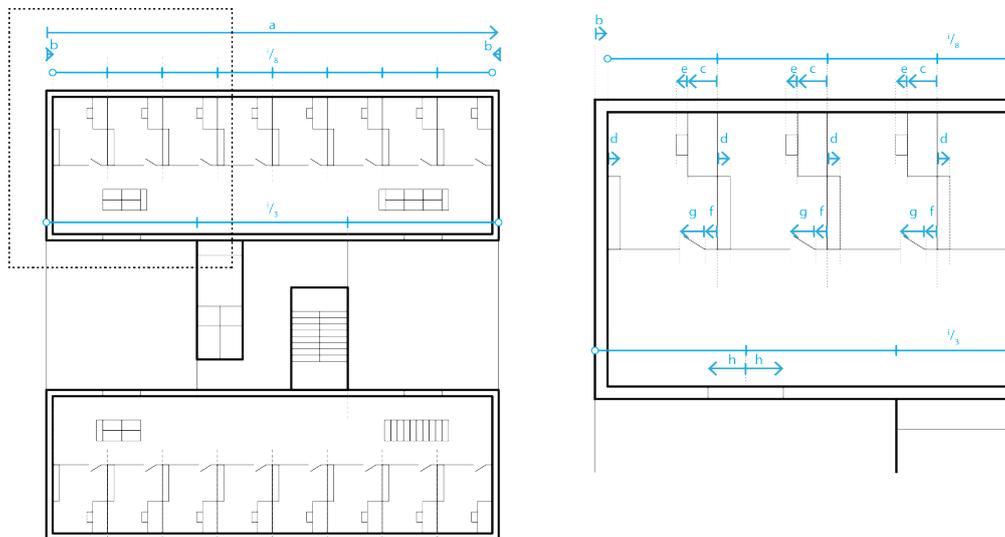


Abb. 7: Beispiel Herzog & De Meuron: Verschiedene Zoomstufen eines Grundrisses

Zusammenfassung

Der beschriebene Ansatz schafft es die eingangs beschriebenen Defizite vorhandener Lösungsansätze zu lösen. Der Mehraufwand im Vergleich zu klassischen CAD-Systemen bleibt sehr gering. Die geometrischen Abhängigkeiten und Parameter, die ein wichtiger Bestandteil der architektonischen Entwurfsabsicht sind, können ohne zusätzliche Nutzereingaben abgebildet werden. Es wurde gezeigt, dass sich diese automatisch aus den Nutzereingaben im Zeichenprozess eindeutig ableiten lassen. Die wichtigste Entscheidung im Entwurf des Systems war die Beschränkung auf ein orthonormales Koordinatensystem. Dadurch konnte eine sehr einfache und verständliche Syntax zur Beschreibung der geometrischen Abhängigkeiten entwickelt werden. Diese orientiert sich an der Art und Weise wie architektonische Geometrien erdacht, sprachlich beschrieben und gezeichnet werden. Auch konnte gezeigt werden, wie sich die wichtigsten Änderungsoperationen effizient durch Änderungen am Konstruktionsgraphen realisieren lassen. Für die Gebrauchstauglichkeit in der architektonischen Praxis ist die Lesbarkeit und Handhabbarkeit von umfangreichen parametrisierten Modellen entscheidend. Der größte Mehrwert des orthonormalen Systems ist die sich daraus ergebende Benutzeroberfläche. Da die Abhängigkeiten einer Dimension isoliert behandelt werden, lässt sich deren Struktur intuitiv lesbar darstellen. Die Überlegungen zur Gestaltung der Benutzeroberfläche sollen einen Ausblick darauf bieten, wie eine solche Oberfläche weiterentwickelt und optimiert werden kann. Aus Sicht der Datenstruktur lässt sich das System problemlos auf dreidimensionale Geometrien erweitern. Es stellt sich dabei die Frage, in wie weit sich die Abhängigkeiten auch auf dreidimensionalen Ansichten darstellen lassen. Des weiteren wäre eine Erweiterung des Systems auf nicht orthogonale Geometrien wünschenswert. Dabei sollte jedoch zumindest lokal immer die Orthogonalität der Geometrie zur Vereinfachung des Systems genutzt werden. Die Definition lokaler orthonormaler Koordinatensysteme, die zueinander verdreht sein können, wäre ein möglicher Lösungsansatz. Die beschriebene Methode kann einen Beitrag zu CAD als wirkliche Designhilfe leisten. Vor allem in frühen Projektphasen, wenn ausgehend von Skizzen eine erste präzise Geometrie entwickelt wird, bietet der Ansatz viele Vorteile und entspricht in vielerlei Hinsicht der Arbeits- und Denkweise von Architekten.

Parametrisierte Architekturzeichnungen: Architektonische Entwurfsabsicht durch parametrisierte Koordinatenabhängigkeiten beschreiben und darstellen

Literatur

- Bentley, K. und Workman, B. (2003). „Does the building industry really need to start over“. In: *A Response from Bentley to Autodesk's BIM/Revit Proposal for the Future*, Bentley. Bentley Systems.
- Eastman, C., Lee, G. und Sacks, R. (2003). „Development of a knowledge-rich CAD system for the North American precast concrete industry“. In: *Automation in Construction* 13 (3), S. 291–312.
- Fitzgerald, W. J. (1981). „Using axial dimensions to determine the proportions of line drawings in computer graphics“. In: *Computer-Aided Design* 13.6, S. 377–382.
- Ganghofer, D. (2016). „Parametrisierte Architekturzeichnungen: Architektonische Entwurfsabsicht durch parametrisierte Koordinatenabhängigkeiten beschreiben und darstellen“. Bachelorarbeit Informatik. Freie Universität Berlin.
- Gossard, D. C., Zuffante, R. P. und Sakurai, H. (1988). „Representing dimensions, tolerances, and features in MCAE systems“. In: *IEEE Computer Graphics and Applications* 8.2, S. 51–59.
- Iyer, G. R. und Mills, J. J. (2006). „Design intent in 2D CAD: definition and survey“. In: *Computer-Aided Design and Applications* 3.1-4, S. 259–267.
- Kondo, K. (1990). „PIGMOD: parametric and interactive geometric modeller for mechanical design“. In: *Computer-Aided Design* 22.10, S. 633–644.
- Light, R., Lin, V. und Gossard, D. (1981). „Variational geometry in CAD“. In: *Computer Graphics* 15.3, S. 171–177.
- Light, R. und Gossard, D. (1982). „Modification of geometric models through variational geometry“. In: *Computer-Aided Design* 14.4, S. 209–214.
- Monedro, J. (2000). „Parametric Design: a review and some experiences“. In: *Automation in Construction* 9.4, S. 369–377.
- Mun, D., Han, S., Kim, J. und Oh, Y. (2003). „A set of standard modeling commands for the history-based parametric approach“. In: *Computer-Aided Design* 35.13, S. 1171–1179.
- Roller, D. (1991). „An approach to computer-aided parametric design“. In: *Computer-Aided Design* 23.5, S. 385–391.
- Roller, D., Schonek, F. und Verroust, A. (1989). „Dimension-driven geometry in CAD: a survey“. In: *Theory and practice of geometric modeling*. Springer, S. 509–523.
- Solano, L. und Brunet, P. (1994). „Constructive constraint-based model for parametric CAD systems“. In: *Computer-Aided Design* 26.8, S. 614–621.
- Sunde, G. (1986). „Specification of shape by dimensions and other geometric constraints“. In: *IFIP WG. 5.2 on Geometric Modeling*.
- Sutherland, I. E. (1963). *Sketchpad: A man-machine graphical communications system*. Technical Report 296.
- Verroust, A., Schonek, F. und Roller, D. (1992). „Rule-oriented method for parameterized computer-aided design“. In: *Computer-Aided Design* 24.10, S. 531–540.

Digitale Handskizzen Analyse: Stiftbasiertes Interface für Design Decision Support

Dorian Zank

*Lehrstuhl für Architekturinformatik, Technische Universität München Arcisstr.21,
80333 München, Deutschland
E-Mail: dorian.zank@mytum.de*

Abstract Trotz der Digitalisierung vieler Arbeitsprozesse, werden im architektonischen Entwurf immer noch hauptsächlich analoge Entwurfswerkzeuge, wie die Handskizze, verwendet. Ziel dieser Arbeit war es, die Potenziale des Rechners frühzeitig und sinnvoll in den Entwurf einzubinden. Hierfür wurde die theoretische Basis für ein stiftbasiertes Werkzeug geschaffen, welches den Architekten in der frühen Entwurfsphase mit Analysen und Simulationen unterstützt. Als Grundlage für das Konzept wurde die Nutzung der Handskizze als Entwurfswerkzeug untersucht. Ebenso wurde analysiert, wie Architekten im Kreativprozess aktuelle Simulationswerkzeuge nutzen, und wo Mängel existieren. Zudem wurde anhand themennaher Lösungen eine Defizitanalyse erstellt. Aus den gesammelten Erkenntnissen konnte dann ein Konzept formuliert werden. Ziel war es, dem Architekten unmittelbar objektive Daten zu seiner Planung zu liefern. Dazu werden direkt in der Handskizze unterstützende Daten, wie etwa Tageslichtsimulationen, visualisiert. Das Konzept wurde prototypisch in Processing umgesetzt. Die Funktion des Prototyps wurde anhand einer Nutzerstudie evaluiert.

Keywords: Handskizze, Analyse, Design-Unterstützung, Stiftbasiert

1 Einleitung

In Zeiten der Digitalisierung vieler Lebensbereiche verlassen sich Menschen immer mehr auf Computer und ihre Unterstützung. Auch die Architektur ist von dieser Entwicklung nicht ausgeschlossen. Planzeichnungen werden in CAD und BIM Software erstellt und mit Hilfe von 3D-Programmen können aufwändige Visualisierungen erstellt werden. Dennoch, oder vielleicht genau deshalb, nutzen Architekten in der Entwurfsphase bevorzugt noch immer analoge Entwurfsmittel. Dabei spielt die Handskizze eine besondere Rolle. Sie dient dem Architekten zur Visualisierung mentaler Prozesse, zur Kommunikation, und auch als persönliches Ausdrucksmittel. Es ist kein Wunder, dass die Handskizze als die Sprache des Architekten gilt (Peichl et al., 2013). Der Computer hingegen wird für die Entwurfsarbeit kaum verwendet, viel häufiger dient er lediglich zur Dokumentation und Ausführung fertiger Gedanken.

Die Ursachen hierfür sind vielfältig. Viele Programme sind nicht auf die Bedürfnisse des frühen Entwurfsprozesses zurechtgeschnitten, oder erschweren durch komplizierte Benutzeroberflächen die Bedienung. Als Folge behindert die Interaktion mit dem Rechner den Entwurfsgedanken, anstatt ihn zu bereichern. Will der Planer sowohl digitale als auch analoge

Entwurfswerkzeuge verwenden, muss er zwischen ihnen wechseln, anstatt sie gleichzeitig nutzen zu können. Diese sequentielle Nutzung ist hinderlich für den Entwurf (Schubert, 2014, S. 2). Ziel der Arbeit war es, den vorhandenen Bruch zwischen analogen und digitalen Werkzeugen zu überbrücken, um so den PC frühzeitig und sinnvoll in den Entwurf einzubinden. Hierfür wurde ein Konzept für ein neues Werkzeug geschaffen, welches den Architekten in seiner Entwurfsarbeit bereichert, ihn aber nicht entmündigt. Da Architekten nach wie vor die analoge Skizze präferieren (Bailey, 2002, S. 137), wird diese als Basis für das Konzept genutzt. Ergänzt man dieses Konzept mit nützlichen, digitalen Tools, verschwinden die Grenzen zwischen digitalen und analogen Medien, und ermöglichen so eine Nutzung beider ohne Einbußen.

Als Grundlage für das Konzept musste zunächst geklärt werden, wie die Handskizze als Entwurfswerkzeug genutzt wird. Dazu wurde der Entwurfsprozess im Allgemeinen, und im Hinblick auf die Handskizze als Entwurfswerkzeug im Speziellen untersucht. Mindestens genauso wichtig war die Frage, wie der Nutzer im Kreativprozess mit Analyse- und Simulationstools interagiert, und vor allem, wo Probleme auftreten. Dazu wurden vorhandene Lösungen untersucht und auf Vor- und Nachteile geprüft. Aus den gesammelten Erkenntnissen konnte dann ein Konzept formuliert werden. Der Lösungsansatz der Arbeit bestand darin, digitale und analoge Medien zu einem funktionierenden Verbund zu kombinieren. Als Folge wurde eine Software angedacht, die den Architekten in seiner kreativen Phase unterstützt, indem helfende Informationen, wie Simulationen oder Analysen, direkt in seine Handskizzen integriert werden. Der Nutzer kann dann wie gewohnt die Skizze als Werkzeug nutzen, profitiert aber gleichermaßen von den Vorteilen rechnergestützter Arbeitsweisen. Ausgehend von objektiven Daten und Zahlen, kann der Entwerfer nun begründete Entscheidungen treffen.

2 Die Skizze

In unserem Sprachgebrauch wird die Skizze als Begriff für ein unfertiges, vom Charakter her lückenhaftes Objekt verstanden. Häufig werden Skizzen als Mittel genutzt, um ohne großen Aufwand verschiedene Ideen zu testen, bevor eine dann ausführlich ausgearbeitet wird. Viele Skizzen teilen sich die Eigenschaft, dass sie, wenn auch nicht detailliert, wesentliche Punkte des zu schaffenden Werkes darstellen. In der Architektur hat die Skizze einen hohen Stellenwert. Sie wird allgemein als wichtiges Entwurfswerkzeug verstanden und angewendet. Vor allem in der frühen Entwurfsphase skizzieren Architekten um ihre Gedanken zu ordnen, zu verstehen und um neue Ideen zu schaffen (Do, 1999, S. 23 (nach Laseau, 1980 und McKim, 1972)).

Während der frühen Entwurfsphase erforscht der Architekt viele verschiedene Möglichkeiten ein gestelltes Problem zu lösen. Die Handskizze dient ihm dabei als externe Repräsentation seiner Ideen, unterstützt ihn beim Visual Thinking und hilft bei der Kommunikation. Eine Ansammlung verschiedenster grafischer Elemente ergibt zusammen mit Annotationen die Darstellung eines Entwurfsgedankens. Dabei werden nicht nur grafische Darstellungen möglicher Grundrisse oder Ansichten produziert, sondern auch Diagramme oder schnelle Schemadarstellungen. Da der Mensch sich nur eine limitierte Anzahl an Ideen merken kann, werden angefertigte Zeichnungen auch als externe Ideenspeicher genutzt (Tversky, 1999, S. 2). Dadurch können auf dem Papier festgehaltene Gedanken zu einem späteren Zeitpunkt einfach wieder abgerufen werden, selbst wenn diese nur angedeutet sind. Auch die Fähigkeit, sich genau definierte Formen vorzustellen, ist begrenzt (Schubert, 2014, S. 22). Eine externe Repräsentation unterstützt den Architekten bei der schrittweisen Konkretisierung. Insgesamt bietet die Handskizze, auch im Vergleich mit anderen üblichen Entwurfswerkzeugen, die größte Flexibilität in der Bedienung. Die Einstiegshürde ist, sowohl durch die benötigten Mittel, Stift und Papier, als auch durch die

verhältnismäßig einfache Erlernung, niedrig gehalten. Zwar schreibt Peter Eisenman „*Wir können nur das entwerfen, was wir zeichnen können.*“ (Eisenman und Schwarz, 1995), verglichen mit der Bedienung eines umfangreichen CAD Programmes jedoch bietet die Handskizze auch ungeübten Architekten die Möglichkeit Ideen schnell zu visualisieren. Im Gegensatz zu anderen Entwurfsmethoden, ist es mit Hilfe der Skizze zudem möglich, vage oder mehrdeutige Ideen darzustellen. Arbeitsmodelle etwa erschweren durch ihre physische Manifestation häufig eine mehrdeutige Interpretation. Auch 3D Modelle oder CAD Zeichnungen am Computer verlangen immer nach eindeutig definierten Koordinaten, sei es im zwei- oder dreidimensionalen Raum, und zwingen den Gestalter früh zu Entscheidungen, die möglicherweise noch nicht fundiert getroffen werden können. Die Abstraktion gibt dem Architekten die Möglichkeit, sich nicht zu früh auf bestimmte Merkmale festlegen zu müssen und Detaillierungen auf einen späteren, geeigneteren Zeitpunkt zu verschieben. Noch nicht durchdachte Bereiche werden einfach nur angedeutet, oder wie beispielsweise bei Raumdiagrammen stark abstrahiert, ohne dabei Rücksicht auf Fenster, Türen oder Wände nehmen zu müssen (Gross und Do, 1996, S. 184). Zu komplexe Darstellungen hingegen können zum Stocken des kreativen Prozesses führen und dadurch kontraproduktiv sein (Schubert, 2014, S. 24).

Die abstrakte und mehrdeutige Natur der Skizze ermöglicht einen weiteren Prozess, der zentral für die Ideenfindung ist: das Visual Thinking. Kern des Visual Thinking ist die Generierung neuer Ideen mit Hilfe von Bildern. Dies basiert darauf, dass man Darstellungen als etwas anderes wahrnimmt, als was man eigentlich sieht (Goldschmidt, 1994, S. 164). Dabei ist es leicht, die Begriffe *Sehen* und *Wahrnehmen* zu verwechseln. Dennoch gibt es einen großen Unterschied: das Auge sieht, das Gehirn nimmt wahr. Die Wahrnehmung ist immer eine persönliche Interpretation dessen ist, was wir sehen (Goldschmidt, 1994, S. 158). Diese Wahrnehmung bezieht sich entsprechend auch auf die Skizze. Die Skizze stellt eine subjektive Auffassung dar, und nicht eine objektive Darstellung (Tversky, 1999, S. 3). Der Begriff „Visual Thinking“ wurde 1971 von Rudolf Arnheim geprägt. Arnheim beschreibt dabei einen Kreislauf aus Idee, Ausführung und Interpretation (Schubert, 2014, S. 19 (nach Arnheim, 1971 und Schön, 1983)). Der Architekt beginnt zunächst mit einem mentalen Bild seiner Idee. Dieses wird zu Papier gebracht und anschließend interpretiert. Dabei liest der Architekt mehr Informationen aus der Skizze heraus, als er für die Darstellung investiert hat. Bei der Interpretation der Skizze ergänzt das Gehirn fehlende Informationen, um bestimmte Formen zu erkennen. Dabei wird es durch kulturelle, gesellschaftliche und soziale Faktoren beeinflusst. Man kann daraus schlussfolgern, dass verschiedene Menschen dasselbe Bild unterschiedlich auffassen können (Seichter, 2002, S. 5). Alternativ können durch das Bild Assoziationen geweckt, oder andere Ideen abgerufen werden (Suwa und Tversky, 1997, S. 386). Beeinflusst durch die gewonnenen Einblicke entsteht ein verändertes mentales Bild infolgedessen der Kreislauf aufs Neue beginnt. Die Skizze wird auch als die Kommunikation zwischen Gehirn und Papier beschrieben (Do, 1999, S. 24). Sie entwickelt sich vom reinen externen Ideenspeicher zu einem Katalysator für den Entwurfsgedanken. Der Zyklus endet normalerweise erst, sobald in der Skizze eine gewisse Kohärenz erreicht wird, oder er aufgrund von fehlendem Feedback abgebrochen wird (Goldschmidt, 1994, S. 164). Auch wenn der architektonische Entwurfsprozess rationellem Denken unterliegt, werden viele Entscheidungen aus Intuition oder künstlerischen Gründen getroffen. Die kreative Lösungsfindung bedient sich dabei oftmals visuellen Darstellungen, wie Skizzen oder Modellen. Deshalb ist Visual Thinking vor allem in der Architektur von entscheidender Bedeutung. Im Gegensatz dazu stehen zum Beispiel Naturwissenschaften, eher seltene Bilder nutzen (Goldschmidt, 1994, S. 160).

3 Analyse & Simulation

Vor allem in der frühen Entwurfsphase werden Entscheidungen getroffen, die den folgenden Entwurf sowohl optisch als auch technisch maßgeblich beeinflussen. Fehlentscheidungen können zu erheblichen Schwierigkeiten in der energetischen Performance und der potenziellen Nutzung eines Gebäudes während der gesamten Lebensdauer führen (Sarhan und Rutherford, 2014, S. 325–326). Deshalb ist es wichtig, dass mögliche Probleme möglichst früh im Entwurfsprozess erkannt und gelöst werden.

Der Computer kann mit Hilfe von Analysen und Simulationen objektive Daten liefern, die den Planer bei der Entscheidungsfindung unterstützen. Um tatsächlich einen Nutzen aus den gewonnenen Informationen zu ziehen, müssen die Ergebnisse jedoch in einer verständlichen Form präsentiert werden (Vanegas et al., 2009, S. 424). Es ist schwer, komplexe Daten bestimmten Entscheidungen in der Gestaltung zuzuordnen (Sarhan und Rutherford, 2014, S. 327). Auch die Visualisierung großer Datenmengen stellt eine Herausforderung dar. Im Rahmen dieser Arbeit mussten noch zusätzliche Anforderungen erfüllt werden, die aus der vagen Natur handgezeichneter Skizzen folgen. Zunächst stellte sich die Frage, was für eine Genauigkeit in der Berechnung und der anschließenden Darstellung sinnvoll ist. Eine millimetergenaue Ausgabe von Raumflächen ist bei schnell gezeichneten Wänden genauso fragwürdig, wie eine vollständige thermale Analyse für ein noch nicht zu Ende gedachtes Gebäude. Entsprechend müssen die Ergebnisse auch visuell präsentiert werden. Schnell interpretierbare Falschfarbdarstellungen sind abstrakten Zahlenreihen gegenüber vorzuziehen. Verschattungen beispielsweise sollten der skizzenhaften Darstellung entsprechend auch keine harten Kanten, sondern vielmehr leicht diffuse Übergänge besitzen.

Aktuelle Analysetools können zwar sehr präzise Daten liefern, erfüllen aber nur selten die Anforderungen an ein Entwurfswerkzeug für die frühe Entwurfsphase. Vor allem zu Beginn verfügt der Planer nur selten über die von vielen Programmen gefragten Informationen (Sarhan und Rutherford, 2014, S. 326). Vielmehr unterliegt der Entwurf ständigen Veränderungen, die für die Simulationen nötige Detaillierung findet noch nicht statt (Sarhan und Rutherford, 2014, S. 326). Zudem liefern solch präzise Analysen auch keine für die frühe Formfindung nützliche Daten. Eine weitere Schwierigkeit bildet die Nutzung der entsprechenden Analysesoftware, die vielen Architekten Probleme bereitet. Vor allem im Bereich der thermischen Planung verhindert eine komplizierte Bedienung häufig, dass die Software überhaupt verwendet wird (Lin und Gerber, 2014, S. 427). Vorhergehende Studien haben zudem gezeigt, dass lediglich 10% der vorhandenen Programme speziell auf die Bedürfnisse von Architekten zugeschnitten sind, und nur etwa 1% auf die frühe Entwurfsphase eingehen (Lin und Gerber, 2014, S. 427). Es ist anzunehmen, dass mögliche Vorteile computergestützter Analysen durch schwere Bedienung verloren gehen können.

Zusammenfassend lässt sich festhalten, dass zusätzliche Informationen, Analysen und Simulationen den Planer mit Hilfe objektiver Daten im Entwurfsprozess unterstützen können. Wichtig ist, dass eine komplizierte Anwendung und Auswertung keine zusätzlichen Probleme schaffen und somit zum entwurfshemmenden Faktor werden. Dabei spielen vor allem die Bedienung und Auswertung eine wichtige Rolle.

4 Defizitanalyse themenverwandter Lösungen

Bereits mit dem Aufkommen des PCs in den 60er-Jahren wurde versucht, die Vorteile von Handskizzen mit den praktischen Möglichkeiten des Computers zu vereinen. Es wurden erste

Zeichenprogramme entwickelt, welche die Ausgangsbasis für heutige CAD und BIM Software bilden. Meistens sind diese Programme aber eher als digitaler Zeichentisch für präzise Plandarstellungen oder für genaue 3D-Modelle zu sehen. Eine Auswahl geeigneter Lösungen wurde auf ihre Eigenschaften und Bedienung hin geprüft. Es wurden 2D und 3D stiftbasierte Programme auf ihre Flexibilität, Möglichkeiten in der Eingabe, die Bedienung, das optische Feedback und den gegebenen Mehrwert untersucht. Zu den 2D basierten Lösungen zählen das Electronic Cocktail Napkin, EsQUIsE, eParSketch und Flatland. Die getesteten 3D Lösungen waren ILoveSketch, CATIA Natural Sketch, Teddy und Chateau.

Jedes der bewerteten Konzepte bietet gegenüber der normalen Handskizze einen gewissen Mehrwert. Nutzer können von Analysen, Flächenberechnungen oder schnellen 3D-Modellen profitieren. Im Gegensatz dazu stehen Schwierigkeiten in der Bedienung oder Einschränkungen im Entwurfsablauf. Letztendlich hängt es von der persönlichen Präferenz ab, ob ein neues Tool akzeptiert und produktiv eingesetzt werden kann. Im Folgenden werden allgemeingültige Beobachtungen zusammengefasst. Jeder architektonische Entwurf ist ein Unikat, jeder Architekt hat eine eigene Art zu arbeiten. Für eine entwurfördernde Nutzung muss das genutzte Werkzeug demnach flexibel auf die unterschiedlichen Anforderungen reagieren können. Die getesteten digitalen Tools haben zusätzlich den Anspruch, einen Mehrwert im Vergleich zu traditionellen Werkzeugen zu schaffen. Grundsätzlich kann jedoch folgendes festgestellt werden: um einen Mehrwert für den Nutzer schaffen zu können, muss der Computer die Skizze verstehen. Reine Strichzeichnungen, die leicht von einem Menschen interpretiert werden können, müssen für den Computer in verständliche Informationen umgewandelt werden. Oftmals passiert dies aber auf Kosten der Flexibilität. Dies gilt auch für die getesteten, auf Handskizzen basierenden Konzepte. Das Electronic Cocktail Napkin oder EsQUIsE setzen zum Beispiel auf eine vorgegebene Symbolik für Skizzenelemente. Teddy oder ILoveSketch hingegen sind auf bestimmte Szenarien spezialisiert, um so die korrekte Funktion der Software zu garantieren. Beides zwingt dem Nutzer einen vorgegebenen Arbeitsprozess auf und senkt dadurch die Flexibilität des digitalen Werkzeuges im Vergleich zum Papier. Bei der Wahl des Entwurfswerkzeuges muss der Nutzer also entscheiden, ob er bereit ist, „Flexibilität“ zu opfern um dafür „Mehrwert“ zu erhalten. Für das zu schaffende Konzept galt es also, mit minimalen Eingriffen in die Arbeitsabläufe, die Skizzen gerade soweit für den Computer verständlich zu machen, dass dieser korrekt damit umgehen kann, um so einen Mehrwert gegenüber der Handskizze zu schaffen.

Nicht nur die vorgesehenen Arbeitsabläufe, auch die Interaktion mit der Software trägt maßgeblich zu ihrer effektiven Anwendung bei. Eine komplizierte Bedienung kann dazu führen, dass Nutzer das Werkzeug wechseln, selbst wenn die gegebene Funktionen die Anforderungen erfüllen (Microsoft, 2000)). Vor allem für entwurfsunterstützende Werkzeuge, ist es wichtig, dass sie wenig Aufmerksamkeit in der Bedienung benötigen, sodass der Nutzer sich ganz auf die Entwurfsaufgabe konzentrieren kann. Die getesteten Konzepte bemühen sich um eine simple Bedienoberfläche, und ersetzen zum großen Teil das bekannte WIMP (Window, Icon, Menu, Pointing Device) Prinzip mit Gesten und dem automatischen Erkennen gezeichneter Elemente. Funktionen, die einen Button benötigen, werden dadurch reduziert und die Interaktion mit der Skizze rückt in den Vordergrund. In ILoveSketch beispielsweise manipuliert der Nutzer seine Skizze komplett mit Hilfe des Stylus. Er kann mit ihm Elemente zeichnen, löschen oder verändern. Sogar die Zeichenebene oder Blickwinkel lassen sich innerhalb der Skizze festlegen. Die sehr direkte Interaktion sorgt für einen hohen Grad der Immersion und fördert so den Entwurfsgedanken. Ein wichtiger Teil der Interaktion ist auch die Art und Weise, wie der Computer mit skizzenhaften Eingaben des Nutzers umgeht. Bei den getesteten Programmen reicht die Bandbreite von unverfälschter Wiedergabe bis hin zu starker Verfälschung der Skizzen. Der Umgang hängt hierbei maßgeblich vom Verwendungszweck der Software ab. eParSketch zum Beispiel nutzt die Skizze um präzise CAD-Pläne zu erstellen. Entsprechend

werden ungenaue Striche begradigt, Ecken verbessert und Kreise perfektioniert. Im Gegensatz dazu steht etwa das Electronic Cocktail Napkin, welches als unterstützendes Werkzeug auf eine originalgetreue Darstellung der Eingaben setzt. Einige Programme unterstützen auch die Eingabe ungenauer Skizzen. Problematisch jedoch ist, dass spätestens wenn die unterstützenden Funktionen des Computers genutzt werden sollen, architektonischen Elemente eindeutig definiert werden müssen. Zusätzlich zu den Eingaben des Nutzers, stellen Programme, die als Design Decision Support dienen, weitere Informationen dar. Diese können einfache Zahlen, wie beispielsweise Raumgrößen sein, aber auch komplexe Simulationsergebnisse. Eine einfach verständliche Darstellung hilft bei der Interpretation und fördert somit das Visual Thinking. Es gilt auch abzuwägen, welche Informationen in welcher Genauigkeit für den Nutzer hilfreich sind. Der skizzenhafte Charakter der Darstellungen muss auch für die Visualisierung etwaiger Informationen beachtet werden.

Ebenfalls entscheidend ist die Hardwareschnittstelle zwischen Mensch und Computer. Größter Unterschied zum Papier ist, dass die Ein- und Ausgabe nicht direkt zwischen Stift und Papier geschieht. Vielmehr trennen weitere Faktoren, wie Stylus, Grafik-Tablet oder Monitor den Nutzer von einer direkten Interaktion mit der Skizze. Problematisch kann auch die Nutzung eines indirekten Zeigegerätes sein, beispielsweise einer Maus. Hierbei ist die manuelle Eingabe von der optischen Ausgabe getrennt, und führt so zu einer Entfremdung von Geste und Ergebnis. Moderne Grafik-Tablets begegnen diesem Problem mit der Integration von Bildschirmen, Fragen wie fehlende Haptik oder begrenzte Größen sind aber noch nicht zur Gänze gelöst (Schubert, 2014, S. 103). Setzt das Programm auf den dreidimensionalen Raum, treten zusätzliche Schwierigkeiten auf. Der Nutzer kann lediglich ein flächiges Interface nutzen, muss aber auch Tiefeninformationen angeben. ILoveSketch und CATIA Natural Sketch begegnen diesem Problem mit einer dauernden Neuausrichtung des Modells, sodass einzelne Teile normal gezeichnet werden können, Teddy hingegen ändert den gesamten Arbeitsablauf. Beides hindert durch die abstrahierte Eingabeform das Visual Thinking, sodass ein produktiver Ansatz zu hinterfragen ist.

5 Das Konzept

Das Fundament für den Lösungsansatz bildet das traditionelle Entwurfswerkzeug Handskizze. Diese wird über eine geeignete Schnittstelle digitalisiert und mit entwurfsunterstützenden Informationen ergänzt. Aus den gezeichneten Grundrissen lassen sich zudem Schnitte generieren, die in Folge weiter bearbeitet werden können. Der Entwurfsprozess ist so in drei Dimensionen möglich. Auf Basis der Skizzen sind verschiedene, unterstützende Analysen und Simulationen möglich. So kann sich der Planer schrittweise an eine Lösung für seine Entwurfsaufgabe herantasten und sich dabei auf objektive Informationen beziehen.

Das Interface stellt dem Nutzer eine Arbeitsfläche zur Verfügung, auf der wie vom Papier gewohnt gezeichnet werden kann. Auf einer zweiten Arbeitsfläche lässt sich eine alternative Ansicht darstellen. Dadurch ist es beispielsweise möglich, zeitgleich einen Grundriss und einen dazu passenden Schnitt anzuzeigen, um so beide simultan zu bearbeiten. Der Nutzer muss lediglich das gewünschte Werkzeug aussuchen, und kann dann sofort das Skizzieren beginnen. Er kann beispielsweise aus den Tools Wand, Fenster oder Tür wählen. Das gewählte Tool bestimmt wie der Rechner mit den gezeichneten Elementen umgeht, und ist ausschlaggebend für die Generierung von Schnitten und die korrekte Funktion von Simulationen. Ein weiterer entscheidender Punkt ist die Applikation und Steuerung der Analysen und Simulationen. Wichtig dabei ist eine schnelle und verständliche Bedienung. Erreicht wird dies, durch

eine Simplifizierung der Funktionen und eine Reduktion der einstellbaren Parameter. Dadurch wird die Anwendung und Auswertung dem skizzenhaften Charakter der zugrunde liegenden Darstellung gerecht.

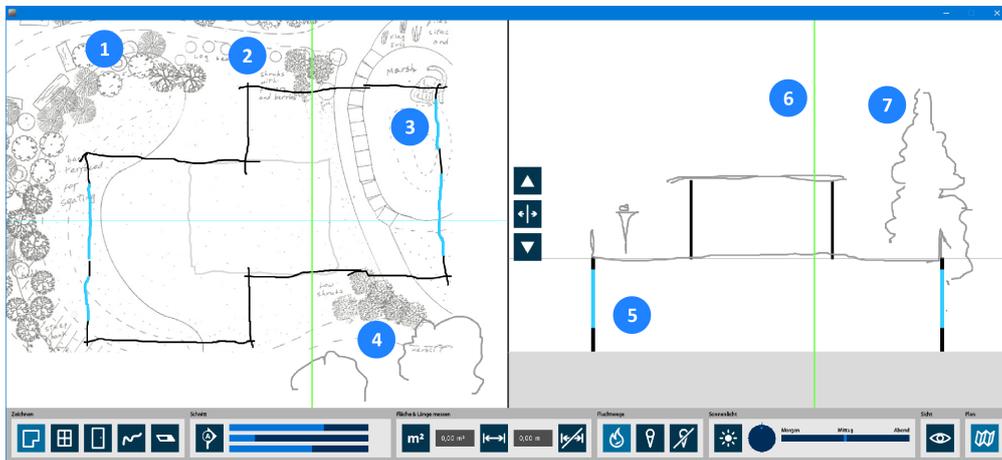


Abb. 1: Übersicht über den Prototyp (1. Angezeigter Plan im Hintergrund, 2. Skizzierte Wand, 3. Fenster, 4. Mit dem Scribble Tool gezeichnete Elemente, 5. Wand im automatisch generierten Schnitt mit Fenster, 6. Guidelines, 7. Scribble im Schnitt)

Um die Flexibilität des Papiers auch am Rechner zu erhalten, gibt das Konzept keine Arbeitsweise vor. Der Planer kann je nach Situation und Vorliebe nur im Schnitt arbeiten, ausschließlich Grundrisse nutzen, oder alle Ansichten kombinieren. Auch gibt es keine spezielle Ikonographie für bauliche Elemente zu beachten, lediglich das korrekte Tool muss gewählt werden. Dadurch treten keine Probleme auf, die bei auf Symbolerkennung von abstrakter Geometrie basierender Software existieren. Individuelle Stile und Symboliken sind problemlos möglich und der beim Skizzieren gegebene, riesige Umfang in Syntax und Semantik wird abgedeckt. Dem Nutzer sind so praktisch keine Regeln vorgegeben und er kann eigene, vertraute Arbeitsprozesse nutzen. Wengleich die Zeichenfunktionen den Grundstein für die Funktion des Konzeptes legen, sind es die Analysen & Simulationen, die den entscheidenden Mehrwert gegenüber dem normalen Papier bieten. Durch die Verschmelzung von Skizze und Analyse stehen dem Nutzer ohne ein Wechsel des Mediums integrierte, digitale Hilfsmittel zu Verfügung. Entwurfsrelevante Informationen können so bereits früh im Denkprozess für die Entscheidungsfindung genutzt werden. Für den zu programmierenden Prototypen wurden beispielhaft die Funktionen Raumgröße, Tageslicht und Fluchtwege implementiert.

6 Implementierung des Prototyps

Die Implementierung des Prototyps hatte einen „Proof-of-Concept“-Charakter, der Funktionsumfang und die Gestaltung des Interfaces ist demnach nicht vollständig. Dennoch sollte der Prototyp die zuvor ausgearbeiteten Prinzipien vollständig darstellen, um so feststellen zu können, ob das Programm wie gewünscht funktioniert und den Planer bei seiner Entwurfsarbeit unterstützen kann. Die umgesetzten Analysen & Simulationen sind lediglich beispielhaft zu sehen, für eine ausgearbeitete Version gibt es noch viele andere mögliche Anwendungen. Der Prototyp wurde in der objektorientierten Programmiersprache Processing geschrieben, welche auf Java basiert. Processing wurde als Programmiersprache gewählt, da sie sich vornehmlich

an Künstler und Programmieranfänger richtet. Die Community dieses Open Source Projektes hat sich zudem als wertvolle Hilfe erwiesen.

Der Prototyp bietet die Zeichentools Wand, Fenster, Tür, Scribble und Radierer. Zusätzlich dazu kann der Nutzer verschiedene Analysen und Simulationen verwenden. Dazu gehören: Raumgröße messen, Tageslichtanalyse, Fluchtwegsimulation und eine Sichtlinienanalyse. Zudem kann der Nutzer auf verschiedenen Stockwerken zeichnen, Guidelines zu Orientierung verwenden und direkt aus dem gezeichneten Grundriss Schnittansichten erstellen. Die Grundlage für alle Simulationen und Analysen in der Skizze, bildet ein unter allen Elementen liegendes Zellenraster. Jede Zelle ist ein Objekt mit Abmessungen von 10×10 Pixeln, und speichert basierend auf den darüber gezeichneten Skizzen eine Vielzahl wichtiger Daten ab. Jede Zelle enthält etwa die Information, was für ein Element sie enthält, ob Wand, Fenster oder Tür.

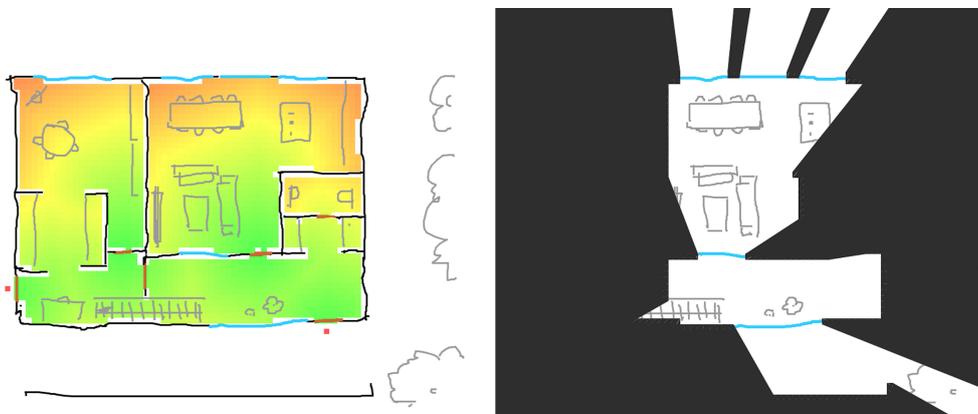


Abb. 2: Visualisierung der Fluchtwege und der Sichtbeziehungen

Der Planer kann direkt in der Skizze Sichtbeziehungen visualisieren, indem durch Wände oder Türen verdeckte Elemente farblich überdeckt werden. Von einer zuvor gewählten Position aus werden Vektoren zu allen Zellen erstellt, welche die Information „Wand“ oder „Tür“ enthalten. Auf Basis dieser Vektoren wird eine flächige Form erstellt, welche dahinterliegende Elemente, ähnlich einem Schatten, überdeckt. Um direkt in der Zeichnung zu erkennen, ob der Brandschutz erfüllt wird, kann der Nutzer mögliche Fluchtwege simulieren. Bereiche die zu weit vom nächsten Ausgang entfernt sind, werden farblich hervorgehoben. Für die Simulation der Fluchtwege wird der A-Stern Algorithmus angewendet, ein Wegfindungsalgorithmus auf Basis des Zellenrasters. A-Stern berechnet für jede Zelle einen Entfernungswert zu einem gegebenen Ziel, wobei Wände und Fenster umgangen, Türen jedoch genutzt werden. Der berechnete Wert gibt die farbliche Darstellung der Zelle vor, und ist so für das optische Feedback verantwortlich. Die Tageslichtsimulation erlaubt dem Nutzer sofort in der Skizze zu sehen, wie sich die Positionierung und Dimensionierung von Fenstern auf die Belichtung des Innenraumes auswirkt. Dabei kann nicht nur die Nordung der Skizze verändert werden, auch die Tageszeit lässt sich einstellen. Um eine realistischere Darstellung von Licht und Schatten zu erhalten, wird nicht nur die Sonne als Lichtquelle genutzt, auch das Umgebungslicht wird verwendet.

Um Raumgrößen zu bestimmen sowie um festzustellen, ob sich ein gezeichnetes Element im Innen- oder Außenraum befindet, wird ein auf dem Zellenraster basierender Füll-Algorithmus verwendet. Dabei wird von einer Start-Zelle ausgehend untersucht, ob die Nachbarzellen passierbar sind, also keine Zeichnerischen Elemente wie eine Wand oder ein Fenster enthalten. Falls ja, werden auch deren Nachbarzellen untersucht. Dieser Prozess wiederholt sich so lange,



Abb. 3: Direkte Belichtung und Umgebungslicht im Vergleich

bis entweder alle Zellen untersucht wurden, oder der der Füll-Algorithmus sich nicht mehr ausbreiten kann.

7 Evaluierung

Durch eine Evaluierung sollte die Nutzbarkeit des Konzeptes mit Hilfe des Prototyps getestet werden. Von Interesse war dabei nicht nur, wie gut unerfahrene Nutzer mit dem Tool umgehen können, sondern auch, ob und wie das digitale Hilfsmittel den Entwurfsprozess beeinflusst. Da sowohl der Entwurfsprozess, als auch die Nutzung von Software auf persönlichen Arbeitsweisen, Erfahrungen und Präferenzen basiert, fußte diese Evaluierung nicht nur auf quantitativem Feedback, sondern bezog auch qualitative Aussagen mit ein. Die Hypothese für diese Evaluierung war, dass das neue Tool das klassische Entwurfswerkzeug Handskizze um zusätzliche Informationen ergänzt, und so den Architekten im Entwurfsprozess unterstützt. Gleichzeitig sollte es möglichst geringe Einschränkungen in der Nutzung gegenüber der analogen Handskizze geben. Die Teilnehmer mussten dieselbe Aufgabe einmal mit Hilfe des neuen Tools und einmal klassisch mit Stift und Papier lösen. Insgesamt 10 Personen haben an der Evaluation teilgenommen. Dabei mussten alle eine gegebene Entwurfsaufgabe mit Hilfe des neuen Tools lösen. Die Testpersonen kannten das Projekt nicht oder nur am am Rande.

Am Ende jeder Sitzung mussten die Testpersonen einen vorbereiteten Fragebogen ausfüllen. Teil dessen war eine SUS Umfrage (System Usability Scale), die eine erste, schnelle Bewertung der Software-Usability ermöglicht. Des Weiteren bestand der Fragebogen aus einer AttrakDiff Umfrage, welche die Attraktivität interaktiver Produkte messen kann. Zuletzt hatten die Testpersonen selber die Möglichkeit, Verbesserungsvorschläge oder persönliche Kommentare festzuhalten. Die Auswertung der Evaluierung bestand aus quantitativem Feedback, also absolute Zahlen aus den SUS und AttrakDiff Umfragen, sowie der zeitlich gemessenen Nutzung der Analyse und Simulationsfunktionen. Das qualitative Feedback beinhaltete Beobachtungen während der Nutzung sowie die gemachten Verbesserungsvorschläge.

Mit dem AttrakDiff Test kann sowohl die wahrgenommene pragmatische, als auch die hedonische Qualität und die Attraktivität eines interaktiven Produktes erfasst werden. Die Analyse aller Werte erlaubt, die Begehrtheit eines interaktiven Produktes anzuzeigen. Das Konfidenz-Rechteck zeigt zusätzlich zur mittleren Ausprägung der Werte, wie einig sich die Probanden

in ihrer Bewertung sind. Die in AttrakDiff erhaltenen Werte zeigen, dass bereits der Prototyp in Richtung „Begehrt“ tendiert. Zudem wird er weder als zu selbstorientiert, noch als zu handlungsorientiert eingestuft. Das recht kleine Konfidenzrechteck deutet darauf hin, dass sich die Probanden in der Bewertung sehr einig waren. Aus dem Diagramm der Mittelwerte lässt sich ablesen, dass das System sowohl in der pragmatischen Qualität, als auch in der hedonischen Qualität positiv aufgenommen wurde. Die Allgemeine Attraktivität wurde im Vergleich sogar noch etwas höher eingestuft. Man kann daraus schließen, dass der Prototyp im Grunde funktioniert und positiv gesehen wird, die Umsetzung aber noch verbessert werden muss.

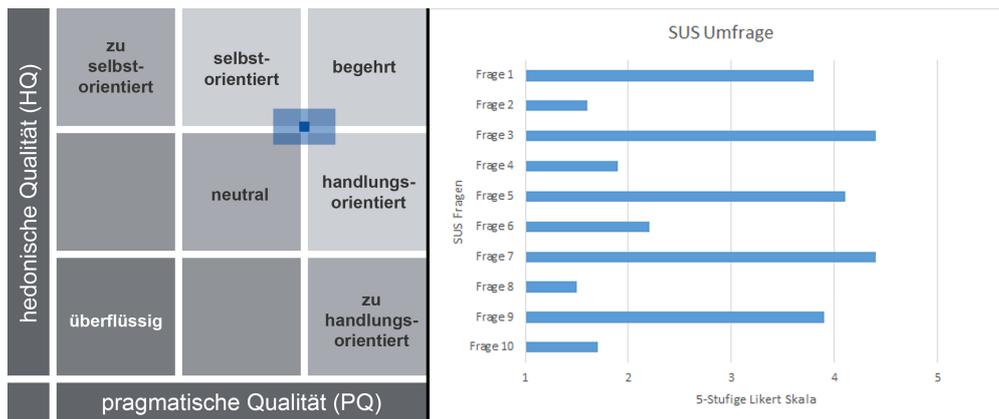


Abb. 4: AttrakDiff Auswertung & SUS-Survey Auswertung (Fragen: 1. Ich denke, dass ich dieses System gerne regelmäßig nutzen würde, 2. Ich fand das System unnötig komplex, 3. Ich denke, das System war leicht zu benutzen, 4. Ich denke, ich würde die Unterstützung einer fachkundigen Person benötigen, um das System benutzen zu können, 5. Ich fand, die verschiedenen Funktionen des Systems waren gut integriert, 6. Ich halte das System für zu inkonsistent, 7. Ich glaube, dass die meisten Menschen sehr schnell lernen würden, mit dem System umzugehen, 8. Ich fand das System sehr umständlich zu benutzen, 9. Ich fühlte mich bei der Nutzung des Systems sehr sicher, 10. Ich musste viele Dinge lernen, bevor ich mit dem System arbeiten konnte.)

Im SUS-Test müssen die Testpersonen zehn Fragen mit Hilfe einer fünfstufigen Likert-Skala beantworten. Dabei wechseln sich positiv und negativ formulierte Fragen ab. Insgesamt können 40 Punkte erreicht werden, die für eine finale Bewertung mit 2,5 multipliziert werden, und so maximalen Punktwert von 100 ermöglichen. Der in der SUS Umfrage erreichte Usability-Wert beträgt 79,25 Punkte (Max 100 Punkte, > 68 = gute Nutzbarkeit) und deutet auf eine mindestens gute Nutzbarkeit des Prototyps. Trotz der nur sehr kurzen Einführung, fand die Mehrzahl der Probanden das System einfach zu verwenden. Die Integration der verschiedenen Funktionen ist bisher nur prototypisch gelöst, was sich auch in dem etwas schlechteren Abschneiden der Frage 5 äußert. Folgerichtig waren sich die Nutzer zum Teil auch in der Bedienung nicht immer sicher. Dennoch war der Prototyp insgesamt einfach zu erlernen, und gut zu nutzen, weshalb die Mehrheit der Probanden das System regelmäßig nutzen würde.

Die gemessenen Daten zeigen, dass die Zeichentools mit einem Anteil von 70% am meisten genutzt wurden. Die Simulationen hingegen kommen zusammen auf etwa 30%. Davon fallen ca. 18% auf die alleinige Nutzung der Simulationen, 12% der Dauer wurden sie in Kombination mit einem Zeichentool genutzt. Dies deckt sich auch mit den gemachten Beobachtungen. Die Probanden prüften zwar häufig ihre Pläne mit Hilfe einer Simulation, wechselten dann aber wieder zurück zu den Zeichentools. Die Kombination von Zeichnung und Analyse wurde unter anderem nur deshalb selten genutzt, da die Performance des Prototyps erheblich darunter

litt. Zudem kann die Tageslichtanalyse wegen ihrer Visualisierung die Lesbarkeit der Skizze einschränken. Insgesamt lässt sich festhalten, dass vor allem die gleichzeitige Nutzung von Zeichentools und Werkzeugen verbessert werden muss. Für eine zukünftige Version des Prototyps muss eine bessere Integration der Simulationen angestrebt werden, sowohl visuell, als auch von der Nutzung.

8 Fazit und Ausblick

In dem vorliegenden Beitrag wurde ein Konzept für die digitale Nutzung von Handskizzen in Kombination mit Analysen und Simulationen vorgestellt. Ausgegangen wurde dabei von der bisherigen Nutzung der Skizze als Entwurfswerkzeug, und von der Anwendung von Analyse-Tools im Designprozess. Zudem sind Erfahrungen aus der Nutzung themenverwandter Lösungen in das formulierte Konzept mit eingeflossen.

Das vorgestellte Konzept erfüllt den Großteil der gestellten Anforderungen, und ermöglicht so eine funktionierende Kombination des etablierten Entwurfswerkzeugs Skizze mit digitalen Hilfsmitteln. Es kommt während der Entwurfsphase zu keinem Wechsel der Entwurfsmittel, wodurch ein ungestörter Denkprozess ermöglicht wird. Das Visual Thinking wird nicht nur durch die Skizze gefördert, sondern zusätzlich durch optisches Feedback der Simulationen und Analysen angeregt. Der Architekt kann, gestützt durch objektive Daten, Entwurfsentscheidungen treffen und begründen. Möglicherweise unbekannte Problemstellen, etwa in der natürlichen Belichtung, können dank eindeutiger Visualisierungen schnell erkannt und direkt in der Skizze behoben werden. Die Evaluierung hat gezeigt, dass auch ungeübte Nutzer leicht mit dem Tool umgehen konnten, und die zusätzlichen Hilfsmittel verwendeten. Zudem kam bereits der Prototyp mit einer Vielzahl unterschiedlicher Zeichenstile zurecht.

Für eine ausgearbeitete Version der Software gibt es dennoch viel Verbesserungspotenzial. Um trotz zusätzlicher Funktionen eine einfache Bedienung zu gewährleisten, sollte auch der Bereich Interaktion und Graphical User Interface verbessert werden. Viele Buttons etwa lassen sich mit bei Tablets üblichen Gesten ersetzen, zum Beispiel „Pinch-to-Zoom“ oder mit drei Fingern wischen, um die Zeichenfläche zu bewegen. Dadurch ist auch eine einfachere Interaktion mit der Skizze möglich. Gesten können auch in der Gestaltung des GUI angewendet werden. Ebenso lassen sich die Werkzeuge und die Analysen verbessern. Durch die Nutzung eines hochwertigen Stylus etwa kann der Rechner zusätzliche Informationen bezüglich Druckstärke oder Zeichenwinkel erhalten, und so das digitale Skizzieren verbessern. Auch das Platzieren von Fenstern und Türen muss verbessert werden, um eine korrekte Funktion der Simulationen selbst bei sehr ungenau skizzierten Grundrissen zu gewährleisten. Eine Versionierungsfunktion kann den Nutzer dabei unterstützen, verschiedene Stände seiner Arbeit einfach abrufbar zu halten. Wichtig ist dies vor allem, wenn mehrere Personen gleichzeitig an einem Projekt arbeiten. Bisher nicht angesprochen, aber nicht minder wichtig ist der Ausbau kollaborativer Funktionen, die mehreren Personen die Möglichkeit geben, gleichzeitig an einem Projekt zu arbeiten. Da sämtliche Eingaben in digitaler Form vorliegen, bietet es sich an, gemeinsames Arbeiten mit Hilfe des Internets ortsunabhängig zu machen.

Literatur

- Bailey, R. (2002). „The Intelligent Sketch Developing a conceptual model for a digital design assistant“. In: Mark J. Clayton (Hg.): *Eternity, infinity and virtuality in architecture. ACADIA 2000*. Washington DC, S. 137–146.
- Do, E. Y.-L. (1999). „The right tool at the right time. Investigation of freehand drawing as an interface to knowledge based design tools“. In:
- Eisenman, P. und Schwarz, U. (1995). „Aura Und Exzess Zur Überwindung der Metaphysik der Architektur“. In: *Passagen-Verl. (Passagen Architektur)*.
- Goldschmidt, G. (1994). „On visual design thinking the vis kids of architecture“. In: *Design studies* 15.2. DOI: 10.1016/0142-694X(94)90022-1., S. 158–174.
- Gross, M. D. und Do, E. Y.-L. (1996). „Ambiguous intentions a paper-like interface for creative design“. In: *UIST '96. Ninth Annual Symposium on User Interface Software and Technology: proceedings of the ACM Symposium on User Interface Software and Technology, Seattle, Washington, USA*, ACM Press, New York, N, S. 183–192.
- Lin, S.-H. und Gerber, D. J. (2014). „Evolutionary energy performance feedback for design. Multidisciplinary design optimization and performance boundaries for design decision support“. In: *Energy and Buildings* 84. DOI: 10.1016/j.enbuild.2014.08.034., S. 426–441.
- Microsoft (Hg.) (2000). *Usability in Software Design*. <https://msdn.microsoft.com/en-us/library/ms997577.aspx>.
- Peichl, G., Barkhofen, E.-M. und Maak, N. (2013). *Gustav Peichl - Die Zeichnung ist die Sprache der Architekten*. Akad. der Künste. Berlin: Gustav Peichl.
- Sarhan, A. und Rutherford, P. (2014). „Integrating sustainability in the architectural design education process. Taxonomy of challenges and guidelines.“ In: *Fusion-Proceedings of the 32nd eCAADe Conference*. Bd. 1. Newcastle upon Tyne: Department of Architecture, Built Environment, Faculty of Engineering und Environment, S. 323–332.
- Schubert, G. (2014). „Interaktionsformen für das digitale Entwerfen: Konzeption und Umsetzung einer rechnergestützten Entwurfsplattform für die städtebaulichen Phasen in der Architektur“. Diss. Technische Universität München, München. Fakultät für Architektur.
- Seichter, H. (2002). „Die architektonische Skizze in virtuellen Umgebungen“. Diplomarbeit. Bauhaus Universität Weimar, Weimar. Lehrstuhl für Informatik in der Architektur.
- Suwa, M. und Tversky, B. (1997). „What do architects and students perceive in their design sketches? A protocol analysis“. In: *Design studies* 18.4, S. 385–403. DOI: 10.1016/S0142-694X(97)00008-2.
- Tversky, B. (1999). „What does drawing reveal about thinking?“ In: *Visual and Spatial Reasoning in Design*. Hrsg. von J. S. Gero und B. Tversky. Sidney: Key Centre of Design Computing und Cognition, S. 93–101.
- Vanegas, C. A., Aliaga, D. G., Benes, B. und Waddell, P. (2009). „Visualization of simulated urban spaces: Inferring parameterized generation of streets, parcels, and aerial imagery“. In: *IEEE Transactions on Visualization and Computer Graphics* 15.3, S. 424–435. DOI: 10.1109/TVCG.2008.193.

Integrierte Visualisierung von Installationsplänen und Installationskosten beim Glasfaser-Breitbandausbau

Jens Wiggenbrock

*Professur Informatik im Bauwesen, Bauhaus-Universität Weimar, Coudraystr. 7, 99423 Weimar
E-Mail: jens.wiggenbrock@uni-weimar.de*

Abstract Durch die zunehmende Nutzung von Internet-basierten Diensten ist der Bedarf an leistungsfähigen Hochgeschwindigkeits-Glasfasernetzen in den letzten Jahren rapide gestiegen. Eine umfassende Planung von neu zu errichtenden Glasfasernetzen im Sinne eines kostengünstigen Breitbandausbaus ist jedoch häufig mit hohem Aufwand verbunden. Die Auswahl möglicher Kabelwege und die anschließende Kalkulation der Installationskosten werden heute in der Regel durch computerbasierte Verfahren unterstützt, wobei allerdings die Installationspläne und die Installationskosten getrennt voneinander dargestellt werden, was wiederum die computergestützte Planung sowie weitere Optimierungsansätze erschwert. Dieser Artikel beschreibt ein modulares Softwaresystem zur computergestützten Planung, Kostenkalkulation und Visualisierung von Glasfasernetzen, das eine integrierte Darstellung von Installationsplänen und Installationskosten ermöglicht. Anstelle der herkömmlichen Darstellung von Installationsplänen, die in der Regel alle geplanten Kabelwege einfarbig auf einer Landkarte visualisiert, wird ein Farbschema zur Anzeige der Installationskosten in Installationsplänen eingesetzt. Der beschriebene Ansatz wird durch ein Anwendungsbeispiel, das die Planung eines Glasfasernetzes innerhalb eines Siedlungsgebietes behandelt, validiert. Die integrierte, farbige Darstellung der Installationskosten in Installationsplänen ermöglicht eine effiziente Identifikation der kostenintensiven Bauabschnitte und befördert kostenoptimierte Planungsansätze. Die intuitive Visualisierung vereinfacht somit die akkurate und kostenoptimierte Planung von Glasfasernetzen.

Keywords: Breitbandausbau, Glasfasernetz, Visualisierung, Infrastrukturplanung, Kostenoptimierung, Ampelmodell

1 Einleitung

Durch die zunehmende Nutzung von Internet-basierten Diensten ist der Bedarf an leistungsfähigen Hochgeschwindigkeits-Kommunikationsnetzen gestiegen (Cisco Systems, 2015). Während die bestehenden Backbone-Verbindungen zwischen den Internet-Hauptknoten den zukünftigen Bedarf weiterhin abdecken, sind erhebliche Anpassungen und Verbesserungen innerhalb der Zugangsnetze nötig (Cisco Systems, 2015; DE-CIX, 2016). Der Breitbandausbau von Glasfasernetzen im Zugangsnetzbereich ist, im Hinblick auf weiter steigende Datenvolumen, zukunftsorientiert und notwendig (Kaminow et al., 2010). Innerhalb von Siedlungsgebieten können die neu zu errichtenden Glasfasernetze verschiedene Netztopologien aufweisen. Eine Netztopologie,

die jedes postalisch adressierbare Gebäude mit einem zentralen Verteilerpunkt durch mindestens eine separate, durchgehende Glasfaser verbindet, wird als „Fiber-to-the-Building (FTTB)“-Netztopologie bezeichnet. Ein Glasfasernetz in FTTB-Netztopologie ist leistungsfähig und flexibel, beispielsweise hinsichtlich späterer Erweiterungen aufgrund einer Nachverdichtung der Siedlungsstruktur, jedoch dadurch auch sehr kostenintensiv. Eine sorgfältige Planung des Glasfasernetzes führt zu einer Reduzierung der Installationskosten (U.S. Department of Transportation, 2014). Die Planung wird in einzelnen Planungsstufen (Vorplanung, Detailplanung, Ausführungs- und Genehmigungsplanung) durch verschiedene Methoden und Softwareanwendungen unterstützt. Die detaillierte Planung eines Glasfasernetzes berücksichtigt viele Planungsparameter, unter anderem die Positionen von Gebäuden sowie den Verlauf von Straßen und Wegen (Comsof NV, 2014; IT-Simplicity, 2014). Im Allgemeinen ist das Ergebnis einer detaillierten Glasfasernetzplanung ein Installationsplan, der die geplanten Kabelwege visualisiert und eine Auflistung, die die zu erwartenden Installations- und Materialkosten darstellt. Installationspläne repräsentieren Kabelwege und Verbindungsstellen der geplanten Glasfaserkabel üblicherweise auf einer Landkarte. Im Allgemeinen werden einzelne Kabelwege innerhalb der Installationspläne durch einfarbige Linien dargestellt. Werden farbige Linien innerhalb der Installationspläne verwendet, so dienen diese für gewöhnlich der Kennzeichnung und Unterscheidung von einzelnen Kabeladern (oder einzelnen Glasfasern) innerhalb eines Kabels (Woodward und Husson, 2002). Neben den zuvor beschriebenen Darstellungsarten kann die farbliche Darstellung auch die Zugehörigkeit der Kabel zu einem gemeinsamen Verteilerknoten (in Telekommunikationsnetzen die so genannten Kabelverzweiger „KVz“) kennzeichnen. Alle zuvor beschriebenen Darstellungsarten haben jedoch gemeinsam, dass die Installationspläne und die Installationskosten dem Planer separat präsentiert werden. Die getrennte Präsentation von Installationsplänen und Installationskosten erschwert jedoch die Analyse der kostenintensivsten Kabelabschnitte durch den Planer.

Dieser Artikel stellt ein modulares Softwaresystem zur computergestützten Planung von Glasfasernetzen vor. Das modulare Softwaresystem beinhaltet ein Visualisierungsmodul zur integrierten Darstellung von Installationsplänen und Installationskosten. Unter der integrierten Darstellung wird im Folgenden die gleichzeitige Lagezeichnung der Kabelwege und Anzeige der jeweiligen Installationskosten durch Farben verstanden. Das Ampelmodell, im englischen unter dem Begriff RAG-System (engl. red, amber, green) bekannt, dient als flexibler Ansatz zur Darstellung der Installationskosten in Installationsplänen (Wiggenbrock und Smarsly, 2016). Die kombinierte Darstellung wird bei der Analyse der kostenintensivsten Kabelabschnitte als Basis für Optimierungsansätze genutzt.

In diesem Artikel wird zuerst das modulare Softwaresystem allgemein beschrieben. Anschließend wird das Visualisierungsmodul des Softwaresystems im Detail dargestellt. Abschließend wird in einem Anwendungsbeispiel, das die Planung eines Glasfasernetzes innerhalb eines Siedlungsgebietes behandelt, das modulare Softwaresystem – insbesondere die integrierte Visualisierung von Installationsplänen und Installationskosten – validiert.

2 Ein modulares Softwaresystem für die Glasfasernetzplanung

Das modulare Softwaresystem für die Glasfasernetzplanung ist darauf ausgelegt, die folgenden drei Hauptverarbeitungsschritte auszuführen: 1) Datenerfassung, 2) Räumliche Analyse und Planung eines Glasfasernetzes auf Basis der erfassten Daten und 3) Visualisierung sowie Export der Planungsdaten. In Anlehnung an die drei genannten Hauptverarbeitungsschritte ist das

Softwaresystem in drei Subsysteme untergliedert; das Eingabe-Subsystem, das Berechnungs-Subsystem und das Ausgabe-Subsystem. Jedes Subsystem beinhaltet, wie in Abbildung 1 dargestellt, weitere Module, welche die Aufgaben des jeweiligen Subsystems unterstützen.

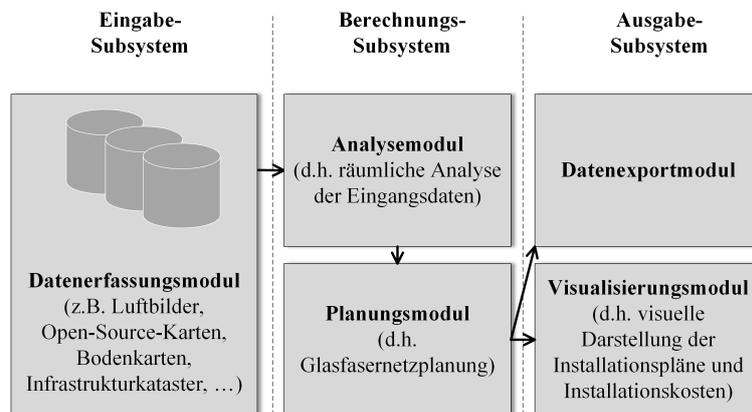


Abb. 1: Subsysteme des modularen Softwaresystems zur Glasfaserplanung.

Das Eingabe-Subsystem beinhaltet Module zur Datenerfassung aus verschiedenen Internet-basierten Datenquellen (mehrheitlich Open-Source, wie zum Beispiel „OpenStreetMap“). Jede Datenquelle wird über ein eigenes Modul angesprochen, das die Datenerfassung (Download) sowie die Integration und Bereinigung der Daten übernimmt. Die Integration und Bereinigung der vielfältig verfügbaren, jedoch heterogenen Daten gewährleistet, dass nur die für die Glasfaserplanung relevanten Daten in das modulare Softwaresystem übernommen werden.

Das Berechnungs-Subsystem enthält Module für die räumliche Analyse der Daten („Analysemodul“) und für die Planung von Glasfasernetzen („Planungsmodul“). Das Analysemodul, wie in Abbildung 1 dargestellt, übernimmt die integrierten Daten des Eingabe-Subsystems, unter anderem die Straßen innerhalb des Planungsgebietes, und generiert einen Routing-Graphen zur Bestimmung möglicher Kabelwege. Der Routing-Graph enthält gewichtete Verbindungen, bestehend aus den Positionen der einzelnen Straßenabschnitte und den zu erwartenden Installationskosten, die sich aus der Boden- und Oberflächenbeschaffenheit der Straße ergeben. Das Planungsmodul nutzt den vom Analysemodul erstellten Routing-Graphen zur Berechnung einer kostenoptimalen Glasfaser-Netztopologie. Die berechnete Glasfaser-Netztopologie besteht aus verschiedenen Kabelwegen, von jedem Gebäude zu einem zentralen Verteilerknoten. Die Kabelwege wiederum bestehen aus Leitungsabschnitten, analog zu den Verbindungen im Routing-Graphen und deren jeweilig zugewiesenen Installationskosten. Die Installationskosten jedes Leitungsabschnitts ergeben sich im Wesentlichen durch die Addition der Materialkosten für das Glasfaserkabel und die Arbeitskosten für den Tiefbau zur unterirdischen Verlegung der Kabel in einem Kabelgraben (Wiggenbrock et al., 2015a,b). Ein Kabelgraben kann mehrere Kabel aufnehmen, sodass die Tiefbaukosten bei der Installationskalkulation nur einmal berechnet werden. Abschließend werden die Planungsdaten des Glasfasernetzes, die Kabelwege, die notwendigen Kabelgräben und die zugehörigen Installationskostenden für die weitere Verarbeitung bereitgestellt.

Das Ausgabe-Subsystem übernimmt schließlich die von dem Berechnungs-Subsystem bereitgestellten Planungsdaten. Das Ausgabe-Subsystem beinhaltet verschiedene Module zur Benutzerinteraktion. Dazu zählt neben einem Modul für den Datenexport (z.B. Tabellenkalkulationsprogramm) das Visualisierungsmodul, das dazu dient, die Daten des Analysemoduls und

des Planungsmoduls auf einer Landkarte anzuzeigen. Das Visualisierungsmodul generiert aus den Planungsdaten eine integrierte Darstellung der Installationspläne und Installationskosten, und wird im folgenden Abschnitt vorgestellt.

3 Integrierte Visualisierung von Installationsplänen und Installationskosten – Visualisierungsmodul

Das Ziel des Visualisierungsmoduls ist die integrierte Darstellung von Installationsplänen und Installationskosten. Die integrierte, intuitive Präsentation der Planungsdaten verbessert die Effizienz und die Genauigkeit im Sinne der kostenoptimierten Planung von Glasfasernetzen.

Wie beschrieben werden die Planungsdaten (Kabelwege, Kabelgräben und die zugehörigen Installationskosten) von dem Visualisierungsmodul übernommen. Die Nutzung einer Farbkodierung stellt einen klaren und intuitiven Ansatz zur Anzeige der Installationskosten innerhalb von Installationsplänen dar. Bei der Erstellung von Temperaturkarten innerhalb von Wetterberichten wird die Farbkodierung bereits erfolgreich eingesetzt. Dabei werden Temperaturen zwischen -20 °C und 40 °C in der Regel durch unterschiedliche Farben (violett, blau, grün, gelb, rot) mit mehreren Abstufungen angezeigt. Aufgrund des großen Wertebereichs (kalt, kühl, behaglich, warm, heiß) sind viele unterschiedliche Farben und Abstufungen notwendig. In der Glasfaserplanung ist das Farbschema, wie es bei Temperaturkarten verwendet wird, für die klare und intuitive Darstellung der Installationskosten innerhalb eines Installationsplans zu umfangreich. Deshalb wird in dieser Studie das sogenannte Ampelmodell (RAG-System) als Farbschema verwendet. Das Ampelmodell wird bereits in verschiedenen Anwendungs- und Forschungsbereichen, wie zum Beispiel im Projektmanagement und der Ökotoxikologie als „Statusanzeige“ genutzt (Albert, 2009; Pritchard, 2013). Im Ampelmodell wird das Verhältnis zwischen dem Istwert und einem Referenzwert durch die Farben rot (hohe Differenz), gelb (mittlere Differenz) und grün (geringe Differenz) angezeigt. Die erforderlichen Schaltschwellen für die Farbwechsel (sog. Toleranzgrenze=rot/gelb und Akzeptanzgrenze=gelb/grün) müssen zuvor definiert werden.

Der in dieser Studie vorgestellte Ansatz zur integrierten Darstellung von Installationskosten in Installationsplänen nutzt das zuvor beschriebene Ampelmodell zur Anzeige der Installationskosten in Bezug auf definierte Schaltschwellen. Die Schaltschwellen können durch den Planer anhand der Eigenschaften des Planungsgebietes frei gewählt, oder aus den Netzparametern automatisch berechnet werden. Als einen ersten Richtwert für die Definition der Schaltschwellen können die durchschnittlichen Installationskosten pro Meter verwendet werden. Diese berechnen sich aus den Gesamtkosten des Glasfasernetzes, dividiert durch die Gesamtlänge der zu verlegenden Glasfasern. Im weiteren Verlauf des Artikels werden die durchschnittlichen Installationskosten pro Meter als Referenzwert für das Ampelmodell verwendet. Die Schaltschwellen für den Farbwechsel zwischen grün/gelb wird auf 100 % und die Schaltschwelle gelb/rot wird auf 200 % des Referenzwertes gesetzt. Dadurch werden alle Kabelabschnitte, deren Installationskosten weniger als die durchschnittlichen Installationskosten betragen, grün dargestellt. Kabelabschnitte, deren Installationskosten zwischen 100 % und 200 % der durchschnittlichen Installationskosten betragen, werden gelb dargestellt. Kabelabschnitte, deren Installationskosten 200 % der durchschnittlichen Installationskosten überschreiten, werden rot dargestellt.

Abbildung 2 zeigt schematisch einen Auszug aus dem Visualisierungsmodul als Schnittstelle zwischen dem Planungsmodul und der visuellen Darstellung innerhalb der grafischen Benutzerumgebung. Sobald die Planungsdaten des Planungsmoduls bereitgestellt wurden, berechnet das Visualisierungsmodul die durchschnittlichen Installationskosten als allgemeinen

Referenzwert für das Ampelmodell. Anschließend wird allen Kabelabschnitten entsprechend der berechneten Installationskosten und den eingestellten Schaltschwellen ein Farbwert des Ampelmodells zugewiesen. Der Prozess endet mit der Serialisierung der Datensätze in einer JSON-Datei (JavaScript Object Notation). Das JSON-Format ist ein häufig genutztes, textbasiertes Datenformat, das den Datenaustausch über das Internet vereinfacht (ECMA International, 2015).

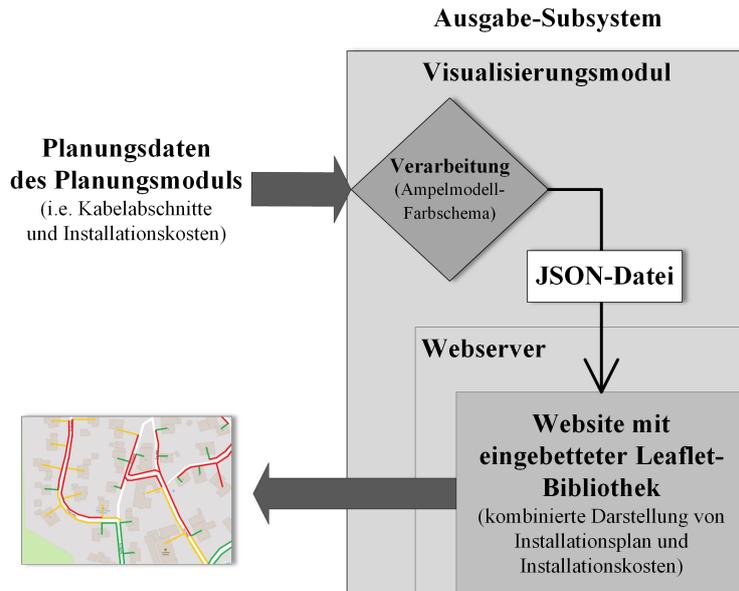


Abb. 2: Architektur des Visualisierungsmoduls.

Die Kartendarstellung der Installationspläne kann mittels externer Softwaremodule und APIs von Kartendienstleistern (z.B. „Bing Maps“, „Google Maps“, „Leaflet“, „Mapbox“, „OpenLayers“) realisiert werden. Alle vorgenannten Softwaremodule und APIs erlauben die Verwendung von Dateien im JSON-Format als Datenquelle für die einzuzeichnenden Kabelabschnitte. In dieser Studie wurde „Leaflet“ (Agafonkin, 2016), eine Open-Source JavaScript Bibliothek, für die Kartendarstellung ausgewählt. Das Visualisierungsmodul stellt als Benutzerinterface eine Webseite bereit, in die die Leaflet-Bibliothek integriert wird. Während des Öffnens der Webseite im Webbrowser wird die zuvor generierte JSON-Datei mit der kombinierten Darstellung der Installationspläne und Installationskosten geladen und an die Leaflet-Bibliothek zur Darstellung auf der Landkarte übertragen.

4 Anwendungsbeispiel

Ein Anwendungsbeispiel, bei dem die Glasfaserplanung für ein Siedlungsgebiet durchgeführt wird, soll das modulare Softwaresystem und die Implementierung des Visualisierungsmoduls validieren. Das Siedlungsgebiet ist in Abbildung 3 dargestellt und umfasst Gebäude und Straßen in ländlicher Siedlungsstruktur. Die Hintergrundkarte und die georeferenzierten Daten zur Berechnung des Glasfasernetzes stammen aus OpenStreetMap. Die Kabelwege des zu errichtenden Glasfasernetzes in FTTB-Netztopologie werden durch das modulare Softwaresystem automatisch berechnet. Dazu wird jedes postalisch adressierbare Gebäude über eine Glasfaser,

Der folgende Abschnitt befasst sich mit der Kostenoptimierung von Glasfasernetzen auf Basis der hier vorgestellten, integrierten Darstellung. In Abbildung 5 kann, durch die integrierte Darstellung des Installationsplans und der Installationskosten, die Verbindung zu dem blau markierten Gebäude intuitiv als kostenintensive Verbindung identifiziert werden. Abbildung 6 zeigt eine optimierte Variante des Glasfasernetzes. In der optimierten Variante wird das Gebäude nicht mehr über die Straße links des Gebäudes, sondern über die Straße rechts des Gebäudes erschlossen. Im Ergebnis verlängert sich die Hausanschlussleitung zwar um 5 Meter, jedoch können 35 Meter separater Leitungsweg entlang der Straße vermieden werden. Die Gesamtkosten für das Glasfasernetzwerk sinken in diesem Fall um 3000 € (Annahme: Installations- und Materialkosten 100 €/m).

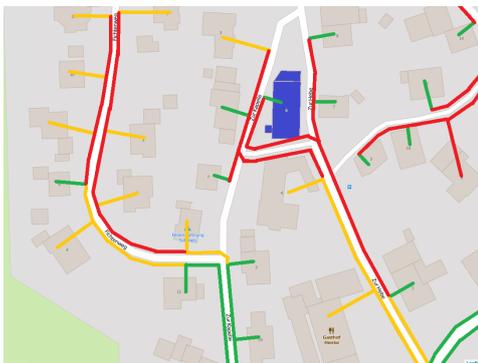


Abb. 5: Installationsplan mit integrierter Darstellung der Installationskosten.

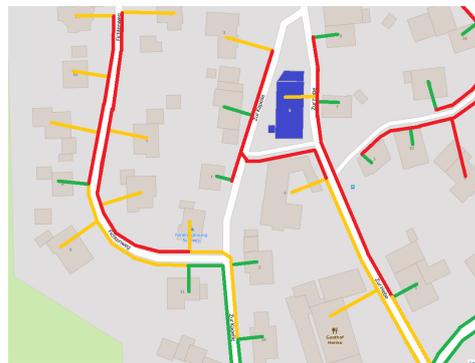


Abb. 6: Optimierter Installationsplan mit integrierter Darstellung der Installationskosten.

5 Zusammenfassung

In diesem Artikel wurde die Notwendigkeit des Glasfaser-Breitbandausbaus skizziert und der fehlende Zusammenhang zwischen der visuellen Darstellung der Installationspläne und der tabellarischen Darstellung der Installationskosten als Problem identifiziert. Ein Ansatz zur integrierten Darstellung von Installationskosten in Installationsplänen wurde vorgestellt und innerhalb eines modularen Softwaresystems implementiert. Die integrierte Darstellung von Installationskosten in Installationsplänen wurde mittels Ampelmodell durchgeführt. In einem Anwendungsbeispiel wurde der beschriebene Visualisierungsansatz validiert. Das Anwendungsbeispiel belegt, dass die integrierte Darstellung von Installationsplänen und Installationskosten einen klaren, intuitiven Überblick verschafft und Optimierungspotentiale erkennen lässt. Im Anwendungsbeispiel wurde die Verbindung zu einem Gebäude innerhalb eines Siedlungsgebietes durch die Farbgebung als kostenintensive Verbindung identifiziert. Eine Veränderung der Hausanschlussleitung hat die Verkabelungs- und Trassenlänge von 35 Meter auf 5 Meter reduziert.

Literatur

- Agafonkin, V. (2016). *Leaflet – an open-source JavaScript library for mobile-friendly interactive maps*. <http://leafletjs.com/> (besucht am 11.02.2016).
- Albert, J. (2009). *Innovations in Food Labelling*. Sawston, UK: Woodhead Publishing.
- Cisco Systems (2015). *Cisco Visual Networking Index: Forecast und Methodology 2014–2019*. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf (besucht am 11.02.2016).
- DE-CIX (2016). *DE-CIX Quick Facts*. <https://www.de-cix.net/about/quick-facts/> (besucht am 11.02.2016).
- Comsof NV (2014). *FTTx planning made smart, fast and easy*. http://www.comsof.com/sites/default/files/2014_fiberplanit_by_comsof_web.pdf (besucht am 11.02.2016).
- ECMA International (2015). *ECMAScript 2015, Language Specification, 6. Edition*. <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf> (besucht am 11.02.2016).
- Kaminow, I., Li, T. und Willner, A. E. (2010). *Optical fiber telecommunications VB: systems and networks*. Elsevier.
- Pritchard, C. (2013). *The project management communications toolkit*. Artech House.
- IT-Simplicity (2014). *FTTH / FTTX Software & Solutions for: Network designers & engineers*. <http://ftthsoftware.com/wp-content/uploads/2014/FTTH-FTTX-network-design-engineer-solutions.pdf> (besucht am 11.02.2016).
- U.S. Department of Transportation (2014). *Equipment Costs for Roadside Telecommunications (RS-TC)*. [http://www.itscosts.its.dot.gov/ITS/benecost.nsf/SubsystemCostsAdjusted?ReadForm&Subsystem=Roadside+Telecommunications+\(RS-TC\)](http://www.itscosts.its.dot.gov/ITS/benecost.nsf/SubsystemCostsAdjusted?ReadForm&Subsystem=Roadside+Telecommunications+(RS-TC)) (besucht am 11.02.2016).
- Wiggenbrock, J., Breide, S. und Smarsly, K. (2015a). In: *Proceedings of the 9th ITG Conference "Broadband Access in Germany"*. (Berlin, Germany, 21. Apr. 2015).
- Wiggenbrock, J., Breide, S. und Smarsly, K. (2015b). „Planung von Glasfaser-Zugangsnetzen in Siedlungsgebieten“. In: (Leipzig, Germany, 7. Mai 2015).
- Wiggenbrock, J. und Smarsly, K. (2015). „A combinatorial heuristic algorithm supporting computer-aided planning of optical fiber networks“. In: *Proceedings of the 22th ITG Conference "Communication Cable Networks"*. (Cologne, Germany, 9. Dez. 2015).
- Wiggenbrock, J. und Smarsly, K. (2016). „Integrated visualization of installation plans and installation costs of fiber optic networks“. In: *Proceedings of the 10th ITG Conference "Broadband Access in Germany 2016"*. (Berlin, Germany, 18. Apr. 2016).
- Woodward, B. und Husson, E. (2002). *Structured Cable Systems*. Hoboken, New Jersey, USA: John Wiley & Sons.