

**Bericht über das zweite Treffen  
des  
Arbeitskreises GROOM  
(Grundlagen der OO-  
Modellierung)  
der GI-Fachgruppe 2.1.9  
mit Schwerpunkt  
Syntax- und Semantikdefinition  
objektorientierter Methoden**

**am 7.-8. Februar 1997 an der TU München**

**Bernhard Rumpe Institut für Informatik TU München**

**[rumpe@informatik.tu-muenchen.de](mailto:rumpe@informatik.tu-muenchen.de)**

**Andy Schürr Lehrstuhl für Informatik III RWTH Aachen**

**[andy@i3.informatik.rwth-aachen.de](mailto:andy@i3.informatik.rwth-aachen.de)**

Das zweite Treffen des Arbeitskreises GROOM (Grundlagen der objekt-orientierten Modellierung) fand im direkten Anschluß an das vierte Treffen der GI-Fachgruppe 2.1.9 OOSE (Objekt-Orientierte Software-Entwicklung) an der TU München statt. An dieser Stelle möchten wir uns bei Herrn Broy und seinen Mitarbeitern für die lokale Organisation des Treffens, die Bereitstellung eines Seminarraums mit fast historischem Ambiente und den kostenlosen Ausschank von Kaffee und anderen Getränken ganz herzlich bedanken.

Das Treffen war mit über 40 Teilnehmern aus den Niederlanden, Schweden, der Schweiz und natürlich Deutschland außerordentlich gut besucht. In seinem Begrüßungsvortrag ging Herr Broy nicht nur auf die dringende Notwendigkeit der Präzisierung und Fortentwicklung objektorientierter Notationen ein, sondern schilderte

auch die Vorzüge Münchens und seines guten Wetters für Spaziergänge sowie Besuche von Museen und Biergärten. Direkt im Anschluß daran begann das eigentliche fachliche Programm mit 9 Vorträgen von jeweils 30 Minuten Länge, verteilt auf drei Sitzungen. Die sich daraus ableitenden lebhaften Diskussionen zu den angeschnittenen Themen (siehe Zusammenfassungen der Vorträge) fanden ihren Höhepunkt in der abschließenden Gesprächsrunde zum aktuellen Thema ``Unified Modelling Language". Sie wurde von Herrn Weber (Fachhochschule Darmstadt) geleitet. Als erstes berichtete Herr Wiegert (SAP, Walldorf) über die weitere Vorgehensweise bei der Standardisierung von OO-Notationen durch die OMG (Object Management Group). Bei ihr wurde Anfang Januar die UML von der Rational Software Corporation als ein Standardisierungsvorschlag eingereicht. Im Laufe der Diskussion zu Themen wie ``Stereotypen", ``Semantik der Aggregation", etc. wurde (zum wiederholten Male) offenbar, wieviele Konstrukte der UML (und anderer OO-Notationen) immer noch eine unklare Semantikdefinition besitzen. Einige Probleme der Teilnehmer des Treffens mit UML konnten im Verlaufe der Diskussion geklärt werden, aber viele andere Punkte - wie etwa die Verfeinerung von Packages - mußten vertagt werden.

So wurde noch einmal der Sinn des Arbeitskreises GROOM deutlich, der sich gerade mit der präzisen (formalen) Definition von OO-Notationen, ihrer Bewertung und Vermittlung sowie ihrer Weiterentwicklung befaßt.

Danach war das fachliche Programm für diesen Tag beendet und ein Besuch des Löwenbräukellers angesagt. Dort stand weniger die Semantikdefinition von UML als vielmehr das Problem der Übersetzung einer bayerischen Speisekarte (z. B. ``Antn" = ``Ente") im Vordergrund.

Am nächsten Morgen wurde das Treffen mit drei weiteren Vorträgen zu je 30 Minuten fortgesetzt. Den Abschluß bildete eine einstündige Diskussion künftiger Aktivitäten des Arbeitskreises GROOM. Sie läßt sich wie folgt zusammenfassen:

1. Das nächste Treffen des Arbeitskreises wird von Herrn Hildebrand (INTEGRATA Unternehmensberatung GmbH) und Herrn Wiegert (SAP AG) organisiert. Es wird voraussichtlich Anfang Oktober

entweder bei INTEGRATA in Düsseldorf oder bei SAP in Walldorf oder beim dpunkt-Verlag in Heidelberg stattfinden.

2. Eine zweitägige Länge von Arbeitskreistreffen fand überwiegend Zustimmung bei den Mitgliedern. Angeregt wurde allerdings, für jeden Tag eines künftigen Treffens einen eigenen inhaltlichen Schwerpunkt zu setzen, so daß auch eintägige Teilnahme sinnvoll möglich ist.

3. Sehr kontrovers wurde der Vorschlag diskutiert, ein künftiges Treffen des Arbeitskreises mit einem eintägigen Tutorial zu Schwerpunktthemen des Arbeitskreises zu verbinden. Hiermit ließe sich ggf. der Anspruch des Arbeitskreises verwirklichen, in punkto Lehre/Vermittlung von OO-Konzepten gemeinsam aktiv zu werden.

4. Wie beim letzten Treffen wurde auch diesmal wieder angeregt, zu einzelnen Teilthemen WWW-Seiten aufzubauen. An dieser Stelle sei darauf hingewiesen, daß zum Themenbereich "OO-Datenbanken vs. relationale Datenbanken" bereits sehr informative WWW-Seiten von Frau Demuth (TU Dresden) aufgebaut wurden (unter der Adresse <http://www.inf.tu-dresden.de/TU/Informatik/ST2/ST/groom/>). Sie sind auch über die Hauptseite von GROOM <http://www-i3.informatik.rwth-aachen.de/groom/> erreichbar.

5. Angeregt und inzwischen von Herrn Börstler (Universität Umeå, Schweden) realisiert wurde ein www-basiertes GROOM-Anschlagbrett, auf dem Ankündigungen, Diskussionsbeiträge etc. "angeschlagen" werden können. Es ist unter der Adresse [http://www.cs.umu.se/~jubo/GROOM/wwwwhats\\_new/](http://www.cs.umu.se/~jubo/GROOM/wwwwhats_new/) erreichbar.

6. Des weiteren wurde eine eigene Mailingliste für GROOM vorgeschlagen und eingerichtet, über die GROOM-interne Diskussionen, Anfragen, etc. abgewickelt werden sollen. Über eine sinnvolle Kopplung der unter Punkt 5 erwähnten Anschlagtafel und der neu eingerichteten Mailingliste wird nachgedacht. Es sei an dieser Stelle explizit darauf hingewiesen, daß alle wichtigen Ankündigungen weiterhin über die Mailingliste der Fachgruppe OOSE abgewickelt werden (sollen). Somit kann jedes Mitglied von GROOM für sich entscheiden, ob es die Diskussionen in der arbeitskreisspezifischen Mailingliste mitverfolgen will. Das Versenden irrelevanter Emails über die neue Mailingliste wird (weitgehend) dadurch verhindert, daß nur Mitglieder der Mailingliste unzensuriert Email verschicken dürfen.

Die Mailingliste [gi\\_ose@wi.leidenuniv.nl](mailto:gi_ose@wi.leidenuniv.nl) der Fachgruppe 2.1.9 kann per email mit dem SUBJECT ``subscribe" an [gi\\_ose-request@wi.leidenuniv.nl](mailto:gi_ose-request@wi.leidenuniv.nl) bezogen werden. Die GROOM Mailingliste [groom@i3.informatik.rwth-aachen.de](mailto:groom@i3.informatik.rwth-aachen.de) kann per email an [majordomo@i3.informatik.rwth-aachen.de](mailto:majordomo@i3.informatik.rwth-aachen.de) mit dem BODY ``subscribe groom" bezogen werden.

7. Angeregt wurde des weiteren, daß Mitglieder von GROOM, die an UML interessiert sind, sich bereits vor dem nächsten Treffen mit diesem Thema auseinandersetzen. Dabei wurde insbesondere von Herrn Oestereich (debis Systemhaus GmbH, Hamburg) folgende Vorgehensweise vorgeschlagen: Zunächst Auflistung der Bestandteile von UML, dann Priorisierung der Bestandteile bezüglich ihrer Wichtigkeit (bzw. des noch notwendigen Klärungsbedarfes), und anschließend Diskussion der identifizierten besonders strittigen/wichtigen Punkte in UML. Als ersten Beitrag hierzu stellt Herr Oestereich ein ``UML-Online-Buch" im WWW unter [http://home.t-online.de/home/Bernd.Oestereich/\\_start.htm](http://home.t-online.de/home/Bernd.Oestereich/_start.htm) zur Verfügung.

8. Zudem wurde es als wünschenswert angesehen, die bereits im WWW verfügbaren Zusammenfassungen der Vorträge (<http://www4.informatik.tu-muenchen.de/~rumpe/groom/>) durch weitere Literaturhinweise und html-Links zu ergänzen. Zudem sollen/werden in naher Zukunft WWW-Seiten aufgebaut, auf denen die einschlägigen Veröffentlichungen der einzelnen Mitglieder des Arbeitskreises aufgeführt sind.

9. Ein bereits vorhandenes OO-Glossar wird von Rainer Burkhardt (TU Ilmenau) demnächst in HTML-Form zur Verfügung gestellt. Eine Arbeitsgruppe von GROOM wird sich mit der Überarbeitung und vor allem Erweiterung dieses Glossars befassen.

10. Schließlich wurde eine Reihe weiterer Aktivitäten für sinnvoll erachtet, zu denen allerdings im großen Kreis keine konkreten Aktivitäten vereinbart wurden. Dazu zählt der Vergleich verschiedener Formalismen zur Definition von OO-Notationen anhand eines konkreten Fallbeispiels, der Austausch von Fallbeispielen und der verstärkte Erfahrungsaustausch zum Thema ``Lehre/Schulung von OOA/OOD".

Zusammenfassend läßt sich wohl zurecht behaupten, daß das Treffen des Arbeitskreises GROOM seinen Zweck - Kennenlernen der

Mitglieder, Identifizierung interessanter Themenbereiche, Austausch von Informationen, Vereinbarung inhaltlicher Zusammenarbeit in kleineren Gruppen - voll und ganz erfüllt. Auch gaben eigentlich alle Vorträge den gewünschten Anlaß zu oft lebhaften und manchmal kontroversen, aber immer spannenden Diskussionen.

## **Vorträge - Zusammenfassungen**

Die zu den letzten beiden Zusammenfassungen gehörenden Vorträge wurden bereits im Rahmen der Fachgruppensitzung 2.1.9 am 6.2.1997 gehalten.

### **Operation Execution und der Object Life Cycle**

**(Barbara Paech, Bernhard Rumpe, Technische Universität München)**

Dieser Vortrag untersucht das Zusammenspiel der Operationsbeschreibungen und der State Transition Diagrams (STD) in UML. Eine Transition im STD kann eine Operation triggern. Bei der Operationsausführung treten aber interne Objektzustände auf, die in den STDs nicht wiedergespiegelt sind. Dies gilt aufgrund von Objektrekursion (während der Ausführung einer Operation auf Objekt X wird ggf. über andere Objekte eine weitere Operation auf X getriggert) selbst bei sequentieller Ausführung. Wir untersuchen semantische Modelle für die Operationsausführung anhand der folgenden Kriterien:

- das Modell beschreibt die bei objektorientierten Programmen übliche sequentielle Operationsausführung, läßt sich aber gut an eine parallele Ausführung (und entsprechende Erweiterungen der STDs wie z.B. Priorität von Operationen) anpassen,
- STD-Zustände entsprechen Datenzuständen und nicht nur Kontrollzuständen,
- das Modell ist auf Erweiterungen von Operationsbeschreibungen und STDs anpaßbar, die die Unterscheidung von Interaktionsoperationen (können an vordefinierten Stellen durch andere Operationen unterbrochen werden) und Transaktionsoperationen (atomarer Zugriff auf den Zustandsraum des Objektes) unterstützen.

Die mathematische Fundierung der untersuchten Ansätze verwendet ein formales, objektorientiertes Systemmodell.

Die mathematische Fundierung der untersuchten Ansätze verwendet das von C. Klein, B. Rumpe, M. Broy vorgestellte objektorientierte Systemmodell: A stream-based mathematical model for distributed information processing systems - SysLab system model.

FMOODS'96, p. 323-338, 1996, Ed. E. Naijm and J. Stefani.

## **Vererbung von Verhaltensbeschreibungen**

**(Jürgen Ebert, Universität Koblenz-Landau, Gregor Engels, Reichsuniversität Leiden (NL))**

Das dynamische Verhalten eines Objekts wird in zahlreichen objektorientierten Modellierungsansätzen mit Hilfe von Zustandsdiagrammen beschrieben. Hierbei wird jeder Klasse ein Zustandsdiagramm zugeordnet, welches den Lebenszyklus eines Objekts dieser Klasse beschreibt. Hierbei stellt sich dann die Frage, in welcher Beziehung die Verhaltensbeschreibungen von Klassen und Unterklassen zueinander stehen. In keinem der existierenden Ansätze sind hierzu konkrete Bedingungen oder Konstruktionsregeln zu finden.

Der Vortrag untersucht diese Beziehung im Detail. Es wird erläutert, dass man zwischen zwei verschiedenen Beziehungen unterscheiden kann, die entweder das beobachtbare oder das aufrufbare Verhalten eines Objekts beschreiben. Für beide Formen werden Konstruktionsregeln erläutert, um aus der Verhaltensbeschreibung einer Klasse die Verhaltensbeschreibung für eine Unterklasse abzuleiten.

## **Unterscheidung synchroner und asynchroner Kommunikation in objektorientierten Analyse- und Entwurfsmethoden**

**(Pierre Metz, Wolfgang Weber, FH Darmstadt)**

Spätestens beim Versuch der Umsetzung eines objektorientierten Modells in ein ausführbares Programmsystem bemerken wir, daß

einige Sachverhalte im Modell nicht präzise genug dargestellt werden können.

Einer dieser Sachverhalte ist die Unterscheidung zwischen logisch synchroner und logisch asynchroner Kommunikation. Unter logisch synchroner Kommunikation verstehen wir, im Gegensatz zur logisch asynchronen Kommunikation, eine Kommunikation, bei der das sendende Objekt auf eine Antwort vom Empfängerobjekt wartet. Der Begriff logisch synchron/asynchron ist an dieser Stelle gewählt worden, um sich von der physisch synchronen/asynchronen Kommunikation zu unterscheiden, die dem Nachrichtenaustausch zwischen physisch konkurrierenden, evtl. parallel ablaufenden Prozessen entspricht. Beispiele für diese Kommunikation sind die Konzepte von CORBA und Konzepte der Interprozeßkommunikation. Zu der im "concurrent object message diagram" der UML (Version 0.8 und 0.9) angesprochenen Kommunikation ist nicht klar ersichtlich, ob es sich um eine logische oder physische Kommunikation handelt. Auch Shlaer/Mellor führen eine Begriffsdefinition zu synchron/asynchron ein, die sich allerdings nur auf den Datenzugriff bezieht und deshalb nicht ausreichend ist. Die von uns vorgeschlagene Definition schließt die Lücken der obigen Begriffsdefinition.

Der Vortrag verdeutlicht die Wichtigkeit der Unterscheidung von logisch synchroner und asynchroner Kommunikation. Er stellt anhand von Beispielen dar, inwieweit in der UML Konzepte bestehen, die zur Modellierung von logischer synchroner/asynchroner Kommunikation Verwendung finden können (z. B. "focus-of-control diagram") und an welchen Stellen die UML noch erweitert/präzisiert werden müßte (z. B. ist das "state diagram" geringfügig zu erweitern und im "object message diagram" ist das Konzept der logischen und physischen Kommunikation streng voneinander zu trennen).

Außerdem wird dargestellt, wie logisch synchrone und asynchrone Nachrichten in Einprozeßsystemen auf Methodenaufrufe abgebildet werden (bzw. in Mehrprozeßsystemen / verteilten Systemen auf physisch asynchrone Nachrichten / synchrone Methodenaufrufe). Als Ergebnis entstehen zusätzliche Methoden in den Klassen der "class diagrams" und es werden "concurrent object message diagrams" mit synchronen / asynchronen Nachrichten auf physischer Ebene abgeleitet. Das so entstandene Modell kann dann 1:1 in einen Programmrahmen umgesetzt werden.

# Das Verständnis des Relationsbegriffs und seine Konsequenzen für die objektorientierte Modellierung

(Oliver Wiegert, SAP Walldorf)

Im Laufe des Jahres 1997 will die OMG ein Metamodell standardisieren, das die Semantik von OOA- und OOD-Modellen verkörpert. Im Rahmen dieses Standardisierungsprozesses fließen voraussichtlich auch Aspekte der Methode nach Martin/Odell in UML ein. Beide Methoden unterscheiden sich heute jedoch noch essentiell in ihrem Verständnis des Relationsbegriffs.

Dieser Beitrag vergleicht mittels formaler Definitionen die beiden Relationsbegriffe. Hierbei sind Unterschiede auf der Exemplar-, auf der Typ- und auf der Meta-Ebene zu erkennen. Er stellt die Kerne der beiden Metamodelle einander gegenüber und veranschaulicht anhand konkreter Beispiele, welche Konsequenzen für die objektorientierte Modellierung mit der jeweiligen Semantik von Relationen verbunden sind.

## Dynamische Verhaltensbeschreibung in UML - Eine Einordnung von Konzepten

(Ruth Breu, Radu Grosu, Technische Universität München)

UML bietet eine Reihe von Konzepten, auf deren Basis das dynamische Verhalten eines Systems beschrieben werden soll. Die wichtigsten Begriffe können folgendermaßen kurz charakterisiert werden.

- *Uses Cases* beschreiben komplexe Transaktionen zwischen dem Anwender und dem System.
- *Operationen* bilden die Schnittstelle eines Objekts.
- *Ereignisse (events)* und *Nachrichten (messages)* sind die Grundelemente der Kommunikation zwischen Objekten.

Im gegenwärtigen Entwicklungsstand von UML sind diese Begriffe nicht integriert und ihre Rolle innerhalb der Beschreibungstechniken unklar.



In diesem Vortrag werden zunächst Beziehungen zwischen Use Cases, Operationen und Ereignissen bzw. Nachrichten herausgearbeitet. Aufbauend auf der dadurch erhaltenen integrierten Sichtweise wird der Einsatz der UML-Beschreibungstechniken bei der Verhaltensmodellierung eines Systems diskutiert. Im Blickfeld stehen dabei sowohl semantische Aspekte einer Systembeschreibung als auch Fragen der Entwurfsmethodik.

## **RECORD - Eine Übersicht**

**(Jürgen Börstler, Dept of Computing Science, 90187 Umeå, SWEDEN)**

Das RECORD Projekt (REquirements COllection, Reuse and Documentation) beschäftigt sich mit bedienfreundlichen Ansätzen zur Definition von (funktionalen) Anforderungen für Softwaresysteme, um Anwender/innen in den Softwareentwicklungsprozess zu "integrieren".

In RECORD werden Anforderungen mit Hilfe von Use Case basierten Formularen definiert. Jeder Use Case wird durch eine Folge von Aktionen zwischen System und Benutzer/in beschrieben. Die Notation ist informell. Mit Hilfe einer Textanalyse und einer Bibliothek wiederverwendbarer Komponenten wird ein initiales objektorientiertes Design erzeugt. Gleichzeitig werden für alle Use Cases, sowie das Design HTML Dokumente erzeugt. Diese Dokumente setzen Anforderungen und Designkomponenten miteinander in Beziehung und unterstützen so die Kommunikation von Anwender/innen und Entwickler/innen.

Ein erster Prototyp mit eingeschränkter Funktionalität ist verfügbar.

Mehr Information ist unter

<http://www.cs.umu.se/~jubo/RECORD.html> erhältlich.

## **Graphgrammatiken als universelle präzise Beschreibungsmethode für graphische Notationen in OOA/D**

**(Gabriele Taentzer, Technische Universität Berlin)**

Objektorientierte Entwicklungsmethoden leben von der graphischen Notation der verschiedensten Strukturen in einem Softwaresystem in

Form von Diagrammen. Es gibt Diagrammarten für Klassenstrukturen, Objektstrukturen, Interaktionen, Zustandsübergänge, etc., die sich in verschiedenen Methoden mehr oder weniger unterscheiden. Diese Diagramme bestehen im allgemeinen aus Einheiten (z.B. Klassen, Objekte oder Zustände), die miteinander in Beziehung stehen. Sie bilden also eine Struktur, die durch einen Graphen modelliert werden kann. Graphgrammatiken eignen sich dann gut, um eine bestimmte Diagrammklasse zu beschreiben.

Graphtransformation bedeutet die regelbasierte Manipulation von Graphen. Ausgehend von einem Startgraphen werden durch wiederholte Regelanwendung alle Graphen der durch eine Graphgrammatik definierten Graphsprache generiert.

Graphgrammatiken sind formal fundiert und lassen sich als Verallgemeinerung von Chomsky-Grammatiken auffassen. In Analogie zur Backus-Naur-Form für textuelle Sprachen können Graphgrammatiken zur präzisen Definition von Diagrammklassen eingesetzt werden, wobei sich die Beschreibung nicht auf die kontextfreien Anteile beschränkt. Diese Definition ist selbst graphisch, denn sie erfolgt in Form von Graphregeln. Damit ist sie im allgemeinen leichter verständlich als vergleichbare textuelle Beschreibungen. Ausserdem wird den Graphregeln ein lokales Layout der neu zu generierenden Anteile durch die graphische Notation inhärent mitgegeben. Darüberhinaus ist Graphtransformation auch zur Beschreibung von Schematransformation und Entwicklungsstrategien für Diagramme geeignet.

## **Eine OMT Beschreibung durch attributierte Grammatiken und Hornklauseln**

**(J. Seemann, J. Wolff v. Gutenberg, Universität Würzburg)**

OO-Entwurfsdiagramme lassen sich als T-Graphen oder Hypergraphen beschreiben. Dabei fassen wir alle "semantiktragenden" Elemente - also Klassen, Assoziationen usw.- als Knoten auf. Ein Diagramm ist eine Menge von meist dreistelligen Relationen. Für das Objektmodell und das dynamische Modell von OMT werden attributierte ELL(1) Grammatiken formuliert. Dadurch ist eine

Abspeicherung von Entwurfsdokumenten im Textformat möglich.  
Diese Texte sind die Grundlage für Validierungs- und Transformationsprogramme.

Die Attribute, die die Konsistenz der Diagramme sichern, und die ggf. auch die graphische Darstellung beschreiben werden mit Prolog-Regeln verwaltet.

Durch eine Präzisierung des Zustandsbegriffes wird ferner versucht, die eigentliche Bedeutung der Entwürfe schon in einer frühen Phase formal darzustellen.

## **Formalizing UML, Can we manage?**

**(C. Pronk, Delft University of Technology)**

The International Standard for the Modula-2 programming language has been defined using both English and the formal specification language VDM-SL. The author of this contribution has been a member of ISO/IEC JTC1/SC22/WG13 (the working group responsible for Modula-2 standardization) since its inception.

The proposed presentation tries to contribute to the goal of formalization of UML by discussing the relevant experiences built-up during the construction of the formal definition of Modula-2 (see ref's below). Issues to be discussed are:

- Choice of formalism to be used
- Form and contents of the Modula-2 standard
- Resolution of ambiguities
- Testing and prototyping the formal definition
- Maintaining the standard
- Standardization in the ISO arena.

References: Pronk, C., N. Plat, A.W.W.M. Biegstraaten. The Use and Construction of Tools for Checking Large Language Definitions. High Integrity Systems, Vol. 1, No 6.

## **Formale Methodik des Entwurfs verteilter objektorientierter Systeme**

**(Bernhard Rumpe, Technische Universität München)**

Der Vortrag stellt einen Teil meiner Dissertation vor. Darin wird eine formale Grundlage für eine objektorientierte Methodik entwickelt. Zur Spezifikation von Struktur und Verhalten verteilter objektorientierter

Systeme werden Beschreibungstechniken für Objektmodelle, Klassen und Transitionssysteme definiert.

Für die zustandsbasierte Beschreibung nichtdeterministischer Komponentenverhaltens wird die Theorie buchstabierender Automaten vorgestellt. Es werden eine konkrete Darstellungsform, eine abstrakte Syntax, und eine denotationelle Semantik charakterisiert.

Für buchstabierende Automaten wird ein Verfeinerungskalkül definiert, der zur Transformation von abstrakten in detaillierte Verhaltensbeschreibungen verwendet werden kann. Es wird argumentiert, daß dieser Kalkül bezüglich der Semantikdefinition korrekt ist. Der Kalkül wird für die Spezialisierung und die Vererbung von Verhaltensbeschreibungen in verteilten objektorientierten Systemen eingesetzt.

Ein Systemmodell charakterisiert eine Menge von verteilten objektorientierten Systemen, die aus asynchron kommunizierenden Agenten aufgebaut sind. Das Systemmodell dient als Basis für die Definition einer integrierten, formalen Semantik für die oben genannten Beschreibungstechniken.

Der vorgestellte Verfeinerungskalkül stellt in einer Methodik verwendbare Entwicklungsschritte zur Verfügung.

Die Verbindung graphischer Beschreibungstechniken mit einer integrierten, formalen Semantik nutzt Synergieeffekte formaler und praxisorientierter Ansätze der Softwaretechnik.

## **Formale objektorientierte Softwareentwicklung mit *FOX***

**(Klaus Achatz, Wolfram Schulte, Universität Ulm)**

Wir stellen eine neue formale, objektorientierte Methode vor, genannt *FOX*. Sie ist eine synergetische Kombination aus der semi-formalen Fusion Methode und der formalen Spezifikationssprache Object-Z.

*FOX* unterscheidet strikt zwischen Analyse und Entwurf, um die Komplexität und die unterschiedlichen Aufgaben der Softwareentwicklung zu bewältigen. In jeder Entwicklungsphase werden Zustands- und Verhaltensspezifikationen erstellt. Sie können graphisch oder textuell sein. Wir geben Beweisverpflichtungen an, um sicher zu stellen, daß die entwickelten Spezifikationen formal

konsistent und vollständig sind, und daß das resultierende System zur Ausgangsspezifikation konform ist. Wir illustrieren die Anwendung von *FOX* an einem einfachen Beispiel, einem Grapheditor.

## **Operationale Semantik für visuelle Sprachen**

**(Jürgen Ebert, Alexander Fronk, Universität Koblenz-Landau)**

Aufbauend auf abstrakte Syntaxgraphen visueller Dokumente, wie sie beispielsweise im Zusammenhang mit objekt-orientierten Entwicklungsansätzen verwendet werden, wird eine Z-Spezifikation der Semantik durch die Angabe semantischer Basen vorgeschlagen. An Beispielen wird gezeigt, wie semantische Basen verschiedener Dokumente miteinander verknüpft werden können, um zu einer integrierten Semantikbeschreibung zu kommen.

## **Formalisierung der Syntax von objektorientierten Methoden**

**(Roger Süttenbach, Universität Koblenz-Landau)**

In diesem Vortrag wird der in Koblenz entwickelte EER/GRAL-Ansatz (Extended Entity Relationship/GRaph specification Language) zur Formalisierung von objektorientierten Methoden verwendet. Durch die Beschreibung mittels ER-Diagrammen und GRAL-Prädikaten ergibt sich eine präzise und integrative Darstellung. Dabei beschränkt sich die Formalisierung zur Zeit auf die abstrakte Syntax der in einer Methode verwendeten Modellierungssprachen inklusive zugehöriger Konsistenzbedingungen.

Formalisierungen für die Booch Methode und die OMT Methode wurden bereits erarbeitet. Dabei wurde besonderes Gewicht auf die Vollständigkeit der Beschreibung hinsichtlich der verwendeten Konzepte und Konsistenzbedingungen gelegt. Die Mittel, Prinzipien und die Vorgehensweise zur Erstellung einer solchen Formalisierung werden an einem anschaulichen Beispiel aus der Booch Methode verdeutlicht.

# Formalisierung der Semantik objektorientierter Methoden

(Cornel Klein, Bernhard Rumpe, Technische Universität München)

Objektorientierte Analyse- und Designmethoden erlauben die Modellierung eines Systems auf verschiedenen Abstraktionsebenen und unter verschiedenen Sichten. Während für einzelne Beschreibungstechniken (wie z.B. ER-Modelle, Automaten) teilweise relativ präzise, formal definierte, Semantiken existieren, fehlt bislang eine integrierte Semantik für die Beschreibungstechniken einer Methode. Diese ist notwendig für semantikerhaltende Transformationen zwischen Dokumenten, für umfangreiche Konsistenzüberprüfungen, sowie für die Erstellung fortgeschrittener Werkzeugunterstützung für die Generierung und Analyse. Wir stellen einen Ansatz zur formalen Definition einer solchen integrierten Semantik vor. Dieser baut auf einem mathematischen Systemmodell auf, welches zentrale Konzepte informationsverarbeitender Systeme wie *Komponente*, *Nachricht*, *Komposition* etc. formal faßt. Zu diesem mathematischen Systemmodell können alle Beschreibungstechniken einer Softwareengineering-Methode in Beziehung gesetzt werden. Wir berichten über Erfahrungen, die wir bei einer derartigen Formalisierung eines Ausschnitts der Unified Modelling Language (UML) gewonnen haben.