

Techniken der Verwaltung dynamischer digitaler Bauwerksmodelle für Revitalisierungsvorhaben¹

Thomas Hauschild, Reinhard Hübler, André Borrmann
Bauhaus– Universität Weimar
Informations– und Wissensverarbeitung²

Zusammenfassung

Modellverwaltungssysteme sind eine geeignete technologische Basis zum Management digitaler Bauwerksmodelle bei Planungstätigkeiten für den Neubau als auch für die Revitalisierung von Bauwerken. Die Unterstützung von Revitalisierungsprozessen impliziert für den Entwurf integrierter Planungsumgebungen spezifische Anforderungen wie die Repräsentation von Informationen, die mit verschiedenen Typen von Vagheit behaftet sind, die Notwendigkeit, den Soll– sowie den Ist– Zustand des Bauwerks abzubilden und die Fähigkeit des Umgangs mit temporal inkonsistenten Modellzuständen. Die erforderliche Dynamik der Domänenmodelle und die erforderliche Nutzbarkeit in Virtual Enterprises stellen weitere Ansprüche an die Realisierungsbasis der Modellverwaltungssysteme.

Zur Implementierung derartiger Systeme erweist es sich als vorteilhaft, Eigenschaften objektorientierter Programmiersprachen mit nichtstatischen Typsystemen auszunutzen, da diese durch die vorhandene Metaebene sowie Introspektions- und Reflektionsmechanismen eine effiziente Realisierungsbasis bereitstellen. Zur effektiven Unterstützung synchroner kooperativer Planungstätigkeiten innerhalb einzelner Fachdisziplinen wurde ein Benachrichtigungsmechanismus realisiert, der an das Modellverwaltungssystem angekoppelte Fachapplikationen über nebenläufig vorgenommene Modifikationen am zugehörigen Domänenmodell oder an Projektinformationen informiert. Weiterhin existiert ein Mechanismus zur vereinfachten Anbindung von existierenden Applikationen, die auf statischen Partialmodellen beruhen oder standardisierte, modellbasierte Austauschformate unterstützen. Abschließend wird eine aus einem zentralen Projektserver, Domänenservern und Domänenclients bestehende hybride Systemarchitektur vorgestellt, die geeignet ist, unter den Randbedingungen kooperativer und geographisch verteilter Arbeit bei Revitalisierungsvorhaben in Virtual Enterprises eingesetzt zu werden.

0 Einleitung

Für die rechnerinterne Repräsentation von Bauwerksinformationen für Vorhaben des Neubaus wie der Revitalisierung von Bauwerken sind digitale Bauwerksmodelle eine geeignete Basis. Allgemein stellen sie eine notwendige Vorbedingung für eine rechnerbasierte Unterstützung

¹ Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Teilprojektes „Digitales Bauwerksmodell als Grundlage der Datenintegrationsebene für Bestandsinformationen“ des SFB 524 „Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken“ gefördert.

² Bauhaus–Universität Weimar,
Informations– und Wissensverarbeitung,
Coudraystraße 13, 99421 Weimar
Telefon: 03643 / 584260, Fax: 03643 / 584292
E-Mail: thomas.hauschild@informatik.uni-weimar.de,
reinhard.huebler@informatik.uni-weimar.de,
andre.borrmann@bauing.uni-weimar.de

der ablaufenden Bauplanungs- und Bauausführungsprozesse dar. Die digitalen Bauwerksmodelle bilden die Grundlage für die Integration der Bauwerksinformationen und zur adäquaten Bereitstellung von für die jeweilige Aufgabe des Bearbeiters notwendigen und relevanten Informationen.

Entwurfsprozesse im allgemeinen und im besonderen diejenigen der Bauwerksrevitalisierung sind durch sehr komplexe Aufgabenstellungen, eine hohe Arbeitsteiligkeit mit zunehmender geographischer Verteilung und phasenweise synchroner bzw. asynchroner Kooperation charakterisiert.

Durch die hohe Komplexität der anfallenden Informationen und den Wunsch nach Modellen, die den Lebenszyklus des betreffenden Bauwerks begleiten werden Ansprüche an die Modelle und ihre rechnerinterne Verwaltung impliziert. [1]

Das Teilprojekt "Digitales Bauwerksmodell für Bestandinformationen" des Sonderforschungsbereichs 524 "Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken" befasst sich mit der Untersuchung und Entwicklung von Methoden zur Modellierung von Bauwerksdaten sowie von Techniken zum Management verteilter, digitaler Bauwerksmodelle.

Bei Revitalisierungsvorhaben bestehen verglichen mit Neubauprojekten gravierende Unterschiede in Bezug auf die Qualität und Quantität der zur Verfügung stehenden Bauwerksinformationen. Die Beschreibungen verschiedener Parameter des bestehenden, zu revitalisierenden Bauwerks sind häufig unvollständig, vage oder unscharf. Durch eine Vielzahl von Faktoren können Differenzen zwischen vorliegenden Dokumenten und dem existierenden Bestand auftreten; häufig beruhen erste Planungstätigkeiten im Bestand auf Vermutungen.

Ein weiterer wesentlicher Unterschied der Revitalisierung zum Neubau besteht in der Notwendigkeit der Abbildung sowohl des Soll- als auch des Ist- Zustandes, der einen initialen, teilweise vagen und unvollständigen Grundzustand des Bauwerksmodells darstellt. Die zur Verfügung stehenden Informationen werden häufig während des Planungsprozesses verfeinert. Daraus resultiert die Notwendigkeit der Unterstützung der Repräsentation von unsicheren, unscharfen und inkonsistenten Bauwerksinformationen sowohl im Modell als auch in dessen Verwaltungskomponenten.

1 Modellierung und Modellverwaltung

Zur Schaffung digitaler Bauwerksmodelle wird das Domänenwissen unter Nutzung des objektorientierten Paradigmas in abstrahierter Form in Taxonomien abgebildet. Diese Taxonomien bestehen bei deskriptiven Modellen aus Kategorien, Klassen mit Attributen sowie Erbschafts-, Aggregations-, und Assoziationsstrukturen. Das konkrete Bauwerk wird durch Instanzen dieser Taxonomieklassen abgebildet.

Da neben den durch Mittel der objektorientierten Modellierung formalisierbaren Informationen auch eine Reihe von nicht oder schwer formalisierbaren Daten während des Planungsprozesses anfallen, müssen auch Informationen wie Kurztexte, digitalisierte Originalzeichnungen, verschiedene CAD-Formate, Bilder von Schäden etc. verwaltet werden können. Durch eine Speicherung in speziellen Attributen der Klassen und deren Instanzen wird es möglich, diese effizient abzufragen bzw. im benötigten Kontext anzubieten.

Bisherige Forschungen haben gezeigt, dass eine Dynamisierung der Domänenmodell- sowie Projektinformationen hinsichtlich deren Inhalte und Strukturierung notwendig ist. Gründe dafür sind unter anderem die sehr langen Bauwerkslebenszyklen und damit Modelllebenszeiten, Änderungen in Vorschriften und Regelwerken, die Anpassbarkeit der Systeme an spezifische Anforderungen bzw. Bausubstanz oder persönliches Know How. Um Objekten zu Planungsständen zuordnen zu können, werden Instanzen mit hierarchisch strukturierten Kennzeichnungssignaturen versehen. Über Suchanfragen, die durch boolesche

Ausdrücke verknüpfte Teilmengen dieser Bäume beschreiben, können verschiedenen Kriterien folgende Objektmengen vom Modellverwaltungskern geliefert werden. Damit wird eine Anbindung von Werkzeugen zur Unterstützung des Prozessmanagements ermöglicht. Modellverwaltungssysteme (MVS) stellen eine technologische Basis zum Management digitaler Bauwerksmodelle dar. Diese müssen einerseits Wissen über die Arbeitsgegenstände der einzelnen beteiligten Fachgebiete, d.h. genannte Taxonomien, abbilden. Andererseits müssen die Informationen des konkreten Revitalisierungsvorhabens verwaltet werden. Das Modellverwaltungssystem muss ferner dem kooperativen und verteilten Charakter der Prozesse Rechnung tragen und den Zugriff auf freigegebene Planungsstände anderer Beteiligter ermöglichen [2]. Ferner sind an derartige MVS noch eine Reihe weiterer Anforderungen wie Unterstützung von Datenversionierungen, Abbildbarkeit von vagen Informationen, Sicherung der Rechtsverbindlichkeit digitaler Modelle usw. zu stellen (Abb.1).

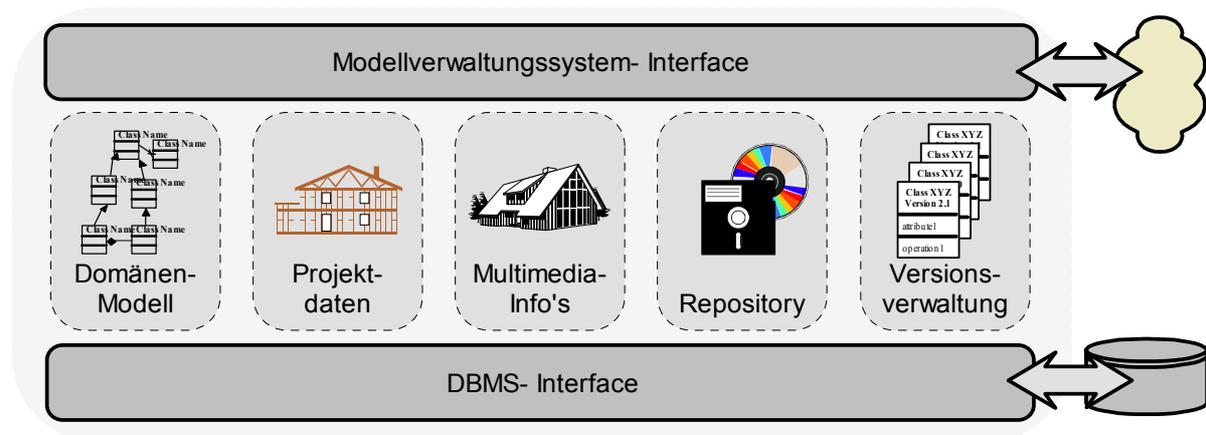


Abbildung 1: Struktur des Modellverwaltungskerns

Für die Realisierung von laufzeitdynamischen Modellverwaltungskernen, die den genannten Anforderungen genügen, gibt es verschiedene Ansätze. Einerseits ist eine vollständige Eigenimplementierung aller Modellverwaltungsfeatures möglich, andererseits kann die Realisierung relevante Eigenschaften bestimmter objektorientierter Programmiersprachen ausnutzen oder die Implementierung kann auf geeigneten Tools basieren.

Im beschriebenen Forschungsprojekt wird die zweitgenannte Implementierungsvariante genutzt, indem die Metaebene objektorientierter Programmiersprachen mit dynamischen Typsystemen für Modellverwaltungszwecke genutzt wird. Diese Sprachen besitzen im allgemeinen sehr effiziente Verfahren des Zugriffs auf Objekte, was darauf basierenden Modellverwaltungssystemen zugute kommt. Ebenso existieren Möglichkeiten der Introspektion, Reflektion und des dynamischen Ladens von Klassenmodulen.

Auf die Informationen im Modellverwaltungssystem kann über ein statisches CORBA-Interface [3] lokal und entfernt zugegriffen werden. Diese Schnittstelle kapselt die notwendige Funktionalität der Metaebene und stellt eine Weiterentwicklung der AKO-Schnittstelle [4] dar.

2 Benachrichtigungsmodul zur Unterstützung synchroner kooperativer Arbeit

Neben dem versionsorientierten, asynchronen Zugriff auf Daten im Modellverwaltungssystem ist von der integrierten Planungsumgebung die phasenweise synchrone Arbeit an einem gemeinsamen Datenbestand durch mehrere Bearbeiter zu unterstützen. Durch diese Form der

kollaborativen Arbeit werden die bei getrennten Datenbeständen auftretenden Inkonsistenzen und der Aufwand zu deren Behebung bei der Zusammenführung gänzlich vermieden. Ausgehend vom Bedarf an Clientapplikationen, deren Datenbestand mit dem des MVS fest gekoppelt bzw. ständig synchronisiert ist, wurde die Architektur für ein Benachrichtigungsmodul auf Grundlage des Modell- Beobachter- Muster entwickelt. Das Konzept sieht vor, dass die clientseitigen, für Darstellung und Interaktion zuständigen Applikationsobjekte die serverseitigen MVS- Objekte beobachten, die das Domänenmodell und dessen Instanzen repräsentieren.

Bei Modifikationen an den Daten der MVS- Objekte werden die verteilt vorliegenden Daten der Applikationsobjekte mit Hilfe eines in das MVS integrierten Benachrichtigungsmoduls auf Grundlage des entkoppelt kommunizierenden CORBA Typed Notification Service synchronisiert [5].

Die dabei zum Einsatz kommende generische Benachrichtigungsschnittstelle ist symmetrisch zur Zugriffsschnittstelle AKO, d.h. zu jeder Modifikationsmethode existiert genau eine Callback- Methode. Letztere übermittelt zudem möglichst viele Informationen über die vorgenommene Modifikation, um einen erneuten verteilten Aufruf durch den Client zu vermeiden. Die Benachrichtigungsschnittstelle muss von den Beobachtern eines MVS- Objektes so implementiert werden, dass der Zustand der Clientapplikation mit dem des Gesamtsystems konsistent gehalten wird (Abb. 2).

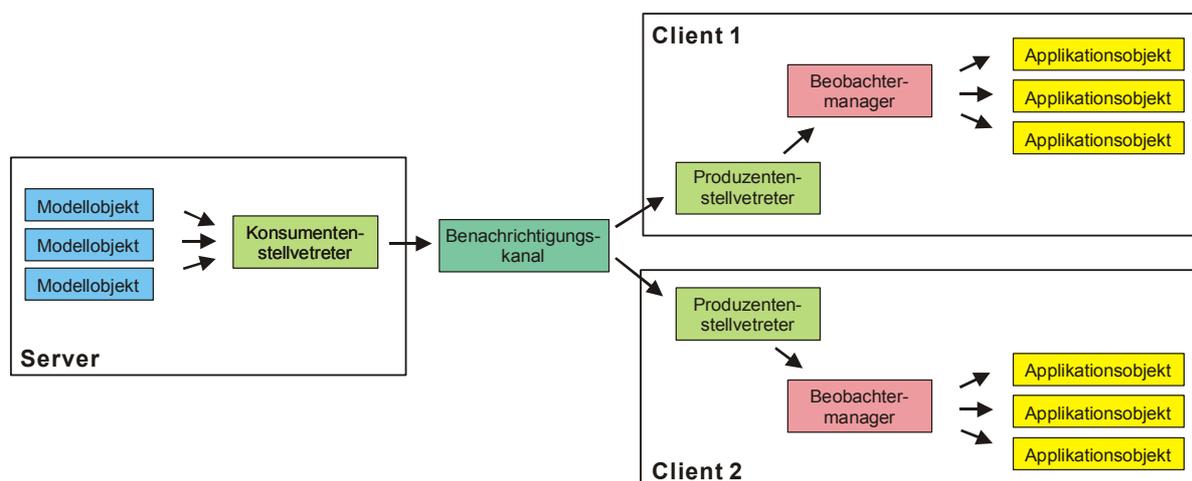


Abbildung 2: Funktionsweise des Benachrichtigungsmoduls

Da alle Clients des Notification Service als CORBA- Objekte implementiert werden müssen, erweist es sich zur Begrenzung des Verwaltungsaufwandes als sinnvoll, die Beobachterfunktionalität mit Hilfe von clientseitigen Beobachtermanagern zu realisieren, welche die Weiterleitung der Callback- Methodenaufrufe an die dafür angemeldeten Beobachter übernehmen.

Durch die Schaffung einer Delegate-Programmschnittstelle für Clients des Modellverwaltungssystems mit Methoden zum An- und Abmelden von Beobachtern direkt an MVS- Objekten bleibt die Funktionsweise des Benachrichtigungsmoduls vollkommen transparent für den Anwendungsprogrammierer.

Die auf Grundlage des Benachrichtigungsmoduls implementierte generische Beobachterapplikation dient der entfernten Überwachung und Fehleranalyse eines Modellverwaltungsservers.

3 Konnektoren für die Anbindung von bereits bestehenden Fachapplikationen

Existierende Fachapplikationen zur Unterstützung des Planungsprozesses basieren i.a. auf einem statischen Datenmodell. Neben den generischen Anwendungen, die auf die Dynamik des Domänenmodells reagieren können, werden zudem auch zukünftig Anwendungen Teil des Gesamtsystems sein, deren Ausschnitt aus dem Domänenmodell nur einer geringen oder gar keiner Dynamik unterworfen ist.

Das entwickelte Konzept der Konnektoren ermöglicht den Anschluss einer Applikation mit statischem Datenmodell an das Modellverwaltungssystem. Dabei sind vor allem solche Anwendungen geeignet, deren internes Datenmodell weitgehend konform zum modellierten Domänenmodell auf dem MVS ist, mit dem sie verbunden werden sollen.

Konnektoren adaptieren die generische AKO-Schnittstelle auf Metaniveau auf eine spezifische Schnittstelle gegenständlichen Niveaus mit Entitäten der Anwendungsdomäne. Sie bieten dabei sowohl Zugriffs- als auch Benachrichtigungsfunktionalität für Applikationen mit statischem Datenmodell, die naturgemäß nicht auf Änderungen am Datenmodell reagieren können.

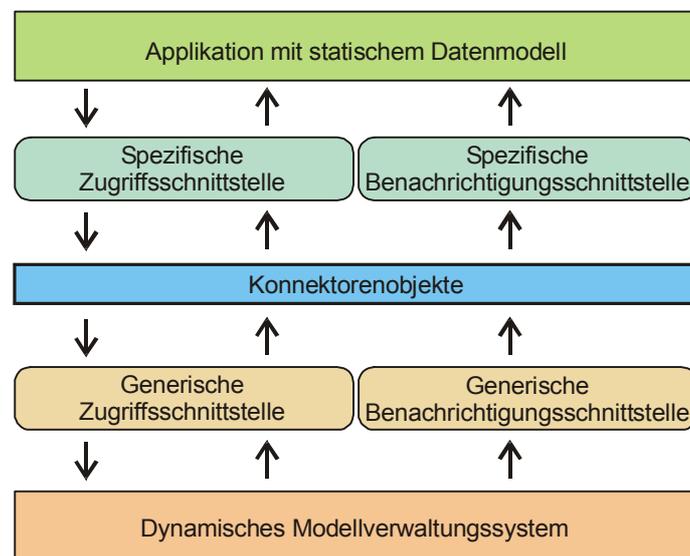


Abbildung 3: Funktionsweise der Konnektoren

Spezifische Aufrufe zum Zugriff auf eine Facette oder Relation werden von den Konnektoren in generische Aufrufe an den Referenzen der Modellobjekte des dynamischen Modellverwaltungssystems und generische Benachrichtigungen vom MVS in spezifische Benachrichtigungen für die eigenen Beobachter übersetzt (Abb. 3).

Konnektoren ermöglichen dem Anwendungsprogrammierer, statt der abstrakten Begriffe der Metaschnittstelle konkrete Termini der Anwendungsdomäne zu verwenden. Dies führt zu einer Verringerung der Komplexität der Programme, sowie zu einer stärkeren Formalisierung, die logische Fehler bereits zur Kompilierzeit erkennen und vermeiden hilft.

Die clientseitige Implementierung der Konnektorenklassen bedeutet die völlige Transparenz der Verteilungsfunktionalität und bietet zudem den Vorteil der direkten Umsetzbarkeit des Modell- Beobachter- Musters.

Dank der semantischen Äquivalenz der AKO- Schnittstelle zu Konstrukten einer objektorientierten Programmiersprache ist es möglich, eine formale Abbildung auf die beim Client eingesetzte Programmiersprache zu definieren. Diese kann zum Zeitpunkt der initialen Einrichtung des Gesamtsystems oder nach größeren Modifikationen am Domänenmodell durch einen Generator abgearbeitet werden, um aus einem im Modellverwaltungssystem

vorgehaltenen Domänenmodell Konnektorenklassen in einer spezifischen Programmiersprache zu generieren.

4 Eine Systemarchitektur für integrierte Planungsunterstützungssysteme

Bei der Modellierung von Bauwerken als auch in der allgemeinen Produktmodellierung sind zentrale, dezentrale und hybride Ansätze denkbar. Der hybride Ansatz versucht, die Vorteile der beiden anderen Varianten zu verbinden und zeichnet sich durch das Vorkommen selbständiger, domänenabhängiger Partialmodelle und eine diese Partialmodelle integrierende Komponente aus. Die konkreten Spezifika einer hybriden Modellarchitektur sind von der konkreten Entwurfsumgebung abhängig und haben sowohl auf das Modellverwaltungssystem als auch auf die gesamte Systemarchitektur Auswirkungen.

Im beschriebenen Forschungsprojekt wird der logische Gesamtinformationsbestand zur Repräsentation des Bauwerks in getrennt verwaltete Domänenmodelle unterteilt. Diese werden jedoch auf semantischer Ebene durch Verknüpfungen und ein Navigationsmodell sowie technisch durch einen zentralen Projektinformationsserver zusammenhängend gehalten. Die verwendete Systemarchitektur besteht aus den Grundkomponenten zentraler Projektserver, Domänenmodellserver und Domänenclient (Abb. 4).

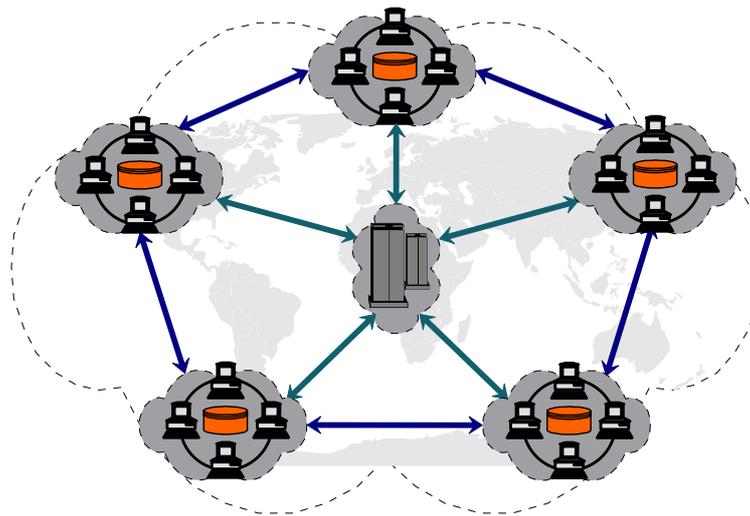


Abbildung 4: Prinzipielle Struktur der Systemarchitektur

Der zentrale Projektserver ist für die Verwaltung projektglobaler Informationen zuständig. Dies betrifft einerseits notwendige Informationen über die technische Infrastruktur des Bauprojekts sowie eine Komponente zum Management der Authentisierungs- und Autorisierungsdaten, andererseits werden in einem zentralen Navigationsmodell Verweise auf spezielle Informationsbestände, die bestimmten Sichten zugeordnet sind, verwaltet.

Die Domänenmodellserver verwalten Klassen- und Objektinformationen der Partialmodelle, also Domänenwissen und Projektinformationen. Daher ist dessen wesentlichste Komponente der oben erwähnte Modellverwaltungskern. Ferner müssen Vorkehrungen zur Sicherung der physikalischen Datenintegrität der Domänenmodellinformationen sowie zur Vermeidung von Problemen infolge synchronen Modellzugriffs getroffen werden; ebenso ist ein lokales Modul zur Verwaltung von Zugriffsrechten erforderlich [6].

Ein Domänenclient befindet sich auf allen Maschinen von Bearbeitern von Entwurfsaufgaben im Revitalisierungsprojekt, um die Funktionalität der Kommunikation mit dem Domänenmodellserver bereitzustellen. Dessen Hauptaufgabe ist es, als Schnittstelle zwischen

Modellverwaltung und Fachapplikation zu fungieren, um den Entwerfenden über Applikationen, die für seine Aufgaben geeignet sind, eine Interaktion mit dem gemeinsamen, verteilten Bauwerksinformationsbestand zu ermöglichen.

5 Ausblick

Der vorliegende Beitrag analysiert Anforderungen an die Gestaltung sowie das Management von digitalen Bauwerksmodellen für Vorhaben der Revitalisierung. Weiterhin werden Ansätze des Designs und der Implementierung von Modellverwaltungskernen und Benachrichtigungsmodulen zur Unterstützung synchroner Kooperation in Einzeldisziplinen beschrieben, ferner wird eine Systemarchitektur vorgestellt, welche die spezifischen Belange von Entwurfstätigkeiten berücksichtigt.

Zur Weiterentwicklung des Systems ist die Nutzung der beschriebenen Mechanismen für Zwecke der Konsistenzsicherung der Informationen in einzelnen Domänenmodellen geplant, da die Integration von konsistenzsichernden Regeln in die Domänenmodelle und diese Regeln überwachende Module in die Modellverwaltung eine Interaktion zwischen den betroffenen Komponenten der Gesamtsystemarchitektur erfordert. Der Grund dafür ist, dass Verletzungen der Konsistenzregeln durch das Modellverwaltungssystem erkannt werden können, aber eine Behebung dieses Zustands nur im Zusammenwirken mit den Fachapplikationen und im allgemeinen durch Interaktion mit dem Fachplaner erfolgen kann.

Das MVS ist dabei für eine constraintbasierte Prüfung innerhalb eines Domänenmodells verantwortlich. Dabei werden einerseits die Informationen des Modells selbst zur Prüfung herangezogen, beispielsweise kann die Einhaltung der korrekten Kardinalitäten an Assoziationen und Aggregationen überwacht werden. Andererseits können Regeln durch Nutzung eines Subsets der Object Constraint Language (OCL) der UML in die Modelle integriert werden, so dass für Attributwerte beispielsweise Minima, Maxima und einfache numerische Abhängigkeiten von anderen Attributwertbelegungen geprüft werden können.

Literatur

1. Hübler, R.; Steinmann, F.: Knowledge-based computer assistance for functionality and shape oriented building design. Proceedings of the 6th ICCCB, Berlin, July 1995, P. Pahl (Ed.), Balkema Rotterdam, 1995
2. Hauschild, T.; Hübler, R.: Aspekte der verteilten Bauwerksmodellierung in kooperativen Entwurfsumgebungen auf Basis dynamischer Objektstrukturen. Proceedings der Internationalen Konferenz für Anwendungen der Informatik und Mathematik in den Ingenieurwissenschaften IKM 2000, Weimar, 2000
3. The Object Management Group (OMG): The Common Object Request Broker Architecture and Specification. OMG Document 98-02-33, Revision 2.2 Framingham, MA, 1998
4. Kolbe, P.; Ranglack, D.; Steinmann, F.: Eine Schnittstelle für dynamische Objektstrukturen für Entwurfsanwendungen. Internationale Konferenz für Anwendungen der Informatik und Mathematik in den Ingenieurwissenschaften IKM '97: Berichte, Bauhaus-Universität Weimar, 1997
5. The Object Management Group: Notification Service Specification, Version 1.0. OMG Document 00-06-20, Framingham MA, 2000
6. Hauschild, T.; Hübler, R.; Schleinitz, M.: Unterstützung von Gruppenarbeit im Bauwerksentwurf auf Basis eines dynamischen objektorientierten Bauwerks-Modelliersystems. GI-Fachtagung CAD 2000: Proceedings, Gesellschaft für Informatik e.V., 2000