

BIM-basierte Bauprozessplanung: Automatisierte Ableitung von Precedence Relationships durch eine räumliche Anfragesprache

Simon Daum, Alexander Braun
TU München, Deutschland
simon.daum@tum.de

Abstract. Die Bauprozessplanung ist noch immer ein manueller und zeitaufwendiger Vorgang. In dieser Veröffentlichung wird eine Methode untersucht die Prozessplanung anhand eines Building Information Models unterstützt und vereinfacht. Dabei werden mit Hilfe der Anfragesprache QL4BIM Abfragen durchgeführt, welche die Ableitung eines Vorrangsbeziehungsgraphen (Precedence Relationship Graph) ermöglicht. Aus diesem wird ein vorläufiger Prozessplan generiert, der als Grundlage für die weitere Prozessplanung verwendet werden kann.

1. Einleitung

Building Information Modeling (BIM) propagiert die Bereitstellung eines digitalen Gebäudemodells bereits in der frühen Planungsphase eines Bauprojekts. Dieses frühzeitig verfügbare Modell kann theoretisch viele nachgeschaltete Prozesse unterstützen, jedoch werden die Möglichkeiten der BIM-Technologie bei weitem nicht ausgereizt. Das fällt im Bereich der Terminplanung auf, in der produktivere Arbeitsweisen auch durch den Einsatz von BIM noch nicht etabliert wurden.

Die Terminplanung stellt sicher, dass alle Teilaktivitäten in einem Bauprojekt in einer zeitlich sinnvollen Reihenfolge und in fest definierten Intervallen ablaufen. Dabei müssen Planer nicht nur technologische Randbedingungen berücksichtigen sondern auch die Verfügbarkeit von Ressourcen wie Material, Maschinen, Arbeitern und Raum beachten.

Obwohl ein BIM Model die Terminplanung vereinfachen könnte, fehlt es an geeigneten computerbasierten Methoden. Stattdessen ist die Terminplanung immer noch eine zeitaufwendige und fehleranfällige, manuelle Tätigkeit. Die Autoren betrachten dies als einen unnötigen Einschnitt im BIM -Workflow.

In aktuellen Veröffentlichungen werden verschiedene Arten der Unterstützung der Terminplanerstellung diskutiert. Eine Möglichkeit ist eine Teilautomatisierung der Prozessgenerierung basierend auf bereits bekannten Prozessdaten (Mikulakova et al. 2010). Dieses Verfahren kann durch die Abfrage von geeigneten Kenndaten einzelner Bauteile (z.B. Wanddicke oder Position des Bauteils) aus einer Datenbank ähnliche Bauteile zuordnen und entsprechende Prozessinformationen zur Verfügung stellen. Zusätzlich können ähnliche Bauteile zusammengefasst werden. Der entstandenen Gruppe wird anschließend lediglich eine Prozessinformation zugewiesen.

Die Prozessplanung kann jedoch auch durch die automatisierte Erkennung von technologischen Abhängigkeiten vereinfacht und zumindest teilautomatisiert werden (Huhnt 2005). Diese Abhängigkeiten sind die wichtigsten Bedingungen in der Bauplanung.

Das Ziel der vorgestellten Methode ist es, einen vorläufigen Terminplan automatisch auf Grundlage eines BIMs zu erstellen. Dieser Terminplan dient dem Planer anschließend als Grundlage für weitere Verbesserungen und Detaillierungen die nicht automatisch erkannt werden konnten.

Der Ansatz umfasst die folgenden Schritte:

1. Die Identifizierung von Abhängigkeiten zwischen Bauteilen durch die räumliche Anfragesprache QL4BIM in Kombination mit gespeichertem Expertenwissen.

2. Erstellung eines *Precedence Relationship Graphen*, der die technologischen Abhängigkeiten zwischen Bauteilen und Konstruktionsaufgaben darstellt.
3. Erzeugung eines vorläufigen Terminplans aus dem erstellten Graphen.

Das angewendete Expertenwissen umfasst technologische Abhängigkeiten die nicht veränderbar sind. Beispielsweise kann eine Deckenplatte nicht ohne tragende Struktur darunter errichtet werden. Zur Bestimmung der konkreten Abhängigkeiten in einem Projekt wird die räumliche Konfiguration von Bauteilen durch eine formale Anfragesprache ausgewertet. Diese Analyse betrachtet unter anderem gleichzeitig die Topologie und die directionale Ausrichtung von Bauteilen. So kann das kombinierte Attribute "berührt von oben" zwischen Stütze und tragender Decke genutzt werden um eine technologische Abhängigkeit zu identifizieren: Die Decke muss vor der Stütze fertiggestellt werden. Allerdings ist die rein geometrische Auswertung eines Modells nicht ausreichend. Zusätzlich muss auch das semantische Modell Informationen liefern um zwischen verschiedenen, gleichwertigen Schlussfolgerungen entscheiden zu können: Ein tragende Wand muss vor der angrenzenden Decke errichtet werden, wohin gegen eine nicht-tragende Wand erst nach Bodenplatte und Decke realisiert werden kann. Die Analyse des semantischen Modells liefert in diesem Beispiel das Attribut „tragend“ bzw. „nicht tragend“.

Das endgültige Ergebnis des vorgestellten Ansatzes ist ein automatisch erzeugter, vorläufiger Terminplan der auf der Analyse des Modells durch QL4BIM und des erzeugten *Precedence Relationship Graphen* basiert. Dieses Vorgehen bringt folgende Vorteile mit sich:

1. Es steht eine Rohversion der vorläufigen Terminplanung zur Verfügung. Dadurch reduziert sich der Aufwand der detaillierten Planung rapide.
2. Die konkrete Analyse eines Modells kann mit wenig Arbeitsaufwand an die Gegebenheiten des jeweiligen Projekts angepasst werden.

2. Anfragesprache

Die Bauindustrie setzt zunehmend auf den Einsatz von BIM-Technologie. Vor allem in Großprojekten versprechen modellbasierte Methoden einen besseren Umgang mit den immensen Datenmengen. Mögliche Fehler in Bezug auf Design und Projektplanung sind im Idealfall leichter in der Datenbasis auszumachen, können so frühzeitig erkannt und vor der Bauphase korrigiert werden. Allerdings können unentdeckte Fehler in der Planung auch beim Einsatz moderner BIM-Werkzeuge zu Schwierigkeiten bei nachgeschalteten Prozess führen. Hohequalitative Ergebnisse und ein effizienter Workflow in der Bauphase kann nur erreicht werden, wenn die Datengrundlage fehlerfrei ist und den zuvor definierten Projektanforderungen genügt.

Daher ist eine leistungsfähige und effiziente, formale Analyse von Gebäudemodellen nötig. Diese kann mit der Anfragesprache QL4BIM erreicht werden (Daum & Borrmann 2014). Im Gegensatz zu einer generellen Anfragesprache wie SQL die im Bereich der relationale Datenbanken den de facto Standard darstellt, ist QL4BIM speziell auf die Analyse von Gebäudemodellen ausgerichtet. Als Datenbasis für QL4BIM dient ein Gebäudemodell das im Industry Foundation Classes (IFC) Format vorliegt. Bei einer Analyse sind die Beziehungen zwischen den im Model enthaltenen Entitäten entscheidend.

Deswegen unterstützt QL4BIM folgende Relationsarten zur Selektion von Bauteilen in Anfragen:

- Topologische Relationen (Abbildung 1):
Disjoint, Equals, Touch, Contains, ContainedBy, Overlaps, Covers und CoveredBy
- Direktionale Relationen (Abbildung 2):
Above, Below, EastOf, WestOf, NorthOf, SouthOf
- Metrische Relationen:
IsFarther, IsCloser
- Attributive Relationen:
Verknüpft über Attributwerte (z.B. IDs) referenzierter Entitäten, vergleichbar mit einem Join in einer relationalen Datenbank

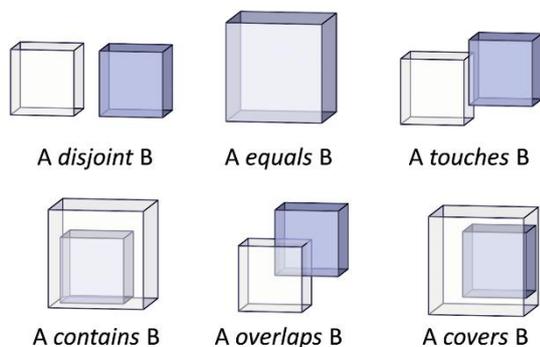


Abbildung 1: Die topologischen Operatoren in QL4BIM

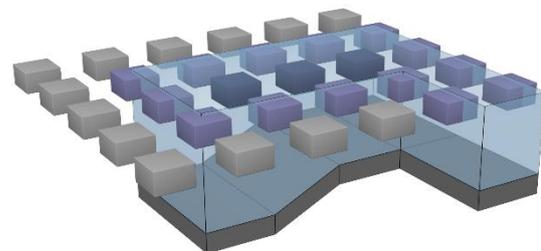


Abbildung 2: Der direktionale Operator Above

Die Auswertung der räumlichen Relationen geschieht auf Basis der expliziten Dreiecksgeometrie der untersuchten Bauteile. Da das IFC-Produktmodell auch implizite Geometrierepräsentationen unterstützt wird in der QL4BIM-Engine jedes Bauteil vorprozessiert und mit einer expliziten Geometrierepräsentation (*IfcTriangulatedFaceSet*) ausgestattet. Zur Auswertung des topologischen Attributes zwischen zwei Bauteilen wird folgendermaßen vorgegangen (Abbildung 3):

- a) Dreieckstest durchführen um die Existenz schneidender und berührender Dreiecke zwischen den Bauteilen zu untersuchen.
- b) In Kombination mit Strahl-basierten Inside/Outside-Tests erfolgt die eindeutige topologische Klassifizierung.
- c) Zur Effizienzsteigerung werden R*-Bäume zur räumlichen Indizierung von Bauteilen und ihren Dreiecken genutzt.

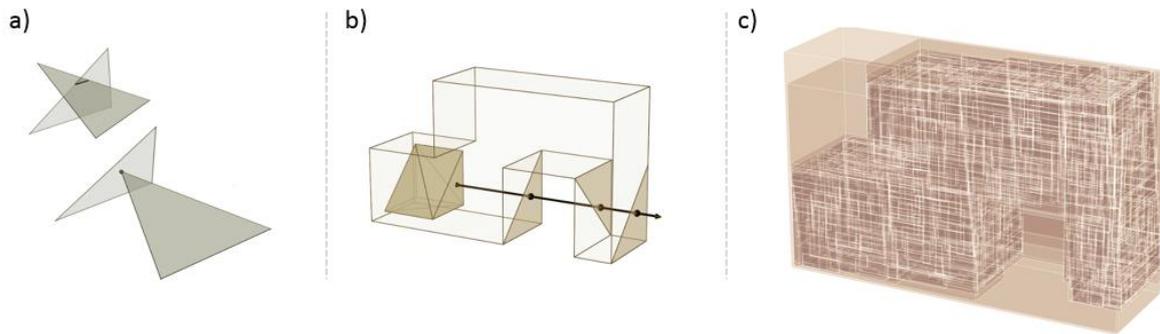


Abbildung 3: Konzept des QL4BIM Systems für topologische Anfragen

Zur Auswertung der direktionalen Relationen wird auf das Vorhandensein von sich überlagernden Dreiecken in die jeweilige Raumrichtung geprüft. Auch hier werden für die Kandidatensuche R*-Bäume eingesetzt. Das Verfahren ist in (Daum & Borrmann 2013a) detailliert beschrieben. Neben der Auswertung von Relationen zwischen Entitäten unterstützt QL4BIM auch die Abfrage von Objektattributen.

Zur Verarbeitung der Anfragen wird Live LINQ genutzt. Hierbei werden QL4BIM Anfragen per .NET Code definiert, on-the-fly kompiliert und anschließend in einer LINQ Anweisung ausgeführt. Das Live LINQ System ist in (Daum & Borrmann 2013b) detailliert beschrieben. Im Folgenden werden vier beispielhafte QL4BIM Anfragen gezeigt. Die erste Abfrage verwendet keine räumlichen Prädikate sondern zeigt die prinzipielle Abfragesyntax die auf Prädikatenlogik basiert. Neben der Filterung durch IFC-Klassen macht Beispiel 2 und 3 Gebrauch von topologischen und direktionalen Prädikaten. Beispiel 4 demonstriert wie verschachtelte, räumliche Abfrage realisieren werden können.

```

1 IfcProducts.Where(p =>
2   p is IfcWallStandardCase && p.GlobalId == "1pJQicIrH4UR_2BqvRP")

```

Abbildung 4: Beispiel 1 - Selektiere Wände mit der GlobalId 1pJQicIrH4UR_2BqvRP

```

1 IfcProducts.Where(p => {
2   var wall1 = IfcProducts[25];
3   return p is IfcWallStandardCase && p.Overlaps(wall1); })

```

Abbildung 5: Beispiel 2 - Selektiere Wände die Wand 1 (Index 25) schneiden

```

1 IfcProducts.Where(p => {
2   var wall1 = IfcProducts [25];
3   var slab1 = IfcProducts [12];
4   return p.Overlaps(wall1) && p.Above(slab1); })

```

Abbildung 6: Beispiel 3 - Selektiere Wände die Wand 1 (Index 25) schneiden und sich über Platte 1 (Index 12) befinden

```

1 var slab1 = IfcProducts [12];
2 var wallsBelow = IfcProducts.Where(p => p is IfcWall && p.Below(slab1));
3
4 IfcProducts.Where(p => {
5     var column1 = IfcProducts [34];
6     return p.Touch(column1) && !wallsBelow.Any(w => w.Touch(p)); })

```

Abbildung 7: Beispiel 4 - Selektiere alle Wände unter Platte 1. Des Weiteren, selektiere alle Bauteile die Stütze 1 (Index 34) berühren, jedoch keine der vorher selektierten Wände

Die beschriebene Funktionalität von QL4BIM kann nicht nur zur Analyse und Verifikation von Gebäudemodellen eingesetzt werden. In (Daum et al. 2014) wurde gezeigt wie diese Anfragesprache zur Generierung von Nachbarschafts- und Erreichbarkeitsgraphen eingesetzt werden kann. Im Folgenden wird beschrieben wie QL4BIM zur Erstellung von *Precedence Relationship Graphen* benutzt wird.

3. Precedence Relationship Graph

Eine gute Möglichkeit für die Darstellung und Verarbeitung von Abhängigkeiten einzelner Bauteile sind Graphen (Enge 2010). Jeder Knoten symbolisiert ein Bauelement, die Kanten zeigen die Abhängigkeiten der einzelnen Bauteile zueinander. Der entstehende *Precedence Relationship Graph* ist gerichtet, da die Abhängigkeiten stets in eine Richtung wirken. Des Weiteren ist der Graph zyklensfrei nachdem Bauteile nicht gegenseitig voneinander abhängig sein können.

Beispiel:

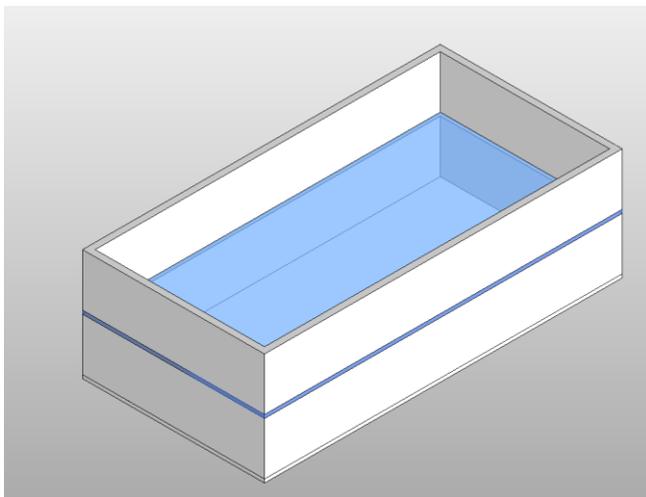


Abbildung 8: Beispielgebäude zur Darstellung der technologischen Abhängigkeiten

Um das Prinzip des Abhängigkeitsgraphen zu erläutern, wird nachfolgend ein Beispielgebäude ausgewertet. Das Testgebäude hat vier Wände und eine Bodenplatte für jede Etage. Folgende Abhängigkeiten können in diesem Beispiel definiert werden: Die Wände im ersten Stock können nicht vor der Bodenplatte oberhalb des Erdgeschosses gebaut werden. Das gleiche gilt für diese Platte und die Wände darunter. Diese Abhängigkeiten werden als technologische Abhängigkeiten definiert. Sämtliche weitere Abhängigkeiten, welche auf einer Baustelle in der Regel auftreten und vom Prozessplaner berücksichtigt werden müssen können nicht automatisiert erfasst werden. Dazu zählen zum Beispiel logistischen Abhängigkeiten.

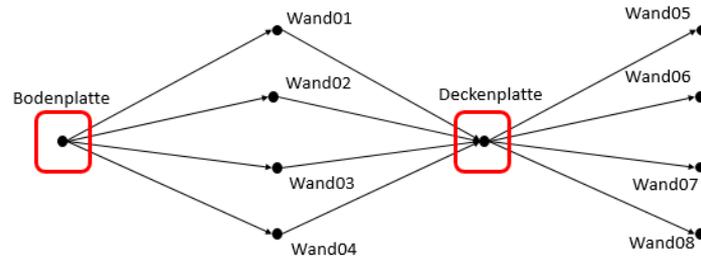


Abbildung 9: Technologische Abhängigkeiten des Beispielgebäudes

Das Diagramm (Abbildung 9) visualisiert die Abhängigkeiten als Graph und zeigt, dass alle folgenden Wände von der Deckenplatte unter ihnen abhängig. Diese Objekte werden in diesem Ansatz als *Checkpoint-Komponenten* klassifiziert. In der Graphentheorie wird ein solcher Knoten Gelenkpunkt genannt, wenn dessen Entfernung aus dem Graphen diesen in mehrere Teilgraphen aufteilen würde (Deo 1974). Diese Bauteile sind besonders wichtige Identifikationspunkte im Abhängigkeitsgraphen, da sie die Gesamtstruktur des Gebäudes strukturiert darstellen.

4. Abhängigkeiten von Bauteilen

Die Abhängigkeiten der einzelnen Bauteile zueinander sollen mit dem in Abschnitt 3 eingeführten *Precedence Relationship Graphen* abgebildet werden. Dieser soll mit Hilfe der Abfragesprache *QLABIM* automatisiert generiert werden.

4.1. Identifizierung von Abhängigkeiten und Generierung des PRG

Für die Generierung sind geeignete Abfragen gegen das Building Information Model notwendig. Um diese zu identifizieren, benötigt es Expertenwissen, welches möglichst allgemeingültig auf eine große Anzahl an Gebäuden anwendbar ist.

Für die Abfragen haben sich mehrere Eigenschaften als geeignet qualifiziert, die Struktur des *Precedence Relationship Graphen* zu erstellen: Für die grobe Strukturierung des Graphen werden in einem ersten Schritt sämtliche tragenden Bauteile gefiltert. Diese werden nach ihrer vertikalen Position sortiert. Dies hat den Hintergrund, dass tragende Bauteile im Normalfall „von unten nach oben“ gebaut werden und somit eine Abhängigkeit zwischen diesen Bauteilen besteht. Mit diesen Informationen kann eine erste Version des Graphen erstellt werden, indem die sich berührenden Bauteile abgefragt werden.

Im nächsten Schritt können die nicht tragenden Bauteile in den Graphen eingefügt werden und diesen erweitern. Dabei werden berührenden, tragenden Bauteile abgefragt und mit den nicht tragenden Bauteilen in Verbindung gesetzt. Fenster oder Durchgänge können so in direkte Abhängigkeit zu der jeweiligen Wand gesetzt werden. Trockenbauteile, welche nach dem Rohbau gebaut werden, sind dementsprechend abhängig von der Deckenplatte „über“ dem betroffenen Stockwerk.

4.2. Abfragen

```
1 var storeys = IfcBuildingStoreys.OrderByDescending(s => s.Name);
2 var productsSets = storeys.Select(s => s.ReferencesElements)
3     .Where(e => GetProperty(„LoadBearing“, „true“))
4 var productToMeetingProducts
5     = productsSets.Select(ps => ps.Select(p =>
6         return new Tuple(p, new List(GetMeetingProductsBelow))));
```

Abbildung 10: QL4BIM Anfrage zur Ermittlung der Relation „Bauteil zu von unten berührenden Bauteilen“

5. Generierung des Terminplans

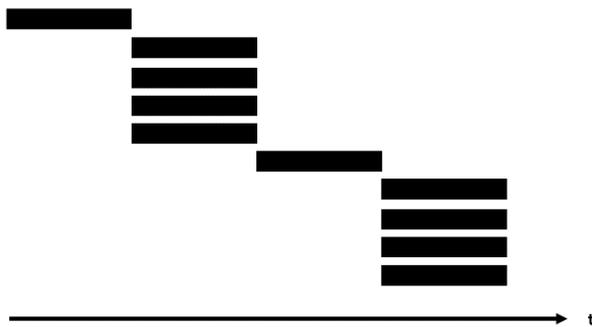


Abbildung 11: Gantt-Diagramm für das Beispielgebäude in Abb. 8

Der Terminplan wird nach Erstellung des Vorrangsbeziehungsgraphen als Gantt-Diagramm generiert. Dabei wird jedem Knoten beziehungsweise Bauteil eine feste Prozessdauer zugeordnet. Die Prozesse werden in der Reihenfolge ihrer Abhängigkeiten angeordnet. Für mehrere Bauteile, die alle von demselben Bauteil abhängen, werden parallel auszuführende Prozesse generiert. Nachdem der Graph gerichtet ist, können diese Informationen direkt abgeleitet werden.

6. Zusammenfassung und Ausblick

Die in diesem Paper vorgestellte Methode zur Generierung eines vorläufigen Prozessplanes basiert auf einer Anfragesprache, welche direkt auf das Building Information Model zugreift. Durch die Erstellung eines Abhängigkeitsgraphen für die einzelnen Bauteile kann eine gute Abschätzung über den zu erwartenden Bauablauf gemacht werden.

Diese Methode stellt jedoch bewusst nur einen ersten Schritt in der Prozessplanung dar. Es können hier keinerlei Aussagen über die jeweilige Prozessdauer gemacht werden. Auch der korrekte Bauzeitpunkt nicht tragender Elemente ist nur schwer korrekt abzuschätzen. Diese Prozesse können jedoch angelegt und in direkten Zusammenhang mit den abhängigen Bauteilen gebracht werden. Dadurch ist auch hier eine logische Verknüpfung vorhanden, welche bei der weiteren Prozessdetailplanung berücksichtigt werden kann.

Der vorliegende Ansatz kann durch die Ansätze von Huhnt, Enge oder Mikulakova et.al. erweitert werden. Dabei wäre eine Anbindung an eine Prozessdatenbank sinnvoll, welche ein umfangreiches Prozesswissen vorhält um auch die Bauzeiten oder übliche Abhängigkeiten abbilden zu können.

7. Literatur

- Daum, S. et al., 2014. Automated generation of building fingerprints using a spatio-semantic query language for building information models. In *European Conference on Product & Process Modelling*. p. In Press.
- Daum, S. & Borrmann, A., 2013a. Boundary Representation-Based Implementation of Spatial BIM Queries. In *Proc. of the EG-ICE Workshop on Intelligent Computing in Engineering, Vienna, Austria*.
- Daum, S. & Borrmann, A., 2013b. Definition and Implementation of Temporal Operators for a 4D Query Language. In *Proc. of the ASCE International Workshop on Computing in Civil Engineering*.
- Daum, S. & Borrmann, A., 2014. Processing of Topological BIM Queries using Boundary Representation Based Methods. *Advanced Engineering Informatics*, In Press.
- Deo, N., 1974. Graph Theory with Applications to Engineering and Computer Science (Prentice Hall Series in Automatic Computation). Available at: <http://dl.acm.org/citation.cfm?id=1096898> [Accessed July 16, 2014].
- Enge, F., 2010. *Muster in Prozessen der Bauablaufplanung*.
- Huhnt, W., 2005. Generating sequences of construction tasks. ... *Conference on Information Technology in Construction* Available at: <http://itc.scix.net/data/works/att/w78-2005-KN-w1-1-Huhnt.pdf> [Accessed February 28, 2014].
- Mikulakova, E. et al., 2010. Knowledge-based schedule generation and evaluation. *Advanced Engineering Informatics*, 24(4), pp.389–403. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S1474034610000546> [Accessed July 15, 2014].