Blickwinkelgesteuerte Transformation von Bauhaus-Graphen nach UML

Julia Wolff, Andreas Winter

Universität Koblenz-Landau, Institut für Softwaretechnik, D-56016 Koblenz

(julian|winter)@uni-koblenz.de

6. Mai 2005

Zusammenfassung

Die Kopplung des Architektur-Rekonstruktions-Werkzeugs Bauhaus mit UML-Werkzeugen ermöglicht die Architekturdarstellung durch UML-Modelle. Es wird gezeigt, wie ausgehend von Architektursichten Bauhaus-Graphen in UML transformiert und visualisiert werden.

1 Motivation

Das Reverse Engineering-Werkzeug Bauhaus unterstützt die Rekonstruktion der Softwarearchitektur bestehender Softwaresysteme. Verschiedene Analysetechniken stehen für die Programmiersprachen Java, C, C++ und Ada zur Verfügung und ermöglichen das Auffinden von Komponenten und deren Beziehungen. Das Ergebnis wird als Graph visualisiert und kann mit Hilfe des enthaltenen Editors bearbeitet werden. [1]

Um eine möglichst umfassende Betrachtung des Systems widerzugeben, haben sich für die Beschreibung von Softwarearchitekturen verschiedene Blickwinkel (viewpoints) etabliert [2, 3, 4]. In dieser Arbeit wird untersucht, welche Blickwinkel aus Bauhaus extrahiert und durch UML-Modelle visualisiert und weiterverarbeitet werden können. Das Vorgehen wird an einem Beispiel verdeutlicht.

2 Bauhaus

Zur Analyse eines Systems mit Bauhaus werden aus dem Quellcode ableitbare Elemente wie z.B. Klassen, Pakete, Attribute, Methoden, Aufrufbeziehungen und Spezialisierungen als Knoten und Kanten im sogenannten Resource Flow Graphen (RFG) gespeichert. Sie bilden die Grundlage für weitere Analysen. In einem semi-automatischen Vorgang können Hypothesen über die möglichen Komponenten aufgestellt und überprüft werden, bis die Rekonstruktion der Architektur abgeschlossen ist. Durch Definition von *Cluster*-Knoten, denen vorhandene Knoten zugeordnet werden, können Subsysteme rekonstruiert werden. Desweiteren können in Bauhaus durch die Auswahl von Knoten- und Kantentypen Sichten (views) erstellt werden, die einen Ausschnitt des RFG zeigen und so eine zielgerichtete Reduktion der Architekturdarstellung ermöglichen. [1]

Aus den vorhandenen Daten und den rekonstruierten Informationen können beispielsweise Sichten nach dem Modulblickwinkel (module viewpoint) von Clements et al. [4] erstellt werden.

3 Blickwinkel

Ein Blickwinkel ist eine Spezifikation für die Erstellung einer Sicht [2]. Er legt fest, welche Elemente und Beziehungen verwendet werden dürfen, um eine bestimmte Perspektive der Architektur zu beschreiben [4]. Unter dem Begriff *Modulblickwinkel* werden mehrere atomare Blickwinkel definiert, die einen Teilaspekt der statischen Sicht auf das System repräsentieren: Dekompositionsblickwinkel, Benutzungsblickwinkel, Spezialisierungsblickwinkel und Schichtenblickwinkel.

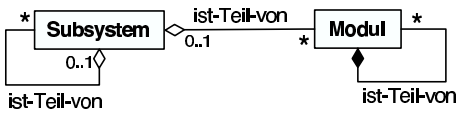

Abbildung 1: Meta-Modell für den Dekompositionsblickwinkel

Beispielsweise zeigt der *Dekompositionsblickwinkel* die Zerlegung des Systems in *Subsysteme* und *Module* durch die Berücksichtigung von *ist-Teil-von*-Beziehungen, vgl. Abbildung 1.

Ist-Teil-von-Beziehungen lassen sich in Bauhaus-Graphen in *Declared_In*- und *Enclosing*-Kanten wieder finden. Abbildung 2 zeigt einen Ausschnitt der Ausgangsdaten (Base View) des Produktlinienmanagement-Systems Kobold [5] mit der Visualisierung von ist-Teil-Von-Kanten als „Boxen“. Die manuell erzeugten *Cluster*-Knoten und die Abbildung der Bestandteile des Systems auf diese Knoten liegen in zwei weiteren Sichten vor.

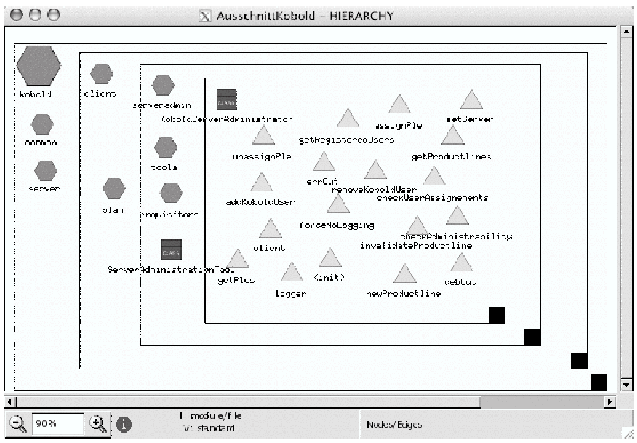

Abbildung 2: Dekompositionsblickwinkel in Bauhaus

Um eine Sicht entsprechend des Dekompositionsblickwinkels zu erzeugen, können die in Bauhaus rekonstruierten *Cluster*-Knoten als Subsysteme betrachten und alle anderen Knoten wie *Class*, *Interface* und *Package* als darin enthaltene Module angesehen werden.
Andere Blickwinkel, wie z.B. der Modulblickwinkel von Hofmeister et al. [3], setzen sich ebenfalls aus mehreren atomaren Blickwinkeln zusammen und erfordern zur Visualisierung die Kombination von mehreren Aspekten.

4 Transformation

Die für die Darstellung des Dekompositionsblickwinkels notwendigen Informationen liegen in Bauhaus in mehreren Sichten vor und müssen in einer Transformation zusammengeführt werden. Die Ausgangsdaten können, gesteuert durch einen Blickwinkel, auf die für einen Blickwinkel relevanten Daten reduziert werden. Die einzelnen Sichten werden als GXL-Graphen (GXL: Graph eXchange Format) gespeichert. GXL ist ein XML-basiertes Austauschformat für Graphen, das eine standardisierte Notationen für den Austausch von Instanzen (Graphen) und deren Schemata zur Verfügung stellt [6]. Ein Metamodell für RFG liegt in Bauhaus implizit vor und kann als GXL-Schema umgesetzt werden [7].
Die GXL-Graphen werden mit Hilfe von XSLT (eXtensible Stylesheet Language Transformation) in ein XMI-Dokument überführt. Diese Datei enthält die textuelle Beschreibung eines UML-Modells nach XMI-Standard [8], das von einem UML-Werkzeug eingelesen und visualisiert werden kann. Das XMI-Dokument ist nach einer UML 2.0 Dokument Spezifikation aufgebaut, die im verwendeten Werkzeug implizit vorliegt. Abbildung 3 zeigt die Transformationsschritte der Sichten in Bauhaus zum UML-Modell.

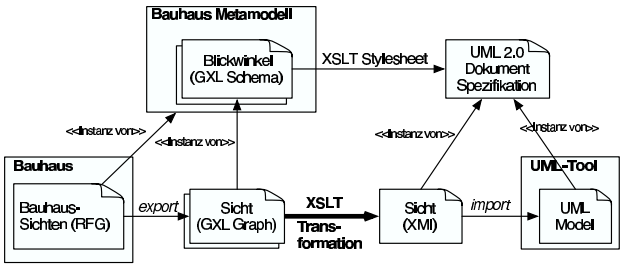

Abbildung 3: Ablauf der Transformation

Für jeden Blickwinkel müssen die Abbildungen von den Elementen des GXL-Graphen auf die Elemente des XMI-Schemas festgelegt werden. Die Visualisierung der XMI-Elemente ist durch den UML Standard [9] festgelelegt und wird durch das UML-Werkzeug realisiert. Der Dekompositionsblickwinkel dient als Beispiel, um die Machbarkeit zu zeigen; für andere Blickwinkel werden weitere Transformationen erstellt und damit entsprechende UML-Modelle erzeugt. Möglich ist außerdem die Darstellung des Benutzungs- und Spezialisierungsblickwinkels, sowie Kombinationen dieser Blickwinkel [7].

5 UML-Werkzeug: Software Modeler

Die verfügbaren UML-Werkzeuge unterscheiden sich nicht nur hinsichtlich der zur Verfügung gestellten UML-Version, sondern auch in der Implementierung des XMI-Standards. In diesem Beispiel wird der Rational Software Modeler v6.0 von IBM [10] für die Visualisierung verwendet. Es werden UML-Diagramme in der Version 2.0 [9] unterstützt und in der XMI Version 2.0 importiert und exportiert. Abbildung 4 zeigt die Darstellung der ausgewählten Sicht im Rational Software Modeler.

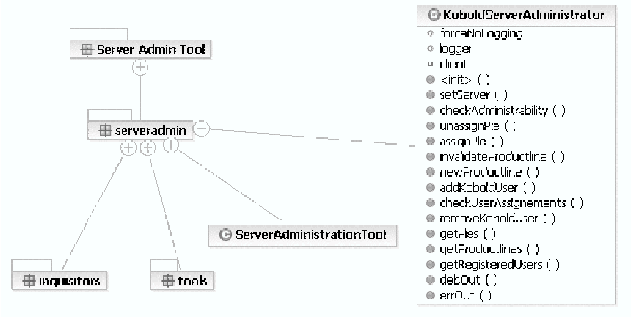

Abbildung 4: Visualisierung des Ausschnitts im Software Modeler

6 Fazit

Mit der Anbindung von Bauhaus an ein UML-Werkzeug wird ein Beitrag zur Werkzeuginteroperabilität geleistet. Durch die Auswahl von einzelnen Blickwinkeln kann die Komplexität der Architekturdarstellung reduziert werden und so das Verständnis von Alt-Systemen erleichtern. Die Transformationen werden auf der Ebene der Instanzen von Graphen und UML-Modellen durchgeführt. Wünschenswert für die Zukunft wäre eine Beschreibungsmöglichkeit für die Transformationen auf Metaebene, d.h. zwischen GXL-Schema und UML-Spezifikation in Abbildung 3.

Literatur

[1] KOSCHKE, Rainer ; EISENBARTH, Thomas ; BELLON, Stefan ; SIMON, Daniel: *Tour de Bauhaus Version 5.0.0*. Stuttgart: Universität Stuttgart, September 2004. http://www.bauhaus-tec.com

[2] IEEE Computer Society: *Standard 1471-2000, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*. September 2000

[3] HOFMEISTER, C. ; NORD, R. ; SONI, D.: *Applied Software Architecture*. Reading : Addison-Wesley Logman, Inc., 2000

[4] CLEMENTS, P. ; BACHMANN, F. ; BASS, L. ; GARLAN, D. ; IVERS, J. ; LITTLE, R. ; NORD, R. ; STAFFORD, J.: *Documenting Software Architecture: Views and Beyond*. Boston : Pearson, 2003

[5] UNI STUTTGART, Studenten an d.: *Ein Produktlinien-Manager: Kobold*. http://kobold.berlios.de/index.html. Version: 2004

[6] HOLT, Richard C. ; SCHÜRR, Andy ; ELLIOT-SIM, Susan ; WINTER, Andreas: GXL: A Graph-Based Standard Exchange Format for Reengineering. In: *SOCP: Science of Computer Programming* (2005, to appear)

[7] WOLFF, Julia: *OBI: Objekt-orientierte Bauhaus Integration*, Universität Koblenz-Landau, erscheint als Diplomarbeit, 2005

[8] OMG: Object Management Group: *XML Metadata Interchange (XMI) Specification, Version 2.0*. http://www.omg.org/cgi-bin/apps/doc?formal/03-05-02.pdf. Version: Mai 2003

[9] OMG: Object Management Group: *UML 2.0 Superstructure Specification*. http://www.omg.org/cgi-bin/doc?ptc/2004-10-02. Version: 2004

[10] *IBM Software - Rational Software Modeler - Product Overview*. http://www-306.ibm.com/software/awdtools/modeler/swmodeler/index.html