

Präzisierung des UML-Metamodells durch ein semantisches Objektmodell

Beitrag für den 7-ten GROOM-Workshop

”UML – Erweiterungen (Profile) und Konzepte der Metamodellierung”

Holger Giese, Jörg Graf und Guido Wirtz
Institut für Informatik, Gruppe für verteilte Systeme
Westfälische Wilhelms-Universität Münster
<http://wwwmath.uni-muenster.de/cs/u/versys/>

Eine der wesentlichen Fragen bei der Betrachtung der UML Erweiterung durch Profiles [2] und im allg. der Metamodellierung im Zusammenhang mit der UML [1] ist, ob die UML eine Semantik oder mehrere besitzt oder besitzen sollte (siehe [3]). Obwohl zahlreiche Bereiche und auch die verschiedenen Phasen von Analyse über Design bis Implementierung teilw. das Modell aus verschiedenem Blickwinkel betrachten, sollte man die Forderung nach Konsistenz und Nachvollziehbarkeit der Übergänge zwischen verschiedenen Modellen (engl. traceability) nicht a priori aufgeben. Vielmehr verlangt diese Frage nach einer differenzierten Betrachtung.

Muß es verschiedene Semantiken für Analyse und Design Modelle geben? Ein Sequenzdiagramm kann bei der Analyse die Bedeutung haben, daß eine Menge gewünschter Abläufe beschrieben wird. Dagegen ordnet man ihm im Designmodell eine eher konstruktive Sicht zu und betrachtet das beschriebene Verhalten häufig als komplette Spezifikation z.B. für eine Methode. Dieser Unterschied in der Bedeutung und Interpretation der Diagramme sollte allerdings besser explizit anstatt implizit erfolgen in dem man unterscheidet, ob eine Menge gewünschter Fälle oder auch nicht gewünschter Fälle oder z.B. eine Methode komplett in ihrem möglichen externen Verhalten beschrieben wird. Dies macht indes keine unterschiedliche Semantiken notwendig. Ein andere Frage ist inwieweit Erweiterungen und spezifisch auf einen Bereich zugeschnittene Teilmodelle der UML zu verschiedenen Semantiken führen. Neben den zahlreichen Bestrebungen die UML zu erweitern, fehlt es leider an einer geeigneten Architektur, die diese Erweiterung gliedert und somit die Betrachtungen verschiedener Erweiterungstypen und die Frage nach notwendiger Konsistenz oder verschiedener Semantik erlaubt. Eine solche Gliederung könnte sich z.B. an der ”Object Management Architecture” der OMG [4] orientieren und würde demnach die folgenden Elemente enthalten:

- Allgemeine Erweiterungen: Sie stellen die fundamentalen Aspekte bereit. In der UML könnte man dies für Erweiterungen der abstrakten Modellsicht um allgemeine Aspekte wie Persistenz, Zeit, Transaktionen,

Verteilung, Nebenläufigkeit, ... nutzen (siehe Scheduling, Performance and Time RFP [5]).

- Vertikale Erweiterungen: Sie stellen bereichsspezifische Dienste zur Verfügung. Dieses kann in der UML mit Ansätzen wie dem EDOC-Profile (siehe [6]) verglichen werden, die die notwendigen Standardobjekte und Strukturen für einen spezifischen Bereich definieren.
- Horizontale Erweiterungen: Sie stellen allg. gebräuchliche Sichten zur Verfügung. Im Fall der UML könnte man so die verschiedenen Programmiersprachen wie Java, C++, ... oder allgemeine Plattformen wie CORBA, DCOM oder EJB integrieren (siehe CORBA Profile [7]).

Für die einzelnen Arten von Erweiterungen ergeben sich dann recht verschiedene Konsistenzfragen. So müssen die fundamentalen Aspekte wie z.B. Persistenz und Zeit miteinander harmonisieren, ansonsten sind keine Systeme, die bei Analyse und Design beide Aspekte berücksichtigen wollen, möglich. Dagegen können horizontale Erweiterungen wie DCOM oder EJB durchaus unverträglich sein. Die Kopplung entsprechender Modelle könnte dann z.B. nur für eine Untermenge an Bestandteilen (z.B. Interfaces) unterstützt werden. Für vertikale Erweiterungen ist es abhängig von dem Bedarf, ob Konsistenz oder sogar Kombinierbarkeit unterstützt werden sollte. Des weiteren stellt sich für eine Erweiterung die Frage welche Semantikaspekte diese umfaßt. Hierbei ergeben sich unserer Meinung nach die folgenden drei Grade:

- Die Erweiterungen führen nur syntaktische Ergänzungen durch. D.h., neu eingeführte Stereotypen und Metamodell Klassen verfügen nicht über OCL Restriktionen (siehe [8]).
- Neben der syntaktischen Erweiterung werden auch OCL Restriktionen zur genaueren statischen Semantik-Beschreibung verwendet.

- Die Erweiterungen sind teilw. auch in ihrem Verhalten zumindest generisch beschrieben.

Im Fall der rein syntaktischen Erweiterungen können verschiedene Profiles nur durch Namenskonflikte zu Problemen führen. D.h. wir können diese durch entsprechende Sichtbarkeitsregeln und Verwendung von Namensräumen teilw. umgehen. Betrachten wir dagegen auch OCL Restriktionen, so kann es zu Widersprüchen bei der Komposition von OCL Regeln kommen, die eine Erfüllbarkeit d.h. die mögliche Existenz einer Implementierung ausschließen. Bei Erweiterungen, denen z.B. durch ein Sequenzdiagramm ein generisches Verhaltensmuster zugeordnet wurde, ist durch die Beschränkung des UML-Metamodells auf Strukturen keine Aussage möglich. Somit muß man konstatieren, daß für solche Erweiterungen nicht einmal ein Konsistenzbegriff möglich ist.

Dieses Problem wollen wir durch das Erweitern des UML Instanzmodells und in Bezug setzen der Verhaltensmodellierungsbestandteile zu diesem Instanzmodell beheben. Auch die Verwendung von OCL Restriktionen kann von dieser Vorgehensweise profitieren insofern die Verhaltensbeschreibungen in Form von Diagrammen bzgl. ihrer Konsistenz mit den OCL Restriktionen allgemein testbar werden. Erst dann lassen sich die Probleme von Erweiterungen, die Verhalten umfassen und deren Konsistenz, durch die eindeutige Abbildung ihrer Konstrukte in miteinander vergleichbare Terme dieses Modells angehen. Dieses Objektmodell soll keine vollständige Semantik für die UML definieren, sondern eher dazu dienen, wesentliche Anforderungen für solche Basismodelle zu ergründen. Der Schwerpunkt des Modells liegt in der Reduktion der UML auf einen kleinen Sprachkern mit wenigen Konstrukten (Schnittstellen, Implementierungsklassen und Beziehungen, usw.), für die auf einer abstrakten Instanzebene eine präzise und konsistente Laufzeitsemantik für jede Verhaltensspezifikation definiert werden kann. Dabei wird im wesentlichen auf die Verhaltensdynamik von Objektstrukturen und den darauf operierenden Methoden mit ihren Synchronisations- und Interaktionseffekten fokussiert. Bestehende Synchronisationsbedingungen zwischen Objekten werden im Modell durch abstrakte Zustände und Transitionen repräsentiert. Für die fundamentalen Erweiterungen, auf denen horizontale oder vertikale Erweiterungen erst aufbauen könnten, ist die Kombinierbarkeit wesentlich. Ebenso muß jede fundamentale Erweiterung eine Interpretation ihrer Zustände und Übergänge ebenso im Basisobjektmodell erlauben, so daß die Frage, ob ein Modell mit Zeit einem nicht zeitbehafteten Sequenzdiagramm genügt, prinzipiell durch Betrachtung des reduzierten Basisobjektmodells möglich wird.

Abschließend läßt sich resümieren, daß eine Differenzierung und somit Inkonsistenzen zwischen verschiedenen Profiles bzgl. der Semantik durchaus sinnvoll sein können, wenn die resultierenden Unverträglichkeiten in der Praxis keine Probleme ergeben. Dagegen halten wir die teilw. geäußerte Meinung, entsprechende Erweiterungen um funda-

mentale Aspekte wie Zeit oder Persistenz in solchen inkompatiblen Profiles vorzunehmen, für gefährlich. Die Standardisierung durch die UML beinhaltet nicht zuletzt die Chance eine bereichsübergreifende universelle Modellierungssprache zu erhalten, die für den Informationsaustausch notwendig ist. Wird nicht erreicht, daß zumindest eine Kernsemantik allen Profiles gemein ist, so stellt sich die Frage wo zu ein gemeinsamer Ausgangspunkt und dessen Beschränkungen überhaupt notwendig war. Auch muß bemerkt werden, daß eine solche Entwicklung ein ansonsten mögliches plattformunabhängiges Design auch für erweiterte Aspekte wie Zeit unmöglich macht bzw. die schrittweise Überführung in ein spezielles Profile aufgrund der verschiedenen Semantiken zumindest zu einem fragwürdigen Unterfangen werden läßt. Mit dem vorgestellten Instanzmodell und dessen schrittweiser und behutsamer Erweiterung um Aspekte wie Zeit und Persistenz ist die unseres Erachtens notwendige konsistente Kernsemantik für die UML möglich. Des weiteren kann die resultierende operationale Sicht wenn möglich zur Simulation und somit interaktiven Validierung eines Designs bereits auf Modellebene verwendet werden.

Literatur

- [1] Object Management Group. *OMG Unified Modelling Language 1.3*, June 1999. OMG document ad/99-06-08.
- [2] Philippe Desfray. White Paper on the Profile Mechanism. Techreport, Object Management Group, 1999. OMG document ad/99-04-07.
- [3] Steve Cook, Anneke Kleppe, Richard Mitchell, Bernhard Rumpe, Jos Warmer, and Alan Wills. Q1: should UMI have a single semantics? In *ECOOP'99 Workshop: UML Semantics FAQ*, Monday 14th June, 1999, Lisabon, Portugal, June 1999.
- [4] Object Management Group. *A Discussion of the Object Management Architecture*, January 1997.
- [5] Object Management Group. *UML Profile for Scheduling, Performance, and Time RFP*, 1999. OMG Document: ad/99-03-13.
- [6] Object Management Group. *UML Profile for Enterprise Distributed Object Computing RFP*, March 1999. OMG Document: ad/99-03-10.
- [7] Object Management Group. *UML Profile for CORBA*, October 1999. Joint Initial Submission Version 1.0 July 30, 1999 OMG Document ad/99-08-01.
- [8] Jos Warmer and Anneke Kleppe. *The Object Constraint Language, Precise Modeling with UML*. Addison-Wesley, 1999.