



Institut für Software & Systems Engineering
Universitätsstraße 6a D-86159 Augsburg



Masterarbeit im Studiengang
„Informatik“

Individualisierbarer Konstruktionsplaner mit automatischer Bauplandeduktion

Sebastian Rossi



Institut für Software & Systems Engineering
Universitätsstraße 6a D-86135 Augsburg



Individualisierbarer Konstruktionsplaner mit automatischer Bauplandeduktion

Sebastian Rossi

Erstgutachter: Prof. Dr. Wolfgang Reif
Zweitgutachter: Prof. Dr. Bernhard Bauer
Matrikelnummer: 1475010
Abgabe der Arbeit: 8. Januar 2023
Betreuer: Constantin Wanninger

Erklärung

Hiermit versichere ich, dass ich diese Masterarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Augsburg, den 8. Januar 2023

Sebastian Rossi

Zusammenfassung

The lack of housing is a growing problem in modern states. High cost of skilled labor in building construction intensifies this problem. Use of full automation in this domain is quite uncommon to this day. This paper presents a solution by using a collection of different robots tasked with individual jobs. With this approach the introduced system can tackle a multitude of problems, which arise from the difficult conditions on construction site. Principles of the Semantic Plug&Play concept are used to ensure a simple application and adaptability. As proof of concept different scenarios with increasing levels of difficulty are used.

Inhaltsverzeichnis

1 Motivation	1
2 Problemstellung	3
2.1 Bausteineindefinition	3
2.2 Wall-Detailing und Tiling	3
2.3 Definition Bauplan	3
3 Fallstudien	5
3.1 Planung und Bauplandeduktion eines LEGO Gebäudes mit Einsteinmauerwerk	5
3.1.1 Problemstellung	6
3.2 Planung und Bauplandeduktion eines Lego Gebäudes mit Verbandsmauerwerk	8
3.3 Szenario mit veränderbaren Bausteintypen	8
3.4 Sternchenaufgabe: Szenario mit "runden" Wänden und arbiträren (nicht rechteckige) Bausteinformen / schräge schnitte	8
4 Related Work	9
4.1 3D Druck und Additive Fertigung von Gebäuden	9
4.2 Legeroboter	9
4.2.1 Digital Plan of Brickwork Layout for Robotic Bricklaying Technology	9
4.3 Materialien	10
4.4 Bausteine	10
4.5 Wall detailing und das (3D) Bin Packing Problem	10
4.6 Ludwigs Dissertation	11
5 Grundlagen	13
5.1 Industry Foundation Classes	13
5.1.1 IFC 4.3.1.0 Aufbau	13
5.1.2 IfcPropertySets und IfcQuantitySets	14
5.1.3 Positionierung von IFCProducts	14
5.1.4 IfcOpeningElement	15
5.2 IFC for Blender	15
5.2.1 Blender	15
5.2.2 blenderbim	15
5.2.3 IfcOpenShell	15
5.3 Building Information Modeling	16
5.4 brick schema	17
5.5 opensourcebim	17

Inhaltsverzeichnis

5.6	LEGO	18
5.7	Mauerwerksbau	18
5.7.1	Mauerwerksverband	19
5.8	BRep	22
5.9	Definitionen	22
5.9.1	Bausteintyp	22
5.9.2	Baustein	22
5.9.3	Wand	22
6	Konzept	23
7	Realisierung	25
7.1	Modellierung	25
7.2	Wall Detailing	25
7.2.1	Konvertieren des IFC zu BREP	25
7.2.2	Überprüfen der modellierten Wände	25
7.2.3	Lösen der Beziehungen	27
7.2.4	Anwenden der Lösung	27
7.2.5	Export	27
8	Proof of Concept	29
9	Fazit und Ausblick	31
Literatur		33

1 Motivation

Der stetig steigende Fachkräftemangel trifft auch das Baugewerbe. Laut einer Umfrage des Hauptverbandes der Deutschen Bauindustrie e.V. stufen über 75 Prozent aller befragten Unternehmen sowohl den vorherrschenden Fachkräftemangel, als auch die steigenden Energie- und Rohstoffpreise als Risiko für das eigene wirtschaftliche Wachstum ein [10]. Abbildung 1.1 illustriert die Entwicklung dieser Sorge über einen Zeitraum von etwas mehr als zwanzig Jahren. Damit ist es wenig überraschend, dass eine Bewegung weg von menschlichen Arbeitskräften hin zur Automatisierung existiert. Neben dem Fachkräftemangel stellt aber auch die geringe Effizienz von Bauvorhaben ein Problem dar, welche sich über den gesamten Planungs- und Bauprozess erstreckt. Diese Ineffizienz entsteht aufgrund der Vielzahl der an Bauprojekten beteiligten Experten und Unternehmen und ist, als *Fragmentierungsproblem der Bauindustrie* bezeichnet, ein bekanntes Problem [22]. Deshalb etablieren sich derzeit Standards, um Bauprojekte digital zu begleiten. Mit diesen

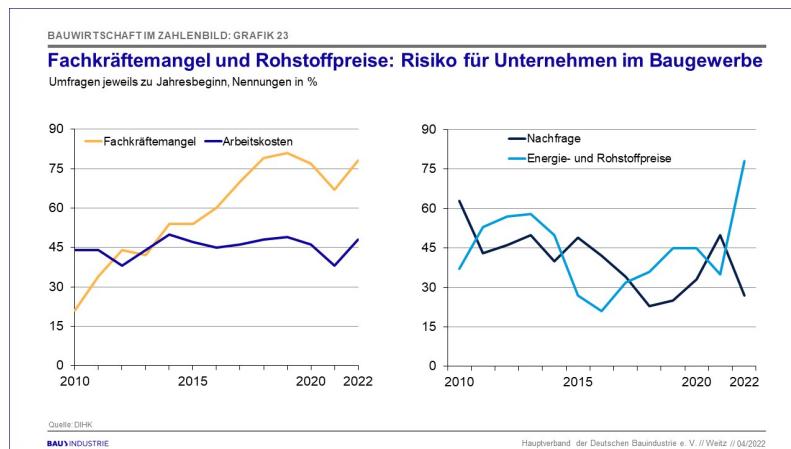


Abbildung 1.1: Während wirtschaftliches Risiko durch eine potentiell sinkende Nachfrage nach Bauanträgen und eventuell steigender Arbeitskosten unverändert blieben oder sogar als weniger relevant bewertet wurden, ist ein deutlicher Anstieg aufgrund des vorherrschenden Fachkräftemangels und der Energie- und Rohstoffpreise zu erkennen.

soll gleichzeitig die Effizienz gesteigert, die Kommunikation zwischen den einzelnen Expertenteams vereinfacht, der Arbeitsplatz "Baustelle sicherer gestaltet und ein resourcensparender Bau ermöglicht werden [6] [26]. Gleichzeitig steigen mit der Zunahme an digitalen Informationen zu Bauprojekten, auch die Möglichkeiten diese besser zu analysieren, zu optimieren und an neue Technologien zu knüpfen. Erst dadurch wurde das seit einigen Jahren erforschte Gebiet der Additiven Fertigung von Gebäuden, etwa mit Beton druckenden Roboterarmen oder mobilen Robotern, realisierbar [3]. [TODO

1 Motivation

nochmal reinschauen ob das einigermaßen als Quelle passt]. Obwohl es mittlerweile viele Projekte zur Additiven Fertigung von Gebäuden gibt, haben diese oft den Nachteil der Nicht-Parallelisierbarkeit der druckenden Roboter, die durch die Höhe der temporären Stützstrukturen (wie Kräne, Gerüste, Aufhängungen) eingeschränkte Bauhöhe und die vergleichsweise lange Bauzeit [TODO Quelle oder lange bauzeit weg]. [Frage wegen Kommentar: Star Wars roboterschwarm/Marsbesiedelung; soll ich hier sagen dass ein derzeit in der SciFi Szene aufgekommener Trend von organisierten Roboterschwärmen, die was bauen gar nicht so blöd ist und wir das ähnlich angehen?] Diesen Einschränkungen soll nun mithilfe eines Schwarmes bodengebundener automoner Roboter, welche gleichzeitig an dem Bauprojekt arbeiten, entgegengewirkt werden. Dabei sollen sich die Roboter auf den Mauern des Gebäudes selbst bewegen können, während sie dieses errichten. In dieser Arbeit liegt der Schwerpunkt allerdings nicht auf dem Entwickeln der Roboter selbst, sondern das Erstellen eines Bauplans für den genannten Roboterschwarm basierend auf einem 3D Modells des Gebäudes. Das Erstellen des Modells innerhalb eines nutzerfreundlichen 3D Editors stellt nicht nur die geometrischen und physikalischen Eigenschaften des Gebäudes digital bereit, sondern verschlankt auch die Kommunikation zwischen Endnutzer und Architekt oder ersetzt letzteren komplett. Gleichzeitig bildet diese Arbeit damit auch den Trend hin zur sogenannten Massenpersonalisierung ab, welcher als Nachfolgetrend zur Massenproduktion und als "heiliger Gral" der Fertigung angesehen wird . Dieser Trend ist auch für die Bauindustrie interessant, denn auch hier schafft die Möglichkeit sämtliche Kundenwünsche an ein Produkt (oder Gebäude) umzusetzen, ohne dafür spezielles Werkzeug herstellen zu müssen, neue Gewinnmöglichkeiten. Mit der Option der Modellierung des Gebäudes durch den Kunden selbst, ist die Kommunikation mit den Architekten über dessen Wünsche effizienter, da beide Parteien zusammen an dem Modell arbeiten können. Im Anschluss an den Designprozess des 3D Models des Gebäudes, soll dieses in einen Bauplan übersetzt werden, welcher alle notwendigen Informationen für den oben genannten Roboterschwarm enthält, sodass dieser das Gebäude selbstständig errichten kann [TODO ugs]. Dieses Konzept wird anhand nachfolgender Fallstudie(n) getestet.

2 Problemstellung

Ziel der Arbeit ist es einen Workflow zu schaffen, welcher es einem Nutzer ermöglicht ein Gebäude in einem 3D Designer zu planen, das im Anschluss in einen durch einen heterogenen Roboterschwarm ausführbaren Bauplan übersetzt wird. Dies erfolgt in mehreren Schritten, welche sich jeweils mit unterschiedlichen Fragestellungen befassen. Anhand der Teilschritte lässt sich diese umfangreiche Problemstellung logisch einteilen und die jeweiligen Kernprobleme und Fragestellungen der einzelnen Schritte werden klar.

Roboter auf Mauer -> Test nicht mit Schaumstoffbricks möglich da gewicht Lücken (Fenster, Türen)

2.1 Bausteindeinition

Wie können wir zu einem bestimmten Wandtyp ein Set an Bausteintypen definieren, mit welchen Wände diesen Typs gebaut werden müssen. Dabei sollen die Bausteine nicht auf Quader beschränkt, sondern beliebige Körpern sein können. Ebenfalls relevant sind eventuelle die Bausteinverbindungen, die ebenfalls Einschränkungen haben. Ein Beispiel dafür ist etwa Mörtel bei Ziegelwänden.

2.2 Wall-Detailing und Tiling

Wie kann man algorithmisch mit einem Set an Bausteintypen eine Wand, welche als Mesh vorliegt, vollständig erbauen und daraus einen Bauplan herleiten (Stichwort Abhängigkeitsgraph).

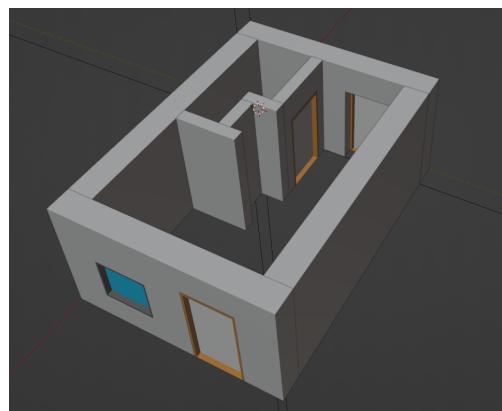
2.3 Definition Bauplan

Was muss ein Bauplan konkret beinhalten? LALAB

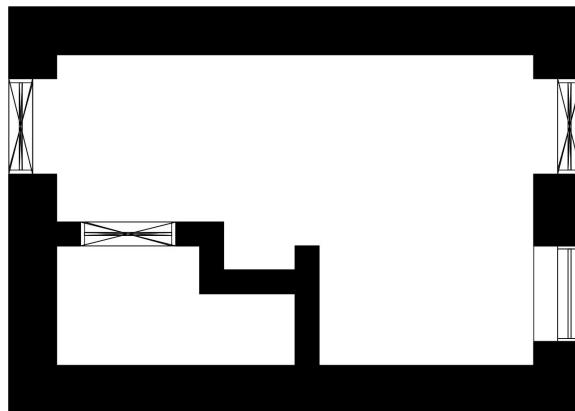
3 Fallstudien

3.1 Planung und Bauplandeduktion eines LEGO Gebäude mit Einsteinmauerwerk

Ziel dieses Szenarios ist es, aus dem in Abbildung 3.1a dargestellten 3D Modell ein Bauplan zu generieren. Zu sehen ist der Plan eines einfachen Hauses mit einem Stockwerk.



(a) 3D Modell innerhalb von Blender.



(b) Gebäudeplan des 3D Modells.

Abbildung 3.1: Modell eines Studentenzimmers (verschiedene Darstellungsformen).

Dieses besitzt eine Eingangstür, eine Terrassentür neben einem Fenster und eine Tür, die das Badezimmer vom Hauptraum trennt. Türen und Fenster stellen eine Herausforderung für den Planungsalgorithmus dar, da der Verlauf einer ansonsten durchgängigen Wand dadurch unterbrochen wird und Lücken aufweist. Details wie Duschen, Betten, Toiletten und ähnliche Komponenten, sind für dieses Szenario irrelevant, da diese keinen Effekt auf die Struktur der Wände haben und wurden aus diesem Grund bewusst weggelassen. Das Modell wurde mithilfe der in Kapitel 5 näher behandelten Technologien erstellt und entspricht in seiner Struktur einem verbreiteten Industriestandard. Dafür wurden zwei Wandtypen definiert, die jeweils unterschiedliche Wanddicken vorgeben. So gibt es breite Außen- und dünne Innenwände. Diese entsprechen in ihren Maßen dem Raster, welches das *LEGO System* (siehe Kapitel 5) vorgibt. Für breite Wände gilt, dass diese immer zwei Noppen breit, mindestens eine Noppe lang und einen Stein hoch sein muss. Für dünne Wände hingegen gilt eine feste Breite von einer Noppe, ebenfalls eine Mindestlänge von einer Noppe und eine Mindesthöhe von einem Stein. Beide Wandtypen können nur Höhen beziehungsweise Längen aufweisen, die jeweils einem Vielfachen der Höhe oder Länge des kleinsten Legosteins aufweisen, der zum Bau der Wände verwendet werden soll (hier der 1x1 Stein). Daraus resultiert ein Raster, welches ebenfalls für die

Ausmaße und Positionen der Fenster und Türen einzuhalten ist. Dieses Raster gilt es, abhängig des ausgewählten Wandtyps, in den Editor zu integrieren, um das Modellieren solcher Gebäude nutzerfreundlich zu gestalten und nicht durch ständiges Messen und Eintragen genauer Positionen oder Maße zu unterbrechen. Nicht nur die Abmessungen

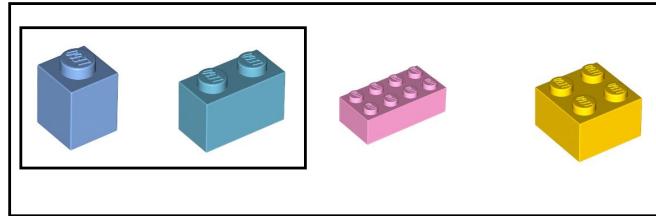


Abbildung 3.2: LEGO Steintypen für die Innen- und Außenwände. (TODO Bild ist sehr hässlich)

der Wände müssen in ein Raster fallen, auch deren Rotation wird in diesem Szenario auf 90° Schritte limitiert. Das stellt in diesem Fall eine vertretbare Enschränkung dar, da es ohnehin dem intuitiven Umgang mit LEGO Steinen und gleichzeitig dem Baustil der meisten einfachen Gebäuden entspricht. Folglich muss ein Format für die Bausteintypen entwickelt werden, aus welchen all diese Informationen abgeleitet werden können. Dieses Format muss sowohl von dem Editor selbst verwendet, als auch zur Berechnung innerhalb des Planungsalgorithmus herangezogen werden. Außerdem werden weitere Regeln benötigt, um den resultierenden Plan näher an das Vorgehen eines realen Baus zu bringen. So werden beim Errichten von Häusern zuerst die Ecken (der Schnittpunkt zweier Wände) um eine Stufe erhöht, um im Anschluss die geraden Wandabschnitte aufzufüllen. Damit wird vermieden, dass in den ohnehin schon komplexeren Eckbereichen auch noch zugeschnittene Ziegel notwenig werden. Stattdessen schneidet man diese erst zurecht, wenn sich dann eher mittig im Wandabschnitt Lücken ergeben, die kleiner sind als die vorhandenen Ziegel. Zwar wird dies im Fall von Legosteinen nie auftreten, aber es ist übersichtlicher das Problem in dem vorliegenden eingeschränkten Szenario zu beschreiben. Zusätzlich müssen Regeln in den Planungsalgorithmus eingeführt werden, die diverse Mauerwerksverbände (vorgestellt in Kapitel 5) erzwingen zu können, um damit die Stabilität und gleichmäßige Kraftverteilung innerhalb einer Wand zu gewährleisten und sich ebenfalls möglichst nah an der Realität des Mauerbaus zu bewegen.

Einschränkungen:

- * Alle Wände verwenden ein Ziegelset, in welchem alle Ziegel die gleiche Höhe haben
- * Alle Wände verwenden den selben Mauerwerksverband
- * Alle anderen sich berührenden Wände stehen in einem 90 Grad Winkel zueinander

3.1.1 Problemstellung

Bevor für das Modell ein passender Bauplan entwickelt werden kann, muss das sogenannte *Wall Detailing* (vgl. Kapitel 4.5) stattfinden. Darunter versteht man das Berechnen einer Bausteinkombination, die eine Wand möglichst gut abbildet. Dabei bedeutet "möglichst gut", dass die Proportionen der Bausteinkombination möglichst den der Ursprungswand entsprechen und keine nicht gewollten Lücken entstehen. In diesem Szenario wird durch

Auflegen des Rasters des LEGO Systems schon während des Modelliervorgangs sicher gestellt, dass das resultierende Modell mit Legosteinen baubar ist. Darum ist das Ziel des *Wall Detailing* Schrittes in diesem Szenario, jede Wand lückenlos mit Bausteinen zu füllen. Zusätzlich existieren folgende Einschränkungen und Eigenschaften, die für das Ergebnis dieses Szenarios gelten sollen.

Überbindemaß Obwohl in dem vorliegenden Modell ausschließlich Einsteinmauerwerke vorgesehen sind, gilt es die in Kapitel 5.7.1 erläuterte Regel zum Überbindemaß von Bausteinen zu beachten. Da es sich hierbei aber um LEGO Steine handelt, die wesentlich kleiner sind als die genormten Formate für Ziegelsteine, wird der für das Überbindemaß vorgesehene Mindestwert von 45mm ignoriert. Außerdem ist ein Versatz von unter 50% der Dicke eines Legosteines in den meisten Fällen nicht umsetzbar, da diese nicht frei übereinander gesteckt werden können. Darum wird für dieses Szenario ein Überbindemaß von exakt 50% der Steindicke verwendet, was den Einschränkungen des Überbindemaßes entspricht, welches einen Mindestversatz von 40% der Steindicke voraussetzt. Für die in Abbildung 3.2 dargestellten Lego Steine kann mit einem solchen Überbindemaß gleichzeitig auch das vorgesehene Raster des Lego Systems eingehalten werden, da diese alle eine gerade Anzahl an Noppen besitzen. TODO Bild von dumm gestapelter LEGO Wand zu LEGO Wand mit Überbindemaß.

Anstoßende Wandstücke Besonders herausfordernd ist die Einhaltung des Überbindemaßes an anstoßenden Wandstücken. Solche Wandstücke entstehen zum Beispiel an Ecken, also 90 Grad zueinanderstehende Wände oder aber an den Übergängen von Außenwänden zu Innenwänden. Wie in Abbildung 3.1a zu erkennen treten in diesem Szenario beide Fälle auf. Zunächst müssen daher alle anstoßenden Wandstücke aus dem Modell gefiltert werden. In Kapitel 5.7.1 werden gängige Praktiken zur Lösung von anstoßenden Wandstücken vorgestellt. Mit diesen Informationen muss der Planungsalgorithmus zunächst eine Lösung für solche Wandstücke finden und anschließend die restlichen Flächen mit dem Vorgehen von oben auffüllen.

Öffnungen Das Fenster und die drei Türen sind hier die einzigen Besonderheiten in dem Modell. Für den Mauerbau bedeutet dies an den entsprechenden Stellen Öffnungen zu lassen. Die Informationen über diese Öffnungen liegen innerhalb des Formates vor, in welchem das Modell erstellt wurde (siehe ??). So lassen sich Größe und Position leicht herausfinden und das Ergebnis der beiden obigen Schritte dementsprechend abändern.

Nun liegt eine Beschreibung des Modells mit dem zuvor definierten Set an Legosteinen vor. Als letzten Schritt gilt es daraus eine Abfolge an Bauinstruktionen zu finden, die unter Beachtung von diversen Einschränkungen zum gewünschten Ergebnis führen. Mit dieser Problemstellung hat sich Ludwig (TODO) in seiner Dissertation auseinandergesetzt (siehe Kapitel 4.6). Einschränkungen, die beim Bau von Legokonstruktionen gelten sind: Deadlock, Überhang etc. (TODO) Das Vorgehen in seiner Arbeit soll für dieses Szenario evaluiert und gegebenenfalls angepasst werden.

3.2 Planung und Bauplandeduktion eines Lego Gebäude mit Verbandsmauerwerk

Kein Einsteinmauerwerk -> dicke Wände, die mit Läufern und Bindern erstellt werden müssen. Eckfälle sehr komplex (siehe Basics)

3.3 Szenario mit veränderbaren Bausteintypen

Wie können wir die Bausteine veränderbar machen, sprich die Bearbeitungsmöglichkeiten während des Baus (schneiden eines Ziegels) miteinbeziehen in unser Bausteinformat. Erstmal das schneiden nur parallel zu den ebenen eines rechteckigen ziegelsteins betrachten, denn bei β chrägen β schnitten entstehen neue komplexe Formen.. Wie generiert man daraus dann Baupläne -> riesiges aber cooles Optimierungsproblem! Vlt geht das richtung constraint programming? Sprich den Bauplan als Lösung für ein beschränktes Problem ansehen. Total viele Fragen, keine Ahnung wo zu beginnen aber klingt cool

3.4 Sternchenaufgabe: Szenario mit "runden" Wänden und arbiträren (nicht rechteckige) Bausteinformen / schräge schnitte

Was machen wir wenn die Wand nicht perfekt mit den vorgegebenen Bausteintypen gebaut werden kann (Beispiel ein runder Turm) Bausteinverbindungen (wie Mörtel) betrachten und als *Verbindungselement* in das Bausteinformat mit aufnehmen? Wie kann man diesen so einschränken, dass nur physisch machbares ausgerechnet wird. Was wenn die Bausteine arbiträre Formen haben und nur in sehr komplexen Mustern eine "dichte" Wand ergeben -> tiling Probleme.

4 Related Work

4.1 3D Druck und Additive Fertigung von Gebäuden

4.2 Legeroboter

4.2.1 Digital Plan of Brickwork Layout for Robotic Bricklaying Technology

In diesem Paper stellen Usmanov et al. ein generelles Vorgehen für das Erstellen eines Ziegel-Legeplans für ein als digitales Modell vorliegendes Gebäude vor [1]. Dieses Vorgehen gliedern sie in sechs Schritte:

1. Das vorliegende IFC Modell (siehe 5.1) nach Wandelementen durchsuchen und diese in das sogennante BREP-Format (siehe 5.8) konvertieren.
2. Das Aufteilen des gesamten Modells in Schichten, die der Modulhöhe des verwendeten Ziegelsteinformats entspricht.
3. Verbindungen von getrennt modellierten Wandelementen heraussuchen. Dies ist zum Beispiel an Eckstücken der Fall, da dafür oft zwei einzelne Wandelemente modelliert werden, welche in einem Winkel zueinander stehen und sich berühren. Für die nachfolgenden Schritte sind diese Verbindungen relevante Informationen.
4. Mit den Informationen der vorhergegangenen Schritte können nun für jede Schicht kritische Bereiche identifiziert werden, an welchen später ein komplexer Legevorgang von Ziegeln von Nöten ist.
5. Nun werden zunächst die kritischen Bereiche anhand einer vorher definierten Legeanleitung mit teilweise angepassten Ziegelsteinen bestückt und im Anschluss die restlichen Bereiche aller Wände mit dem ausgewählten Standardziegel aufgefüllt. In diesem Schritt werden zusätzlich Fenster- und Türstürze über deren Öffnungen in den Wänden gelegt.
6. Dieser Schritt fasst das Anzeigen als 3D Modell und Konvertieren des Resultats in eine nicht konkreter definierte Listenform zusammen. Das Ergebnis für das von den Autoren ausgewählte Beispielgebäude ist in Abbildung 4.1 zu sehen.

Besonders detailliert ist ihr Ansatz für das Finden von besagten kritischen Teilbereichen einer Wand mithilfe einiger mathematischer Gleichungen, die alle Wände zueinander in Beziehung stellen. Diese Teilbereiche sind Wanddecken, T-Kreuzungen (wie etwa der Übergang einer Innenwand an eine Außenwand) und Öffnungen innerhalb einer Wand und sind ebenfalls in Kapitel 3.1.1 dieser Arbeit als Problemstellung aufgeführt. Anhand der von ihnen zusammengetragenen Informationen konnten die Autoren erfolgreich die

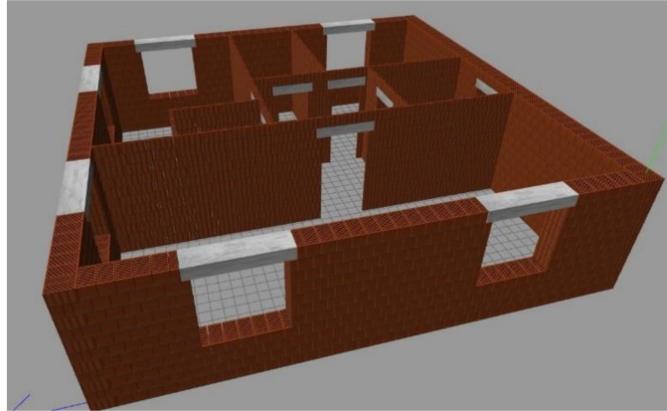


Abbildung 4.1: Ergebnis des Verfahrens zur Erstellung eines Ziegel-Legeplans nach Usmanov et al. [1].

bestmögliche Platzierung von vier Roboterarmen errechnen, die den schnellstmöglichen Bau des Gebäudes ermöglichen. Dennoch werden zum Schluss noch einige Einschränkungen ihres Verfahrens angesprochen. Vor allem die Beschränkung auf 90 Grad Ecken und das Gebunden sein an einen einzigen Standardziegel werden als besonders restriktiv wahrgenommen.

Insbesondere die Verwendung von verschiedenen Ziegelformaten und das algorithmische Finden von Lösungen an den kritischen Teilbereichen einer Wand, anstatt diese als Legeplan vorliegen zu haben, sind Ziele dieser Arbeit. Dennoch sind einige Punkte ihres Vorgehens auch zwangsläufig notwendige Schritte für diese Arbeit und werden in einer ähnlichen Form vorzufinden sein.

4.3 Materialien

4.4 Bausteine

4.5 Wall detailing und das (3D) Bin Packing Problem

TODO hinführen über Bin Packing hin zu Spezialfall "Wall detailing mit arbiträren Bausteinen und Eigenschaften (wie versetzen der ziegel)

TODO über bin packing schreiben, erklären paper suchen, lösungsansätze zu np hartem problem

Xu Chengran et al. haben in ihrem Paper "Optimal brick layout of masonry walls based on intelligent evolutionary algorithm and building information modeling" verschiedene Optimierungsansätze aus dem Bereich des 2D Packaging Problems getestet [2]. Konkret wurden drei Algorithmen verwendet: Differential Evolution, Particle Swarm Optimization und Neighbourhood Field Optimization. Außerdem wird ein dreiphasiges Vorgehen vorgeschlagen: Data collection, Brick layout und Data Output. Dieses Vorgehen eignet sich auch für das Finden von Bausteinkonfigurationen in dieser Arbeit, da zuerst alle

relevanten geometrischen Daten (in diesem Fall Wände, Fenster, Türen usw.) aus dem 3D Modell gesammelt werden müssen, bevor das Detailing stattfinden kann. Nach dem Optimieren der Bausteinkonfiguration muss das Ergebnis ebenfalls in ein Format gebracht werden, das für die folgenden Schritte verwendet und eventuell auch dem Nutzer angezeigt werden kann.

Soft items: <https://arxiv.org/abs/2206.15116>

Irregular Shaped items: <https://link.springer.com/content/pdf/10.1631/FITEE.1400421.pdf>

"Parametric Blockwall-Assembly Algorithms for the Automated Generation of Virtual Wall Mockups Using BIM"

4.6 Ludwigs Dissertation

5 Grundlagen

Wie bereits oben erwähnt, wird für diese Arbeit ein geeignetes Speicherformat für 3D Modelle von Gebäuden benötigt. Um den Bau von Gebäuden zu automatisieren, ist es notwendig die Domäne *Gebäude* vollständig digital abbilden zu können. Hilfreich ist dabei, wichtige Daten über bestimmte Bestandteile des Gebäudes direkt in das Modell zu integrieren auf Basis derer etwa Kostenberechnungen durchgeführt oder Materialmengen herausgefunden werden können. Da diese oft von verschiedenen Experten vieler Fachbereiche (etwa aus den Bereichen der Architektur, des Bauwesens oder der Statik) stammen, muss das Format sehr flexibel und im besten Fall auch zeitgleich bearbeitbar sein. Dafür werden seit 2000 die *Industry Foundation Classes* (IFC) von buildingsmart entwickelt, deren Anwendung im internationalen Bauwesen mittlerweile weit verbreitet ist [15].

5.1 Industry Foundation Classes

In der Spezifikation des Standards selbst, wird dieser wie folgt beschrieben: "Die Industry Foundation Classes (IFC) sind ein offener internationaler Standard für Daten des Building Information Model (BIM), welche zwischen Softwareanwendungen ausgetauscht, gemeinsam genutzt und von den verschiedenen Akteuren der Bauindustrie und des Gebäudemanagements verwendet werden. Der Standard enthält Definitionen für Daten, die für die Lebenszyklen von Gebäude- und Infrastrukturarbeiten erforderlich sind. Die bis jetzt in die IFC aufgenommenen Infrastrukturtypen umfassen Brücken, Straßen, Eisenbahnen, Wasserstraßen und Hafenanlagen"(aus dem Englischen) [17]. Eine frühere Version des IFC Standards ist unter der Bezeichnung ISO 16739[18] registriert. Da die IFC aber nach wie vor kontinuierlich weiterentwickelt werden, wird in dieser Arbeit die derzeit neueste Version verwendet. Diese ist die IFC Spezifikation 4.3.1.0 [16]. Das verbreitetste Austauschformat für IFC ist das Step Physical File Format, welches im ISO 10303 Teil 21 registriert ist [13]. Zudem gibt es speicherreduziertere Formate wie ifcZip oder für Menschen lesbarere Formate wie ifcXML [14] [11].

5.1.1 IFC 4.3.1.0 Aufbau

Im Grunde definieren die *Industry Foundation Classes* eine Vielzahl an Klassen, die in einer komplexen Hierarchie angeordnet den Grundstock des Datenmodells bilden. Diese sind anfangs abstrakte Konzepte, die sich mit zunehmender Tiefe in der Hierarchie konkretisieren.

Da sich diese Arbeit zum größten Teil mit aus Wänden bestehenden Gebäuden befasst, wird nachfolgend die Klasse *IfcWall* wiederholt als Beispiel herangezogen. Der für diese

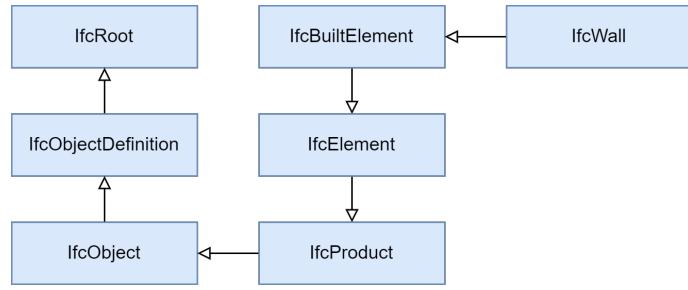


Abbildung 5.1: Klassenhierarchie am Beispiel der Klasse *IfcWall*

Klasse relevante Ausschnitt aus der Klassenhierarchie ist in Abbildung 5.1 dargestellt. Objekte werden von dem Standard in Relation zueinander gestellt, um komplexere Zusammenhänge darzustellen. In Abbildung 5.2 erkennt man den Zusammenhang zwischen einem Objekt des Types *IfcWall*, des Stockerwerks, welches diese Wand referenziert und wiederum selbst Teil eines *IfcBuildings* ist, bis hin zur obersten Komponente eines Ifc Projektes, dem gleichnamigen *IfcProject*.

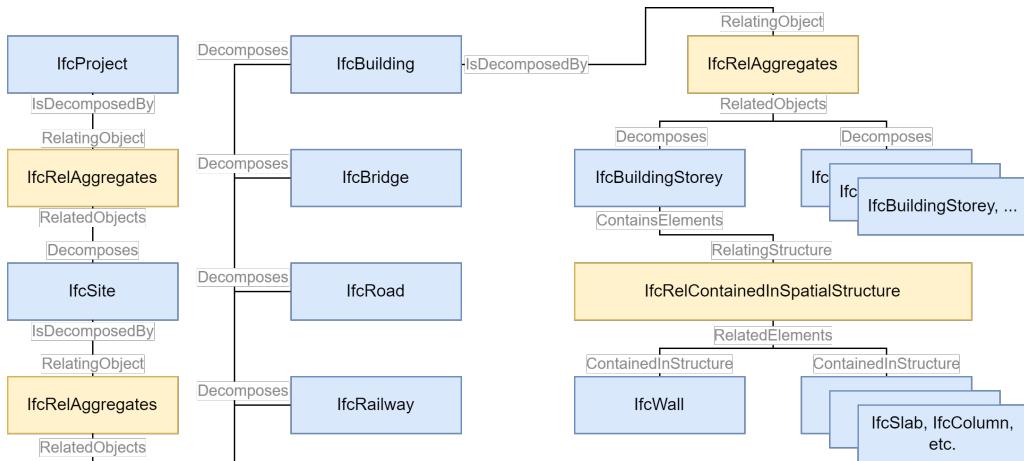


Abbildung 5.2: Relation der *IfcWall* und einem *IfcProjekt*

5.1.2 IfcPropertySets und IfcQuantitySets

Mit dem Erweitern des Gebäudemodells um möglichst viele Informationen, schafft man einen detaillierten digitalen Zwilling, der neben der bloßen Darstellung des Gebäudes als 3D Modell zb. auch eine präzisere Kosten- und Zeitschätzung für den Bau ermöglicht [14]. Dies kann mithilfe der IfcPropertySets und der IfcQuantitySets umgesetzt werden blablabla.

5.1.3 Positionierung von IFCProducts

TODO: Positionierung ist anscheinend bisschen verschachtelt Blablabla so können aus einem IFC Projekt alle relevanten Klassen mit den für diese Arbeit erforderlichen Eigen-

schaften extrahiert werden.

5.1.4 IfcOpeningElement

Wie funktionieren Löcher in z.B. IfcWall? -> IfcOpeningElement. Am Beispiel eines Fensters erklären. In dessen Doku steht dass Windows auf ein IfcOpeningElement verweisen, welches ein Stück Wand wegmacht und z.b. durch das Fenster aufgefüllt wird. TODO: mit rausparsing und aus wandmeshes rauschneiden vor detailing.

ifcopeningelement um lücken in wände zu machen. "The opening element stands for opening, recess or chase, all reflecting voids. It represents a void within any element that has physical manifestation. Openings can be inserted into walls, slabs, beams, columns, or other elements. Äuszug aus Doku. und This also includes IfcOpeningElements, the mechanism used to extract openings for windows and doors, typically from walls.

5.2 IFC for Blender

5.2.1 Blender

Blender ist eines der beliebtesten Open Source Programme zur Modellierung von 3D Modellen und Animationen [7]. Aufgrund dessen existieren auch eine Vielzahl an freien Erweiterungen bzw. Plugins - unter anderem auch eine Integration von IFC Projekten.

5.2.2 blenderbim

Neben kommerziellen Produkten wie etwa revit von autodesk [23] zur Modellierung von IFC Modellen, gibt es auch für Blender ein freies Plugin, um IFC Modelle zu erstellen [8]. Dieses Plugin ermöglicht es neben dem bloßen Designen des Gebäudes in kurzer Zeit z.B. detaillierte Zeichnungen verschiedener Perspektiven herauszuarbeiten, die z.B. von Bauingenieuren verwendet werden können, um einzelne Stockwerke oder Verkabelungen zu planen. Blenderbim selbst kapselt unter anderem die Open Source Python Bibliothek *IfcOpenShell*, sodass diese in der Blender Laufzeitumgebung zur Verfügung steht [12].

5.2.3 IfcOpenShell

```
1 import ifcopenshell
2 from ifcopenshell import geom
3 from stl import mesh, Mode
4 import numpy as np
5
6 settings = ifcopenshell.geom.settings()
7 settings.set(settings.USE_WORLD_COORDS, True)
8
9 ifc_file = ifcopenshell.open("../models/sample_house.ifc")
10 products = ifc_file.by_type("IfcProduct")
11 meshes = []
12
13 for product in products:
```

```
14 if product.Representation and product.is_a("IfcWall"):
15     shape = ifcopenshell.geom.create_shape(settings, product)
16     vertices = np.array(shape.geometry.verts).reshape((-1, 3))
17     edges = np.array(shape.geometry.edges)
18     faces = np.array(shape.geometry.faces).reshape((-1, 3))
19
20     m = mesh.Mesh(np.zeros(faces.shape[0], dtype=mesh.Mesh.dtype))
21     for i, f in enumerate(faces):
22         for j in range(3):
23             m.vectors[i][j] = vertices[f[j], :]
24     meshes.append(m)
25
26 # Create the combined mesh
27 combined = mesh.Mesh(np.concatenate([m.data for m in meshes]))
28 combined.save('cube.stl', mode=Mode.ASCII)
```

Listing 5.1: Beispielprogrammcode um bestimmte Daten aus einem IFC File zu laden und daraus ein Mesh zu generieren

TODO Beispielcode zum extrahieren einer Wand und deren geometrischen Eigenschaften
conda create -n masterarbeit conda activate masterarbeit conda install -c conda-forge ifcopenshell conda install -c conda-forge ipykernel python -m ipykernel install --user --name=masterarbeit conda install -c conda-forge pythonocc-core=7.7.0 conda install -c conda-forge meshplot

conda create -n masterarbeit conda activate masterarbeit conda install -c conda-forge ifcopenshell conda install -c conda-forge pythonocc-core=7.7.0 conda install -c anaconda pyqt

the ABS models to solve floor layout problems constraint solver / programming

5.3 Building Information Modeling

Ein weiterer Punkt, der für die Verwendung von IFC spricht ist das sogenannte *Building Information Modeling* (BIM)[9]. Ein Definitionsvorschlag lautet wie folgt: „BIM ist definiert als der Einsatz von Informations- und Kommunikationstechnologien zur Verschlankung der Prozesse im Lebenszyklus von Gebäuden, um eine sicherere und produktivere Umgebung für die Bewohner zu schaffen, die Umwelt so wenig wie möglich zu belasten und die Effizienz der Betriebsabläufe für die Eigentümer während des gesamten Lebenszyklus des Gebäudes zu erhöhen“ (Übersetzt aus dem Englischen)[21]. Zum Lebenszyklus eines Gebäudes gehören etwa anfangs das Planen und Designen, später das Bauen, das Verwenden und Instandhalten und nach eventuellen Renovierungen das Abreißen. BIM kommt in all diesen Phasen zum Tragen und erleichtert diese Prozesse durch Anbieten einer einheitlichen Schnittstelle für alle am Infrastrukturbau und -management beteiligten Personen. Zusätzlich ermöglicht BIM eine exakte Dokumentation des Geschehens in sämtlichen Phasen des Bauwerks, was unter anderem zu einer genaueren Zeit- und Kostenplanung führt. Auch Verantwortlichkeiten sind Teil von BIM, was zu einer erhöhten Produktivität beiträgt. Um nun das Zusammenarbeiten der unterschiedlichen Fachbereiche zu erleichtern, gibt es sogenannte BIM-Server auf welchen mehrere Arbeitende synchron an einem Projekt arbeiten können, während sie jeweils die für ihren Aufgabenbereich passende Ansicht vor sich haben. BIM-Server unterstützen zusätzlich

eine Versionierung des Fortschritts an einem Projekt.

In einem Gespräch mit einem Ingenieur aus dem Bereich „Energysystemtechnik“ kam zur Sprache, dass viele Bereiche von BIM noch nicht ganz Einzug in Deutschland gefunden haben. Eben jene „Kollaboration über einen BIM-Server mit Änderungsmanagement etc. [sei] (noch) nicht üblich, da noch nicht alle Beteiligten dazu in der Lage sind. Vor allem Bauherren, Architekten und Baufirmen können es nicht“. Weiter sei „auch unklar, wer für falsche Angaben haftet und wer die Konsistenz aller Daten gewährleistet“. Auf der anderen Seite sei „das im BIM festegelegte Datenformat IFC das Maß der Dinge und auch bei uns so in Verwendung“. Auch das Einpflügen „ergänzende[r] Bauteilinformationen (z.B. zu Gewicht, Dämmwert, Recyclebarkeit, CO₂ Fußabdruck, etc.)“ finden Einsatz und sind Teil seines Alltags. Für ihn wichtig ist ebenfalls der Betrieb des Gebäudes. Hier unterstützt BIM, indem sämtliche Teile der Installationen in einem Gebäude, wie z.B Fensterdichtungen, Kabel, Rohre, Sicherungen oder eine Umwälzpumpe individuelle Teilenummern zugewiesen bekommen, hinter welchen alle Daten wie etwa Hersteller, Bestellnummern, Lebensdauer, Wartungshistorie oder Entsorgungsnachweise vermerkt sind. Dies wurde allerdings „angesichts der Realität der Handwerker und Gebäudenutzer für völlig unrealistisch und auch etwas over-engineered“ eingestuft. Trotzdem sei „BIM [...] das große Ding in der Bauwelt und der einzige echte Standard“.

5.4 brick schema

ChatGPT „While IFC is primarily focused on representing building information for interoperability between software applications, BrickSchema focuses on providing a standardized and semantically rich representation of building systems and their components. In practice, BrickSchema can be used in conjunction with IFC to enhance the semantic representation and analysis of building data, particularly when it comes to systems-level information.“

By using IFC as a foundation for representing the overall building information and combining it with the more detailed and specialized semantic representation offered by BrickSchema, it becomes possible to achieve a comprehensive and interoperable representation of building information that can support various use cases, including energy modeling, fault detection and diagnosis, and optimization of building performance.“

5.5 opensourcebim

Während es vorwiegend kommerzielle Produkte gibt, die Unternehmen das Arbeiten mit BIM ermöglichen, existiert auch hier eine Open Source Bewegung. Darin enthalten sind 70 Repositories, unter anderem ein BIM-Server inklusive diverser Clients für Endanwender und Werkzeuge, um einfacher mit den IFC Files zu agieren[25]. Ein kurzer Test hat gezeigt, dass die in Blender modellierte IFC Files tatsächlich über einen „Anzeige-Client“, der mit einer BIM-Server verbunden ist, angezeigt werden können. Obwohl die Verwendung des BIM-Servers für diese Arbeit nicht notwendig ist, besteht die Option diesen künftig mit in den Workflow zu integrieren, da damit auch das simultante Arbeiten an einem IFC File möglich ist, was in Blender nur teilweise und mit dem Einsatz von

Plugins ermöglicht wird. Dabei ist fraglich, ob diese Plugins dann ebenfalls die IFC Erweiterung unterstützen. Das stellt einen Praxisbezug zum aktuell verwendeten Stand dieser Technologien her, was in der oftmals konzeptionellen Natur der Forschung nicht immer der Fall ist. Wie auch Blender unterstützt BIM-Server das Einbinden von eigenen Plugins, sodass eine Erweiterung um neue Funktionalität möglich ist. Die Plugins werden in Java geschrieben. Der Server bietet aber auch eine REST Schnittstelle an, um Clients in anderen Sprachen anzubinden.

5.6 LEGO

Ein 1x1 LEGO Stein hat eine quadratische Grundfläche von $7.8mm \times 7.8mm$. Zwischen zwei nebeneinander platzierten Steinen ist ein Abstand von $0.2mm$. Daraus ergibt sich ein Rastermaß von $8mm \times 8mm$. In Abbildung 5.3 werden zur Veranschaulichung die Maße des populären 2x4 Steines aufgeschlüsselt. Die für ein dreidimensionales Raster noch fehlende Größe ist die Höhe der Steine. Diese beträgt $9.6mm$. Der Abstand zwischen zwei übereinander gestapelten Steinen hängt von dem Druck ab, der beim Zusammenstecken geleistet wurde. Dennoch kann dieser vernachlässigt, sprich als Abstand von $0.0mm$ gewertet werden.

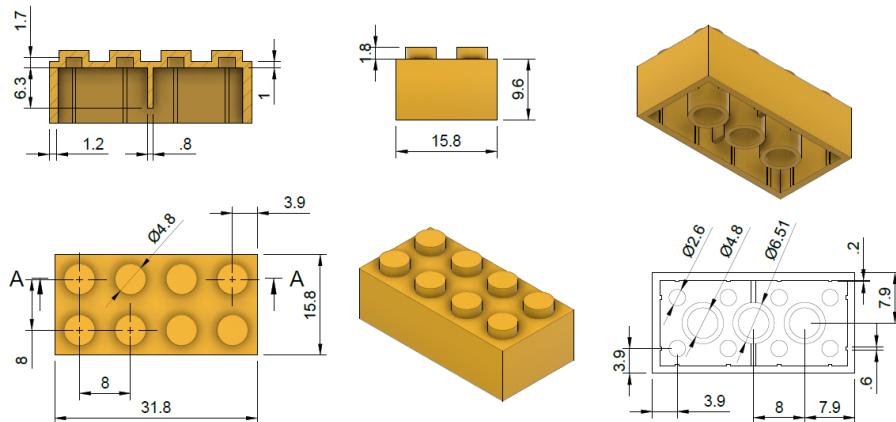


Abbildung 5.3: Maße des Standard 2x4 LEGO Steins [19]

5.7 Mauerwerksbau

Der Mauerwerksbau ist eine Art des Massivbaus, bei welchem Natur- oder Formsteine aufgeschichtet werden, um Wände beziehungsweise Mauern zu errichten. Eine derart gebaute Wand besteht demnach aus Steinen und den dazwischen entstehenden Fugen. Mörtel ist dabei nicht zwangsläufig notwendig. Man spricht von trocken versetzten Steinen oder einer Trockenmauer, wenn darauf verzichtet wird. Heutzutage wird fast ausschließlich mit quaderförmigen Formsteinen gebaut. Zur Beschreibung solcher Formsteine existieren zwei relevante Größen. Die eine ist das sogenannte *Baunennmaß*, mit dem die tatsächliche Größe des Steins angegeben wird. Die andere das *Baurichtmaß*, das

sich aus Baunennmaß und dem Fugenmaß zusammensetzt. Baurichtmaße sind gemäß dem oktametrischen Maßsystem immer ein Vielfaches von $12,5\text{cm}$ (das entspricht $1/8\text{m}$) und mindestens $6,25\text{cm}$. Dies gilt sowohl für die Breite als auch die Höhe der Steine. Das System ist in der DIN 4172 Maßordnung im Hochbau geregelt und ist das fest definierte Grundmaß für das Bauwesen in Europa[27].

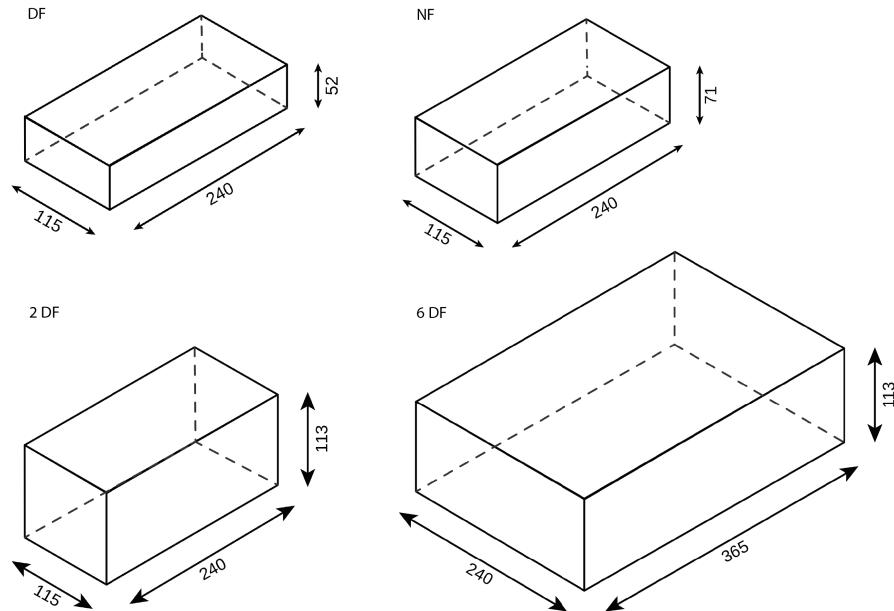


Abbildung 5.4: Darstellung verschiedener Steinformate nach DIN 4172 (Baunennmaß in Millimetern) [24]

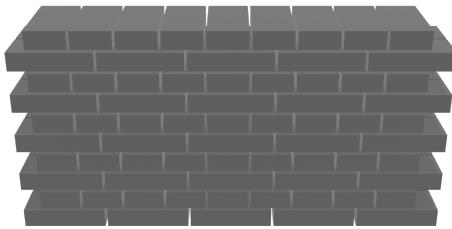
Daraus gehen insbesondere folgende zwei Formate für Ziegelsteine hervor: Das Normalformat (NF) mit $240 \times 115 \times 71\text{mm}$ und das Dünformat (DF) mit $240 \times 115 \times 52\text{mm}$ (Länge x Breite x Höhe). Alle anderen Formate werden mithilfe dieser beiden Grundsteine angegeben. So sind zum Beispiel die in Abbildung 5.4 gezeigten 2 DF und 6 DF Steine eine Kombination aus mehreren Steinen im Dünformat. Dabei sieht die Norm ein Fugenmaß von 1cm für Stoßfugen (vertikal) und 1.2cm für Lagerfugen (horizontal) vor. Für Systeme, die eine schmalere oder keine Fuge benötigen, werden entsprechend größere Steine hergestellt, um der Maßordnung zu entsprechen. Mithilfe des Systems ist man zusätzlich in der Lage Türen und Fenster an die daraus entstehenden Öffnungen anzupassen und vermeidet zeitaufwendiges, nachträgliches Anpassen.

5.7.1 Mauerwerksverband

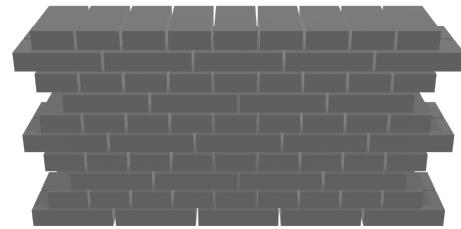
Als Mauerwerksverband bezeichnet man bestimmte, gleichmäßige Anordnungen von Mauersteinen, um einen homogenen Mauerwerkskörper zu erreichen [20]. Damit kann eine gleichmäßige Kraftverteilung innerhalb der Mauer gewährleistet werden. Eine wichtige Rolle nimmt dabei das Überbindemaß ein, welches die Mindestüberlappung von Mauersteinen aus zwei Schichten der Mauer vorgibt. Für das planmäßige Überbindemaß l_{ol} gilt für übliche Mauersteine mit Schichthöhen $h_u \leq 249\text{mm}$ nach DIN EN 1996-1-1:

$l_{ol} \geq 0,4h_u \geq 45\text{mm}$ [5][28]. Zudem wird darin die Mindestwanddicke für tragendes Mauerwerk, sofern aus Gründen der Standsicherheit, der Bauphysik oder des Brandschutzes nicht größere Dicken erforderlich sind"[5], auf $t_{min} = 115\text{mm}$ festgelegt [28]. Dies ist, wie in Abbildung 5.4 zu sehen, exakt die Breite der kleinsten Ziegelformate NF und DF.

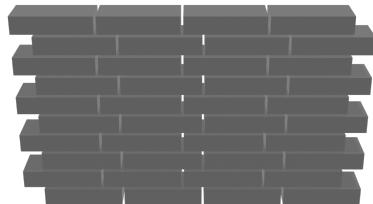
Man unterscheidet zwei Arten von Mauerwerk: das Einsteinmauerwerk und das Verbandsmauerwerk. Wie schon dem Namen zu entnehmen, handelt es sich beim Einsteinmauerwerk um ein Mauerwerk, bei welchem die Wanddicke der Steindicke entspricht. Hier muss das Überbindemaß lediglich über die Wandlängsrichtung eingehalten werden. Bei Verbandsmauerwerk gilt dies zusätzlich für die Wandquerrichtung. [4]



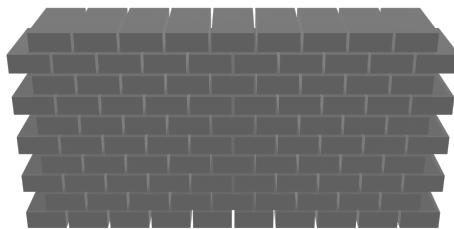
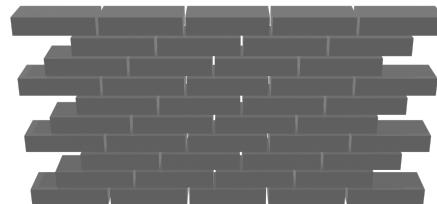
(a) Blockverband.



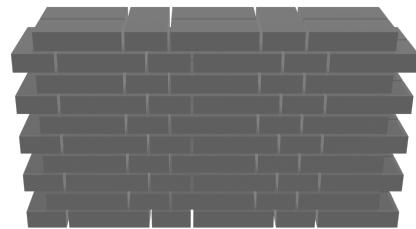
(b) Kreuzverband.



(c) Mittlerer Läuferverband (*Versatz: 1/4 der Steinlänge*). (d) Schleppender Läuferverband (*Versatz: 1/3 der Steinlänge*).



(e) Kopf/Binderverband.



(f) Gotischer Verband.

Abbildung 5.5: Typische Mauerwerksverbände.

Treffen zwei oder mehrere Wandstücke aufeinander, so gilt es diese miteinander zu

verzahnen. Dabei muss nach wie vor das Überbindemaß eingehalten werden. Es treten verschiedene Fälle ein: TODO Bilder!

Ecken Zwei Wandenden stehen im rechten Winkel zueinander und bilden so eine Ecke.

Kreuzungen Zwei Wandenden treffen von beiden Seiten einer dritten Wand an der selben Stelle aufeinander und bilden zusammen eine Kreuzung. Dabei ist es möglich, dass die aufeinandertreffenden Wände jeweils unterschiedliche Dicken aufweisen.

T-Kreuzungen Ein Wandende steht senkrecht auf einer anderen Wand und bildet damit eine T-Kreuzung. Auch hier kann es vorkommen, dass die beiden Wandstücke unterschiedliche Dicken haben. Sowohl bei Kreuzungen als auch T-Kreuzungen kann auf das aufwendige Verahnen verzichtet und stattdessen die sogenannte Stumpfstoßtechnik angewandt werden. Dabei werden Stahlanker zwischen der Wand und den darauf treffenden "Stumpfen" Wandenden verwendet, um die beiden Wände sicher miteinander zu verbinden.

Wandenden Eine Wandende steht frei im Raum, ohne ein anderes Wandstück zu berühren. Dafür muss der verwendete Mauerwerksverband zu einem geraden Abschluss gebracht werden. Öffnungen innerhalb eines Wandstücks können in der selben Art behandelt werden, da der vorherrschende Verband in den betroffenen Schichten gerade unterbrochen werden muss. Über Öffnungen für Fenster und Türen wird ein sogenannter Sturz gelegt, welcher ebenfalls in den der Wand zugrunde liegenden Mauerwerksverband eingebunden werden muss.

Die Lösungen für die oben genannten Situationen variieren je nach angestrebten Mauerwerksverband und der verwendeten Modulgröße stark. Gleichzeitig beeinflussen unterschiedliche Ecklösungen die Art in der ein Wandende gemauert werden muss. TODO Bild von ecklösung und kreuzung für läuferverband und einem größerem verband:

DIN 1053-1 (wurde durch DIN EN 1996-1-1 ersetzt) anstoßendes Wandstück: der Bereich an dem zwei Wandsegmente aneinanderstoßen z.B. eine Ecke Verzahnung: aus <https://baulexikon.beuth.de/VERZAHNUNG.HTM>: Verzahnung, im Mauerwerksbau übliche Technik, beim Herstellen einer Wand eine Verbindungsstelle für eine später zu errichtende und in die bereits bestehende einzubindende Wand den Verbandsregeln entsprechend vorzubereiten. Es gibt Lochverzahnung, stehende und liegende Verzahnung. Nur die letztgenannte Verzahnungsart gilt nach DIN 1053-1 als ausreichende Verbindung zwischen tragenden und aussteifenden Mauerwerkswänden Überbindemaß ist wichtig um Mauerwerksverbände zu bewerten

TODO Rund Wände, keine 90 Grad Ecken

5.8 BRep

5.9 Definitionen

5.9.1 Bausteintyp

5.9.2 Baustein

5.9.3 Wand

6 Konzept

7 Realisierung

7.1 Modellierung

Blenderplugin, IFCWallTypes Module die das Raster vorgeben etc.

7.2 Wall Detailing

Als „Wall Detailing“ (sprich das „Detailieren von Wänden“) wird in dieser Arbeit der Vorgang ein als geometrischer Körper definiertes Wandstück in ein konkretes Mauerwerk zu überführen, bezeichnet. Innerhalb des IFC Standards werden einige mathematische/-geometrische Repräsentationen der sogenannten *IFCWall* unterstützt, um neben einfachen Boxen auch komplexere Formen abbilden zu können. Beispielsweise ist es möglich in das Modell eines Hauses zunehmend dünner werdende Wandstücke, kurvige Wandstücke oder Wandstücke, welche nur durch ein arbiträres Vieleck beschrieben werden können zu integrieren. Allerdings ist es für die Fallstudien dieser Arbeit zunächst ausreichend nur Wandstücke, welche als einfacher geometrischer Quader vorliegen, zu beachten. Das nachfolgende schrittweise Vorgehen weist aufgrund des annähernd gleichen Ergebnisses zwangsläufig Ähnlichkeiten zu dem von Usmanov et al. auf [1].

7.2.1 Konvertieren des IFC zu BREP

Den ersten Schritt stellt das Extrahieren aller notwendigen Daten aus dem vorliegenden IFC Modell dar. Für die Fallstudien dieser Arbeit sind sowohl alle Objekte des Typs *IFCWall* als auch die, etwa durch Fenster oder Türen entstehenden, daran angeknüpften Objekte vom Typ *IfcOpeningElement* (siehe 5.1.4) von Interesse. Zusätzlich werden aus den, in den *IfcPropertySets* der Wandstücke hinterlegten Daten, Informationen über das zu verwendende Modul ausgelesen. Mithilfe der Werkzeuge der in Kapitel 5 vorgestellten Python Bibliothek *ifcopenshell* (siehe 5.2.3) ist dies intuitiv möglich. TODO Sätze zum Code, Code TODO kleines Klassendiagramm von Wall/WallLayerGroup und Opening und BrickInformation

7.2.2 Überprüfen der modellierten Wände

Filtern

Da sich diese Arbeit vorerst ausschließlich mit quaderförmigen Wandstücken beschäftigt, müssen zunächst alle zuvor aus dem IFC Modell extrahierten Wandstücke auf diese Eigenschaft geprüft werden. Somit ist gewährleistet, dass lediglich passende Wandstücke

an die nachfolgenden Schritte weitergegeben werden. TODO Satz zum Code, Code iscubic

Anwenden des Moduls

Mit dem zu jedem Wandstück festgelegten Modul werden nun alle Wandstücke in Schichten aufgeteilt. Deren Höhe entspricht im Normalfall der Höhe des jeweiligen Moduls. Lediglich die oberste Schicht kann durch falsch modellierte Wandstücke eine niedrigere Schichthöhe aufweisen. Dies ist der Fall, wenn die Gesamthöhe des Wandstücks nicht exakt einem Vielfachen der Höhe des Moduls entspricht und ein nicht aufzuteilender Rest existiert. Das Aufteilen in Schichten erleichtert es im Anschluss Berechnungen an Wandstücken durchzuführen und Beziehungen zwischen ihnen zu finden.

Kombinieren passender Wandstücke

Eine solche Beziehung stellen etwa Wandstücke dar, die zwar in dem Modellierungsprozess des Gebäudes durch mehrere einzelne Objekte realisiert wurden, eigentlich aber eine Einheit darstellen. Daher werden in diesem Schritt alle Wandstücke miteinander verglichen und eventuell kombiniert, sodass jeweils ein gefundenes Paar durch ein einzelnes Wandstück representiert wird. Um zwei Wandstücke zu kombinieren müssen folgende Eigenschaften gelten:

- * Beide Wandstücke verwenden das selbe Modul und sind während der Modellierung mit den gleichen Wandtyp annotiert worden. -> Dies verhindert das Kombinieren unterschiedlich dicker Wände
- * Die lokalen Z-Achsen beider Wandstücke sind parallel. -> Dies verhindert das Kombinieren ungleich rotierter Wandstücke
- * Sie stehen auf der selben Höhe oder versetzt um ein Vielfaches der gemeinsamen Modulhöhe.
- * Die lokalen X-Achsen beider Wandstücke sind ebenfalls parallel.
- * Mindestens eine Schicht des einen Wandstücks berührt oder überlappt eine des anderen.

TODO: Bilder und Erklärung der verschiedenen Fälle Steht ein Wandstück in X-Richtung versetzt auf einem anderem, so ist es notwendig diesen Versatz während dem nachfolgenden Detailing zu berücksichtigen. Ignoriert man diesen Versatz kann das zu den in Abbildung TODO gezeigten Fehlern führen. Dieser, nachfolgend als x_offset bezeichnete Versatz ist definiert durch die Differenz zwischen der kleinsten lokalen X-Koordinate aller Schichten eines Wandstückes und der lokalen X-Koordinate der zu betrachtenden Schicht. Der daraus resultierende Wert wird später dazu verwendet den anzuwendenden Mauerwerksverband erst an der passenden Stelle zu beginnen. Dadurch erzielt man einen einheitlichen Verband über das gesamte Wandstück und verhindert den in Abbildung TODO gezeigten Fehlerfall. Eine weitere Eigenschaft, die aus dem Kombinieren mehrerer Wandstücke entstehen kann, ist das vorhandensein unterbrochender Schichten beziehungsweise mehrerer Schichten auf einer Höhe innerhalb eines Wandstückes. Dies ist ebenfalls in Abbildung TODO dargestellt. Durch das Einbeziehen des x_offset können derartige Situationen jedoch ebenfalls gelöst werden, da für jedes Teilstück einer Schicht ein eigener x_offset berechnet wird. Auch dies ist der Abbildung TODO zu entnehmen.

Beziehungen finden

Nun wird die aus dem vorherigen Schritt entstandene neue Menge an Wandstücken auf weitere Beziehungen untersucht. Für das Wall Detailing relevante Beziehungen stellen Ecken, T-Kreuzungen und X-Kreuzungen dar. Es werden wie oben wiederum nur die Wandstücke miteinander verglichen, die mit dem selben Wandtyp annotiert wurden und das gleiche Modul verwenden. Diese Beziehungen ähneln sich stark und besitzen demnach einige geteilte Eigenschaften: * Die lokalen Z-Achsen beider Wandstücke sind parallel. * Sie stehen auf der selben Höhe oder versetzt um ein Vielfaches der gemeinsamen Modulhöhe. * Mindestens eines der beiden Wandstücke endet auf einem anderen, sodass mindestens eine Schicht das andere Wandstück direkt berührt. Außerdem Durchlaufe alle Wandstücke und schau, ob diese andere berühren. Dadurch können Ecken, T-Kreuzungen und Kreuzungen gefunden werden. Schwierigkeiten: Verschiedenste Arten solche Ecken und Kreuzungen zu modellieren

7.2.3 Lösen der Beziehungen

Je nach gewählten Verband müssen z.B. Ecken (deren Baupläne im vornherein definiert wurden) so angeordnet werden, dass die dazwischenliegenden Wandstücke lückenlos eingefüllt werden können. Dafür muss ein sogenannter "plan_offset" für jede Wand und jede Ecke gefunden werden. Dieser gibt an, an welchem Index der Bauplandefinition das Wandstück (von unten) beginnen muss, um insgesamt einen einheitlichen Wandkörper ohne Lücken zu bilden. Voraussetzung dafür ist natürlich ein passender Eckplan zu dem gewählten Verband.

Schwierigkeiten: Eckpläne insgesamt, Ecken ragen in die sie bildenden Wände hinein. Diese müssen dementsprechend verkleinert werden.

7.2.4 Anwenden der Lösung

Ablaufen aller Ecken und Wände und einsetzen der Ziegel gemäß den gefundenen Lösungen.

7.2.5 Export

Abhängigkeitsgraph und Ontologie für Regelwerk

8 Proof of Concept

9 Fazit und Ausblick

Literatur

- [1] Vjačeslav Usmanov, Jan Illetško und Rostislav Šulc. „Digital Plan of Brickwork Layout for Robotic Bricklaying Technology“. In: *Sustainability* 13.7 (Apr. 2021), S. 3905. doi: 10.3390/su13073905. URL: <https://doi.org/10.3390/su13073905>.
- [2] Chengran Xu u. a. „Optimal brick layout of masonry walls based on intelligent evolutionary algorithm and building information modeling“. In: *Automation in Construction* 129 (Sep. 2021), S. 103824. ISSN: 0926-5805. doi: 10.1016/J.AUTCON.2021.103824.
- [3] Kathrin Dörfler u. a. „Additive Manufacturing using mobile robots: Opportunities and challenges for building construction“. In: *Cement and Concrete Research* 158 (Aug. 2022), S. 106772. ISSN: 0008-8846. doi: 10.1016/J.CEMCONRES.2022.106772.
- [4] *05_maurerfibel_kap-4.pdf*. https://www.kalksandstein.de/media/08_downloadcenter/05_maurerfibel_kap-4.pdf. (Accessed on 06/29/2023).
- [5] *Bemessung von Ziegelmauerwerk nach DIN EN 1996-3/NA:2019-12*. https://www.wienerberger.de/content/dam/wienerberger/germany/marketing/documents-magazines/instructions-guidelines/wall/DE_MKT_DOC_POR_Bemessung_Ziegelmauerwerk.pdf. (Accessed on 06/29/2023).
- [6] *BIM for Health and Safety in Construction | Autodesk University*. <https://www.autodesk.com/autodesk-university/article/BIM-Health-and-Safety-Construction-2017>. (Accessed on 04/18/2023).
- [7] *blender.org - Home of the Blender project - Free and Open 3D Creation Software*. <https://www.blender.org/>. (Accessed on 02/16/2023).
- [8] *BlenderBIM Add-on - beautiful, detailed, and data-rich OpenBIM*. <https://blenderbim.org/>. (Accessed on 02/16/2023).
- [9] *Building Information Modeling – Wikipedia*. https://de.wikipedia.org/wiki/Building_Information_Modeling. (Accessed on 02/16/2023).
- [10] *Fachkräftemangel und Rohstoffpreise – Die Deutsche Bauindustrie*. <https://www.bauindustrie.de/zahlen-fakten/bauwirtschaft-im-zahlenbild/fachkraeftemangel-und-rohstoffpreise>. (Accessed on 03/31/2023).
- [11] *IFC Formats - buildingSMART Technical*. <https://technical.buildingsmart.org/standards/ifc/ifc-formats/>. (Accessed on 02/16/2023).
- [12] *IfcOpenShell. IfcOpenShell - The open source IFC toolkit and geometry engine*. <https://ifcopenshell.org/>. (Accessed on 05/05/2023).
- [13] *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*. <https://www.iso.org/standard/63141.html>. (Accessed on 05/05/2023).

- [14] *Industry Foundation Classes – Wikipedia*. https://de.wikipedia.org/wiki/Industry_Foundation_Classes. (Accessed on 02/16/2023).
- [15] *Industry Foundation Classes (IFC) - buildingSMART Technical*. <https://technical.buildingsmart.org/standards/ifc>. (Accessed on 02/16/2023).
- [16] buildingSMART International. *IFC 4.3.1.0 Spezification*. <https://ifc43-docs.standards.buildingsmart.org/>. (Accessed on 05/05/2023).
- [17] buildingSMART International. *IFC 4.3.1.0 Spezification Chapter 1 Scope*. <https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/content/scope.htm>. (Accessed on 05/05/2023).
- [18] ISO - ISO 16739-1:2018 - *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema*. <https://www.iso.org/standard/70303.html.norm>. (Accessed on 02/16/2023).
- [19] LEGO Brick Dimensions and Measurements - Christoph Bartneck, Ph.D. <https://www.bartneck.de/2019/04/21/lego-brick-dimensions-and-measurements/>. (Accessed on 07/07/2023).
- [20] Mauerwerksverband - Baulexikon. <https://baulexikon.beuth.de/MAUERWERKSVERBAND.HTML>. (Accessed on 06/29/2023).
- [21] Microsoft Word - *ConstructionLifecycleManagementwithBIM_080109*. https://www.researchgate.net/profile/Yusuf-Arayici/publication/243972464_Building_information_modelling_BIM_for_Construction_Lifecycle_Management/links/54f4456c0cf2f9e34f094781/Building-information-modelling-BIM-for-Construction-Lifecycle-Management.pdf. (Accessed on 02/16/2023).
- [22] Mohd Nasrun u. a. „Impact of Fragmentation Issue in Construction Industry: An Overview; Impact of Fragmentation Issue in Construction Industry: An Overview“. In: (). doi: 10.1051/C. URL: <http://dx.doi.org/10.1051/matecconf/20141501009>.
- [23] Revit-Software | BIM-Software | Autodesk. <https://www.autodesk.de/products/revit/>. (Accessed on 02/16/2023).
- [24] Steinformat – Wikipedia. <https://de.m.wikipedia.org/wiki/Steinformat>. (Accessed on 07/25/2023).
- [25] The open source BIM collective. <https://github.com/opensourceBIM>. (Accessed on 02/16/2023).
- [26] Top 10 Benefits of BIM in Construction. <https://bim360resources.autodesk.com/connect-construct/top-10-benefits-of-bim-in-construction>. (Accessed on 04/18/2023).
- [27] DIN 4172:2015-09 Maßordnung im Hochbau. Norm. 2015.
- [28] Eurocode 6: Bemessung und Konstruktion von Mauerwerksbauten - Teil 1-1: Allgemeine Regeln für bewehrtes und unbewehrtes Mauerwerk. Norm. 2012.