

VERIFIKATION SEMANTISCHER KONZEPTE IN DIGITALEN BAUWERKSMODELLEN MIT OBJECT CONSTRAINT LANGUAGE

Bernold Kraft, Torsten Krämer

Technische Universität Berlin, Fachgebiet Bauinformatik

Abstract: Ein Prinzip des Building Information Modeling ist die Verfügbarkeit und die Anreicherung eines digitalen Gebäudemodells mit weiteren geometrischen und semantischen Informationen. Dabei werden zur Lebenszeit des Modells semantische Informationen als Eigenschaften der Objekte mit impliziten Informationen vermischt, die aus den spezifischen Eigenschaften des BIM-Werkzeugs resultieren. Zur Prüfung der Korrektheit aller Eigenschaften werden Konzepte definiert, die einen Wissenstand zu einem Zeitpunkt untersuchen. Dabei ist die Aussage der Korrektheit nur eine Vorstufe zur Definition einer Menge von Eigenschaften, die zu korrigieren sind. Da eine Korrektur immer in Bezug zur jeweiligen BIM-Plattform steht, ist eine plattform-abhängige Prüfung gegen eine unabhängige Wissensbasis ein erster Schritt, die semantische Korrektheit von digitalen Gebäudemodellen zu erreichen. In diesem Paper wird ein Ansatz präsentiert, der die standardisierte Sprache der Object Constraint Language (OCL) nutzt, um ein Modell einer BIM-Plattform gegen eine Wissensbasis zu prüfen.

Schlüsselwörter: Building Information Modeling, Semantic modeling, Model validation, Object Constraint Language

1. Einführung

Der Prozess des Building Information Modeling (BIM) ist im Prinzip Software-ungebunden. Trotzdem stellt eine zur Arbeitsweise passende BIM Plattform bislang die größte Hürde dar. Ein Ziel dieser Prozesse ist Informationen einzugeben, zu synchronisieren und sie nachgelagerten Prozessen als Eingangsdaten zur Verfügung zu stellen (Eastman et al. 2011).

Ein wesentlicher Fokus von BIM liegt in der Verwaltung von Informationen, die einem oder mehreren Objekten einer Instanz zugeordnet werden können. Der Term Information ist dabei anwendungsbezogen und wird durch die Eigenschaften, die in Relation mit dem Objekt stehen, widerspiegelt. Informationen können auf verschieden Art und Weise in einer Instanz eines Modells eingebettet werden. Nach (Eastman et al. 2009) werden Informationen vom Modellierer als instanzgebundene Attribute gegeben oder sie werden abgeleitet auf Basis anderer bereits

verfügbarer Informationen. Die Software des BIM Werkzeugs selbst enthält Informationen, die in ihrem Metamodell gebunden sind. Dazu gehören unter anderem definierte Bauteiltypen und Vorgaben in Form fest definierter Bauteilstrukturen. Das Erstellen einer Modellinstanz vereint beide Arten von Informationen. Dabei entstehen im Verlauf der Einbettung von Informationen Abhängigkeiten zur Modellierungsplattform, da bestimmte Eigenschaften der Software als Plattform genutzt werden, ohne diese explizit in der Instanz zu definieren. In der Lebenszeit eines digitalen Gebäudemodells formiert sich somit eine Semantik von Komponenten, wie z.B. Bauteilen. Eine Komponente ist hierbei ein benanntes Objekt eines Modells, das über eine Identifikation verfügt. Als Semantik einer Komponente eines digitalen Gebäudemodells wird in diesem Paper die Bedeutung dieser Komponente in der Gesamtkonfiguration des Modells definiert, die durch die Menge aller Eigenschaften, die direkt oder indirekt mit der Komponente in Beziehung stehen.

Ein Modell wird zur Lebenszeit mit semantischen Informationen angereichert, die durch zusätzliche Eigenschaften der Objekte zum Ausdruck gebracht werden. Bei der Abbildung von Informationen auf Eigenschaften gelten dieselben Grundsätze wie bei der Normalisierung von relationalen Datenbanken. Die Zielsetzung der Atomarität und der funktionale Unabhängigkeit von Eigenschaften führen zu einer kombinatorischen Vielfalt. Dabei stellen bestimmte Gruppierungen von konkreten Ausprägungen der Eigenschaften semantische Konzepte dar. Diese semantischen Konzepte () sind konform zu einer oder mehrerer Aussagen () über das betreffende Objekt des Modells. Weiterhin setzt die Korrektheit berechneter Informationen die semantische Korrektheit der existierenden Informationen voraus. Die semantische Korrektheit einer Komponente setzt die Konjunktion von Vollständigkeit und formaler Korrektheit voraus. Zur Prüfung der Korrektheit von Modellen wurden verschiedene Verfahren vorgeschlagen (Eastman et al. 2009, Pauwels et al. 2011), die auf dem gemeinsamen Referenzmodell der Industry Foundation Classes (IFC) fußen auf der Basis von EXPRESS (ISO 10303-11).

In diesem Paper wird ein Ansatz präsentiert, der eine aktive Brücke zwischen einem öffentlich zugänglichen Objektgraphen einer Instanz eines BIM Werkzeuges und einer externen Wissensbasis erzeugt. Dabei wird die formale Korrektheit eines Objekts durch die Ausprägung aller Attribute mit einer bekannten Wertemenge gewährleistet. Die Frage, ob die Semantik des Objekts korrekt ist, kann durch die Prüfung gegen ein oder mehrere semantische Konzepte beantwortet werden. Dabei wird aus einem praxisnahen Beispiel in einem ersten Schritt eine formale Repräsentation einer Wissensbasis abgeleitet. Diese kann dann in eine Laufzeitbedingung auf Basis der Object Constraint Language (OMG OCL 2012) umgeformt werden. Die semantische Korrektheit eines Objektes wird an die Erfüllung aller formulierten Bedingungen gebunden.

2. Ansätze der Modellvalidierung und Abgrenzung

(Eastman et al. 2009) stellten einen Überblick über den Stand der Modellvalidierung und den prinzipiellen Ansätzen und Arbeitsweisen zusammen. Dabei stehen existierende Software-Lösungen im Vordergrund. Offensichtlich wird im Konsens der Veröffentlichung, dass keine allgemein anwendbare Sprache zur Modellierung von bauspezifischen Bedingungen an ein digitales Gebäudemodell existiert.

Die IFC selbst stellt auf Basis der EXPRESS Sprache eine Möglichkeit bereit, Regeln als Laufzeitbedingungen einer Instanz der IFC zu definieren (ISO 10303-11). Auf Basis dieser Spezifikation wurde von (Niemeijer et al. 2009) ein prototypisches Werkzeug vorgestellt, das die Eigenschaften der Modellelemente der IFC nutzt, um eine Modellvalidierung anzustoßen. Die IFC erfasst jedoch nur einen Bruchteil der benötigten Informationen und wurde nicht für den Zweck der Modellierung von Fachwissen entworfen (Niemeijer et al. 2009). Zur Kompensation dieses Mangels existiert eine Erweiterung durch die Entität `IfcPropertySet`, die eine definierte Menge von `IfcPropertySingleValue` enthält. Eine solche Eigenschaft besteht aus einer Identifikation (`IfcIdentifier`), dem Wertebereich (`IfcUnit`) und einem spezifischen Wert (`IfcValue`) des Bereichs.

Semantische Konzepte, die formal oder informal gegeben werden können, sind Teil einer Wissensbasis. Der Konzeptraum einer Wissensbasis ist generell offen und wird im Verlauf der Lebenszeit eines Modells ständig verfeinert. Somit existiert keine direkte Verbindung zwischen dem oder den Konzepten einer Komponente und der Komponente des Modells selbst. Ob eine Bauwerkskomponente die korrekte Semantik trägt, muss zur Laufzeit geknüpft an einen Wissenszeitpunkt auf Basis der Aussagenlogik beantwortet werden. Dabei ist eine Wissensbasis immer Anwendungsbezogen. Eine Komponente kann verschiedene Anwendungen haben. (Pauwels et al. 2011) stellen auf Basis der Logiksprache Notation 3 (N3) (Berners-Lee et al. 2008) eine Methode vor, die die Ontology Web Language (W3C OWL2¹) zur Repräsentation von semantischen Konzepten und die Erweiterung N3 als Grundlage für das logische Schließen verwendet. Die vorgestellten Methoden verwenden die IFC als Metamodell. Die semantische Korrektheit der Typen aller Objekte dieser Instanz wird vorausgesetzt. Die Art und Weise, wie eine Instanz aus einem BIM Werkzeug in die IFC transformiert wird, regelt das jeweilige Werkzeug. Um das Problem der verschiedenartigen Transformationen einzugrenzen, werden entsprechende Validierungswerkzeuge und Zertifizierungsprozesse für Software-Produzenten angeboten (Geiger et al. 2012). Die Prüfung von Software-Modellen gegen eine allgemeine

1 <http://www.w3.org/TR/owl2-overview/>

formale Wissensbasis ist aktueller Gegenstand der Forschung in der Informationstechnologie. Eine Verbindung zwischen einer plattformunabhängigen Wissensbasis und einem Software-Metamodell kann die semantische Korrektheit des Modells in Bezug auf den gesamten Entwicklungsprozess qualitativ absichern. (Queralt et al. 2012) beschreiben hierzu einen Weg, der eine Abbildung zwischen einer Beschreibungslogik (hier *ALCJ*) und einem definierten Fragment der Object Constraint Language (OCL) benutzt.

Die Anwendung der Modellvalidierung im BIM Umfeld in der Praxis hat gezeigt, dass die Qualität der digitalen Modelle in Bezug auf die Konformität zu aufgestellten Regeln verbessert werden kann (Eastman et al. 2009). Dabei wird dem Modellierer selbst aber das Erkennen der notwendigen Änderungen überlassen, obwohl die Korrekturen oftmals klar formal definiert werden können. Der Modellexport und Modellimport ist in vielen Fällen asymmetrisch. So steht der Modellierer vor dem Problem, notwendige Änderungen per Hand nachzutragen. Andererseits entstehen Abhängigkeiten zwischen dem BIM Werkzeug und der Instanz, so dass Korrekturen nur abstrakt formuliert werden können. Damit eine Änderung effizient im Kontext eines BIM Werkzeuges erfolgen kann, ist somit eine Verbindung zwischen plattformabhängigen Modell und der abstrakten Wissensbasis notwendig, die auf Grundlage eines abstrahierten Meta-Modells operiert.

3. Exemplarische Problemstellung

Die Prüfung der semantischen Korrektheit der Informationen eines einzelnen Bauteils setzt eine formale Korrektheit und lokale Vollständigkeit der Eigenschaften dieses Bauteils in Bezug auf die Anwendung der Informationen voraus. Wird eine einfache Aussage wie z.B.

Aussage **A** sei „das Bauteil ist eine Brandschutzwand“

als Information gewertet, so müssen alle Eigenschaften des modellierten Bauteils mit dieser Aussage konform sein. Voraussetzung ist, dass die Eigenschaften vollständig sind und die Aussage jeder einzelnen Eigenschaft qualifiziert werden kann. Aus dem genannten Beispiel ergeben sich drei Eigenschaften (vergl. Tabelle 1, Zeile 1-3). Eigenschaften wie Dicke, Material und Modelltyp sind Teil des Metamodells der Software. Die Rohdichteklasse ist eine Eigenschaft, die implizit mit der Funktion der Wand und dem Material in Abhängigkeit steht, hier jedoch explizit gegeben wurde.

Tabelle 1: Fachspezifische Eigenschaften und semantisches Konzept

	Identifikation der Eigenschaft	Qualifizierte Wertemenge	Semantisches Konzept $SC(A)$
1	Modelltyp	Wand, ...	Wand
2	Funktion	Brandschutzwand, ...	Brandschutzwand
3	Brandschutzklasse	F30, F90, ...	F90
4	Dicke	$10 \text{ cm} \leq ? \leq 36,5 \text{ cm}$	$? \geq 17,5 \text{ cm}$
5	Material	Mauerwerk, Stahlbeton, ...	Mauerwerk
6	Rohdichteklasse	1,4; 1,8; ...	1,8

Dabei wird die Aussage durch das Vorhandensein eines semantischen Konzeptes, das konform zur anfänglichen Aussage **A** ist, realisiert. Im Projektablauf werden Informationen verschiedener Fachdomänen und Anwendungsgebiete im Modell abgebildet. Die brandschutztechnische Begutachtung kann erst auf Basis einer existierenden Geometrie und einer räumlichen Konzeption erfolgen. Daher existieren bereits Abhängigkeiten in Form von geometrischen Parametern, wie z.B. der Dicke und anderen Parametern, wie z.B. Materialien, die bereits im Vorfeld definiert wurden. Weiterhin existieren Auflagen in Form von Normen (hier in DIN 4102-4/A1:2004-11). Das semantische Konzept wird daher erweitert werden müssen. In der weiteren Diskussion wird eine Mauerwerkswand betrachtet (vergl. Tabelle 1, Zeile 4-6).

4. Konkrete Modellierung des Beispiels

Die bisherige Formulierung war unabhängig von einem möglichen Software-Metamodell. Exemplarisch wird im Folgenden die Metamodellierung von Revit der Firma Autodesk (Version 2013) verwendet. Diese Modellierung zeichnet sich durch eine Typisierung von Bauteilen und verschiedener Abstraktionsebenen der Geometrie aus. Im weiteren Verlauf wird die Betrachtung von Komponenten auf Bauteile mit geometrischer Repräsentation eingeschränkt.

Die Abbildung von Eigenschaften erfolgt als Attribut eines Bauteils. Dabei geht eine große Anzahl von BIM Werkzeugen den gleichen Weg, wie schon aus Sicht der IFC beschrieben. Ein Attribut besteht hiernach aus einer Identifikation und einem konkreten Wert einer Wertemenge. Die Identifikation kann – muss aber nicht – gleichzeitig der semantische Name eines Attributs sein. Die Wertemenge kann – muss aber nicht – einschränkbar sein. Diese Funktionalitäten sind in den Werkzeugen verschieden. Zudem existiert eine Abgrenzung zwischen den Begriffen Attribut und Parameter. Nach (Lee 2006) können Parameter als Verhaltensmerkmale von Komponenten eines digitalen Gebäudemodells angesehen werden.

Die Definition von Building Object Behaviour (BOB) beschreibt den Zusammenhang zwischen Parameter und Zustandsänderung des Objektes eines Modells aus Sicht des Modellierers. Da Verhaltensmerkmale generell Abhängigkeiten zum Metamodell aufweisen (hier Revit), ist die Definition eines Parameters im Einzelnen nicht vom Modell trennbar. Im Beispiel des semantischen Konzepts ist die Dicke der Wand ein Parameter aus Sicht des Metamodells von Revit.

Revit selbst stellt vier Abstraktionsebenen bereit. Ausgehend von einem „Exemplar“, das ein physisch existentes Objekt eines Gebäudemodells darstellt, wird eine Abstraktionsschicht „Typ“ für ein konkretes Typensystem angeboten. Jedes Exemplar hat genau einen Typ. Jeder Typ ist Teil einer generischen Geometriebeschreibung – der „Familie“. Ein oberer Abschluss existiert als reine Klassifikationsschicht, die als „Kategorie“ bezeichnet wird. Anders als bei den übrigen drei Ebenen gibt es zwischen Familie und Kategorie keine eindeutige Abbildung. Eine typische Konvention in der Modellierung kann die Ausnutzung des Typensystems sein, da die Dicke als Parameter bereits auf Ebene des Typs festgelegt werden muss. Die konkrete Instanz nutzt diese Metainformationen. Ein Typ kann einen Namen tragen, der zur Identifikation eines Konzepts genutzt werden kann. Zur Laufzeit kann ein Exemplar den Typ ändern, wenn er zur selben Familie gehört. Das Typensystem von Revit beruht auf einem System, dass geometrische Eigenschaften an Subelemente vererbt.

5. Formalisierung semantischer Konzepte

Die bisherige Formulierung des Konzepts $SC(A)$ beruhte auf dem Wissen, an welchen Eigenschaften des Konzepts eine Aussage A bestätigt werden kann. Diese Eigenschaften wurden auf Attribute eines Revit-Typs abgebildet. Damit eine Aussage prüfbar werden kann, muss das Konzept auf Grundlage mengentheoretischer Teilaussagen formalisiert werden. Bei diesem Übergang wird ein umfassendes Wissensgebiet der Beschreibungslogik (DL) (engl. description logic) angerissen, die die Brücke zur Formulierung von Wissensbasen darstellt (Baader et al. 2003). DL zerlegt die Wissensbasis in zwei Teile (Box). Während die Terminal Box (TBox) die Axiome zur Bildung von Konzepten und Begriffen ausdrückt, stellt die Assertion Box (ABox) grundlegende Aussagen zu Individuen auf, die reale Objekte der Welt darstellen. Ausgehend von der größten gemeinsamen Basis T , dem Super-Konzept, werden alle Konzepte formuliert. In einem ersten Schritt ist die Benennung der Individuen, als grundlegende Aussage notwendig.

Material $\equiv \{Mauerwerk, \dots\}$
Modelltyp $\equiv \{Wand, \dots\}$
Funktion $\equiv \{Brandschutzwand, \dots\}$
Brandschutzklasse $\equiv \{F90, \dots\}$

Weiterhin wird hier das Konzept als das größte gemeinsame Konzept definiert, das eine qualifizierte existenzielle Belegung der notwendigen Attribute beschreibt.

Komponente $\sqsubseteq \top \sqcap \exists material. Material \sqcap \exists modeltyp. Modelltyp$
Wand $\sqsubseteq Komponente \sqcap \forall modeltyp. Wand$
KomponenteMitBrandschutz $\sqsubseteq Komponente \sqcap \exists brandschutzklasse. Brandschutzklasse$
 $\sqcap \exists funktion. Funktion$
Brandwand $\sqsubseteq KomponenteMitBrandschutz \sqcap Wand \sqcap \forall brandschutzklasse. F90$
 $\sqcap \forall funktion. Brandschutzwand$

Tatsächlich ist der Sprachumfang von DL in der Basissprache \mathcal{AL} (engl. attributive language) beschränkt auf konkrete Individuen. Eine Erweiterung $\mathcal{Q} - \mathcal{SHJO}$ (Lutz 2001) ermöglicht die zumindest theoretische Verwendung von rationalen Zahlen und Ordnungsrelationen. Die Rollenrestriktion für die Rohdichteklasse und Dicke der Wand sind somit als universelle Bedingungen abbildbar.

MauerwerksBrandschutzwand $\sqsubseteq Brandwand \sqcap \forall rohdichteklasse. \geq_{1,8} \sqcap \forall dicke. \geq_{17,5cm}$

Ungelöst verbleibt an dieser Stelle die Einheitenproblematik. Ein Bezugssystem existiert im Rahmen von DL nicht. Es ist Teil der Interpretation.

6. Prüfung von Modellen zur Laufzeit

Eine Software als BIM Werkzeug setzt stets eine mehr oder minder stimmige Metamodellierung der Komponenten eines Modells voraus. Die Grundlage bildet das Prinzip der modellgetriebenen Softwarearchitektur. Teile des Metamodells können öffentlich sein und in einem Application Programming Interface (API) zugänglich sein oder durch eine Exportschnittstelle verfügbar sein. Andere Teile, die eine empfindliche Geschäftslogik enthalten sind verdeckt bzw. im Export nicht abgebildet. Um eine Software auslieferbar und anwendbar zu gestalten, wird der Konzeptraum beschränkt auf die Entwurfszeit der Software. Dies ist der vertragliche Zustand einer Software gegenüber dem Endanwender. Alle Funktionalitäten der Software beruhen auf diesem geschlossenen Konzeptraum.

Als einheitliche Modellierungssprache wurde von Objekt Management Group (OMG) die Unified Modeling Language (OMG UML 2011) veröffentlicht. UML ist dabei eine Realisierung der Meta-Object Facility (MOF), die ebenfalls von der OMG gepflegt wird (OMG MOF 2011). MOF unterteilt den Prozess der Modellierung in vier Ebenen beginnend mit der Ebene M0, die die konkrete Instanz darstellt und als Modell realisiert wird. Die Ebene M1 beschreibt das Metamodell eines Modells. Alle Modelle eines Metamodells sind konform zu diesem Metamodell. Die Ebenen M2 und M3 werden als Metameta-Modellebene sowie Metameta-Metamodellebene bezeichnet. Durch die Erkenntnisse bei der Umsetzung des MOF Konzepts für das Eclipse Modeling Project², wurde MOF in Complete MOF (CMOF) und Essential MOF aufgeteilt (EMOF), das eine Untermenge von CMOF ist. Als Referenzimplementierung wurde das Meta-Metamodel Ecore im Eclipse Modeling Framework umgesetzt. Einen Überblick über die Beziehungen zwischen MOF, EMOF und Ecore geben (Mohamed et al. 2007).

Zur Beschreibung struktureller Verhaltenscharakteristika wurde die Sprache OCL von der OMG entworfen (Willink 2011). Mit der Version 2 der Modellierungssprache UML wurde OCL als eigenständige Sprache definiert. OCL verbindet Ausdrücke der Mengenlogik (z.B. Quantoren) mit einem typisierten Software-Metamodell und kann auf dieser Basis semantische Teilkonzepte innerhalb eines Modells validieren. Da OCL als eine von Seiteneffekten freie Sprache umgesetzt ist, kann sie im Modell selbst keine direkten Änderungen erzielen. Daher kann sie nicht von außen zur Inferenz von Fakten in das Modell verwendet werden.

Das Metamodell der BIM Plattform entstammt einem unbekanntem Meta-Metamodell. Damit eine Prüfung auf Basis von OCL möglich wird, ist die Transformation des exportierten Modells in ein MOF konformes Modell notwendig. Die Gewinnung des Modells kann teilweise auf Halbautomatismen beruhen, insofern eine Reflektion für Sprachelemente des Software-Metamodells vorhanden ist. Bei der Erzeugung des spezialisierten Metamodells spielt der Anwendungsfall, also die konkrete Menge von semantischen Konzepten eine große Rolle. Als Ziel Metamodell wird ein generisches Komponenten-Metamodell definiert, das die prinzipiellen Konzepte fasst. Als Ableitung kann ein spezialisiertes Metamodell mit konkreten Eigenschaften für den Anwendungsfall erzeugt werden (vergl. Abb. 1).

2 <http://www.eclipse.org/modeling/>

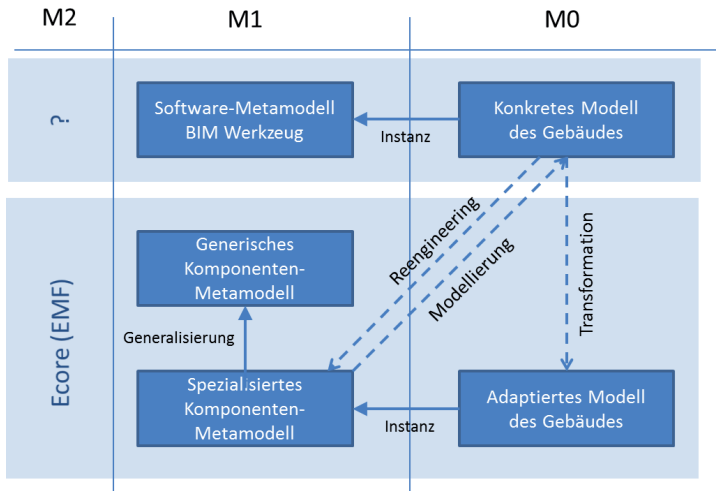


Abbildung 1 - Metamodell-Ansatz für ein adaptiertes Gebäudemodell

Existiert dieses spezielle Metamodell so kann über eine Menge aus Abbildungsvorschriften ein adaptiertes Modell des digitalen Gebäudemodells aus dem ursprünglichen Modell gewonnen werden. Für die spätere Änderungsverfolgung ist es notwendig, diese Relationen zu persistieren.

Das Ecore-Modell, das auf der M2 Stufe Grundlage für das Eclipse Modeling Framework (EMF) (Eclipse Foundation, EMF) bildet, wird als Metamodell des spezialisierten Komponenten-Metamodells verwendet.

7. Abbildung semantischer Konzepte in das Metamodell

Ein spezialisiertes Metamodell kann einerseits durch Modellierung einer Wissensbasis als auch durch Reengineering eines Modells mit aggregierter Wissensbasis erzeugt werden. Im beschriebenen Fall ist das Konzept einer Brandschutzwand maßgebend. In jedem Fall sind anwendungsbezogene Eigenschaften als Attribute in das Metamodell übernommen worden.

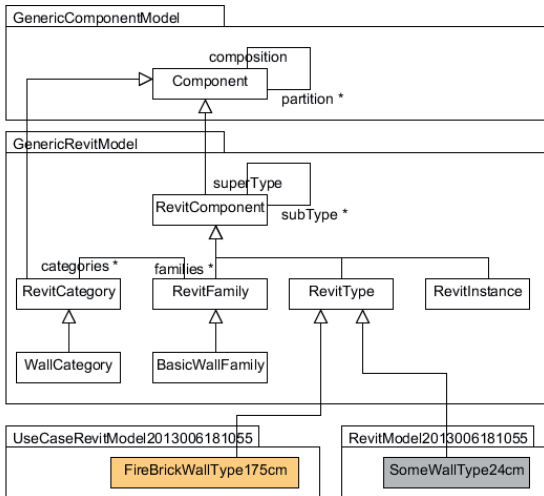


Abbildung 2 – Modellierungsstufen des Komponentenmodells in UML

Die Schwierigkeit besteht in der Abbildung des formalisierten in \mathcal{Q} – *SHO* beschriebenen Konzepts. Nach dem Prinzip von MDA werden der Entwurfszeitpunkt des Metamodells und die Lebenszeit der Instanz modelltechnisch in Ebenen getrennt. Zum Entwurfszeitpunkt können nur existenzielle Rollenbeschränkungen abgebildet werden. Die universellen Rollenbeschränkungen werden zur Laufzeit per OCL prüffähig, wenn, und nur dann wenn, die existenziellen Rollen existieren. Damit ist eine Bindung von Wissensbasis der Konzepte mit existenziellen Bedingungen und dem Metamodell notwendig. Weiterhin ist das Modell (die Instanz) ab dem Zeitpunkt der Konstruktion an das Metamodell gebunden. Um eine generelle Prüffähigkeit mit OCL zu erreichen, ist demzufolge die Einbindung aller Konzepte auf ein Objekt des Typs „RevitInstance“ notwendig. Hieraus leitet sich der logische Schluss ab, dass die existenziellen Konzepte auf Schnittstellen (Klassen ohne Verhalten) abgebildet werden.

KomponenteMitBrandschutz \mapsto *IFireResistentComponent*
Komponente \mapsto *IInvalidComponent*

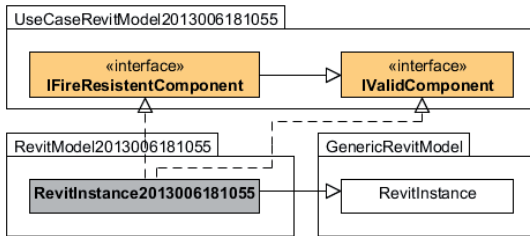


Abbildung 3 – Zustand der Wissensbasis zum Zeitpunkt 2013006181055

Die Prüfung, ob ein Objekt des Typs „RevitInstance“ das gültige semantische Konzept repräsentiert, stellt im Kern eine Implikation dar, die die Eigenschaft der Funktion mit denen der Brandschutzklassifikation und Dicke verbindet.

```

RevitInstance.allInstances() → select(oclIsKindOf(IFireResistentComponent))
→ select(oclAsType(IValidComponent).modelltyp = Modeltyp
:: Wand and oclAsType(IValidComponent).material = Material :: Mauerwerk) → forall( funktion
= Funktion :: Brandschutzwand implies brandschutz = BrandschutzKlasse :: F90 and rohdichteklasse
≥ 1.8 and dicke ≥ 0.175)

```

Dabei werden die Individuen der ABox, namentlich der Modelltyp, das Material, die Funktion sowie die Brandschutzklasse als Aufzählungstypen in das Metamodell übernommen. Die Problematik der Einheiten kann hier durch die Konvention von Standardeinheiten gelöst werden.

8. Schlussfolgerungen

Ausgehend von der Problematik, dass ein digitales Gebäudemodell zur Lebenszeit mit semantischen Konzepten angereichert wird und dass die Eigenschaften des konkreten BIM Werkzeugs die Art und Weise der Einbettung beeinflussen, wurde ein generelles Verfahren zur Formalisierung dieser semantischen Konzepte gezeigt. Das Ziel der Prüfung dieser Konzepte ist die Beschreibung einer Differenz zwischen dem Ist-Modell und dem Soll-Modell. Voraussetzung hierfür ist die Berechnung der Objekte, die nicht dem Soll-Modell genügen. Bedingt auf der Vermischung von Eigenschaften des Metamodells der Software und der Wissensbasis müssen notwendige Änderungen bei Widersprüchen in der Instanz auch die Logik des Software-Metamodells berücksichtigen.

In diesem Paper wurde eine Möglichkeit gezeigt, auf Basis der Sprache OCL zur Laufzeit Prüfungen von semantischen Konzepten an einem adaptierten Modell zu bewerkstelligen. Generell ist hierbei die Abbildung von semantischen Konzepten auf Modellelemente eines Software-Metamodells problematisch. Die Welt eines offenen Konzeptraums, der zum Zeitpunkt der Adaption einer BIM Instanz unvollständig ist, kollidiert mit dem zum Konstruktionszeitpunkt fixen Metamodell einer Software. Eine Versionierung von Modellen und Metamodellen in einem Repository hilft zumindest den Zeitpunkt beider Seiten konkret abzubilden. Auch birgt die Verbindung der Semantik eines Bauteils und dessen Geometrie eine weitere Komplexität, die hier gänzlich unbeachtet blieb. Hierzu sind weitere Schritte notwendig (Kraft und Huhnt 2013). In jedem Fall kann eine Aussage über die semantische Korrektheit eines Modells nur beantwortet werden, wenn alle notwendigen Objekte einer Instanz zum Prüfzeitpunkt bekannt sind. Da OCL gleichzeitig als Query Language fungiert, ist die Ermittlung einer Änderungsmenge direkt aus der Bedingung per Negation möglich.

Vorteil der hier gezeigten Methodik ist die Nutzung des Umfangs der standardisierten Sprache OCL. Bei der Adaption einer BIM Instanz können Metainformationen zum Modell direkt genutzt werden und Verhaltensmuster implementiert werden. Ein spezielles Problem der DL basierenden Sprachen sind numerische Konzepte. In OCL wird analog zu UML eine Unterscheidung zwischen Datentyp und Konzept (Klasse) betrieben. Ordnungsrelationen sind ein fester Bestandteil von OCL. Nachteil der vorgestellten Methodik ist, dass eine offene Konzeptwelt nur mit erheblichem Aufwand abgebildet werden kann und die in OCL ausgedrückten Regeln von der Typisierung des adaptierten Komponentenmodells zum Konstruktionszeitpunkt abhängen. Weiterhin ist die Inferenz von Fakten über OCL nicht direkt möglich. Für eine tatsächliche offene Konzeptwelt ist dies ein Hauptmerkmal.

Literatur

Baader, F., D. Calvanese, D. McGuinness, D. Nardi, und P. Patel-Schneider (2003). *The Description Logic Handbook*. Cambridge University Press.

Berners-Lee, T., D. Connolly, L. Kagal, Y. Scharf, und J. Hendler (2008). „N3logic: A logical framework for the world wide web“. *Theory and Practice of Logic Programming* 8, Nr. 3, pp. 249–269.

Eastman, C., P. Teicholz, R. Sacks und K. Liston (2011), *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. John Wiley and Sons.

Eastman, C., J. Lee, Y. Jeong und J. Lee (2009) „Automatic rule-based checking of building designs“. *Automation in Construction* 18, Nr. 8, pp. 1011–1033

Geiger, A., K. Hausknecht, und K. Linhard (2012) „BIM Software Zertifizierungskonzept von buildingSMART“. In *Forum Bauinformatik 2012*, pp. 1–8. Bochum: Europäischer Universitätsverlag Bochum.

- Kraft, B. und W. Huhnt (2013), „Identifying Incorrectly Specified Objects for Specific Information Derivations in Digital Building Models“, eg-ice Wien Juli 2013 (als Beitrag angenommen)
- Lee, G., R. Sacks und C. Eastman (2006), „Specifying parametric building object behavior (BOB) for a building information modeling system“. *Automation in construction* 15, Nr. 6, pp. 758–776.
- Lutz, C. (2001) „Adding Numbers to the SHIQ Description Logic – First Results“. In *IN PROCEEDINGS OF THE EIGHTH INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING (KR2002)*, 191–202. Morgan.
- Mohamed, M., M. Romdhani, und K. Ghedira (2007), „MOF-EMF Alignment“. In *Third International Conference on Autonomic and Autonomous Systems*, Athens.
- Niemeijer, R.A., B. Vries, und J. Beetz (2009), „Check-mate: Automatic constraint checking of IFC models“. *Managing IT in Construction/Managing Construction for Tomorrow*. London: CRC Press, pp 479–486.
- Queralt, A., A. Artale, D. Calvanese, und E. Teniente, (2012), „OCL-Lite: Finite reasoning on UML/OCL conceptual schemas“. *Data & Knowledge Engineering* 73, Nr. 0 (März 2012): 1–22. doi:10.1016/j.datak.2011.09.004.
- OMG, Object Constraint Language 2.3.1, 1/2011
- OMG, Meta Object Facility 2.4.1, 8/2011
- OMG, Unified Modeling Language 2.4.1, 8/2011
- Pauwels, P., D. Van Deursen, R. Verstraeten, J. De Roo, R. De Meyer, R. Van de Walle und J. Van Campenhout (2011), „A semantic rule checking environment for building performance checking“. *Automation in Construction* 20, Nr. 5, pp. 506–518.
- Willink, E. (2011), „Modeling the OCL Standard Library“. *Electronic Communications of the EASST* 44

