

Auf dem Weg zu einer
**Integrierten formalen Semantik
für Teile der UML**

Markus Klein, Benjamin Braatz

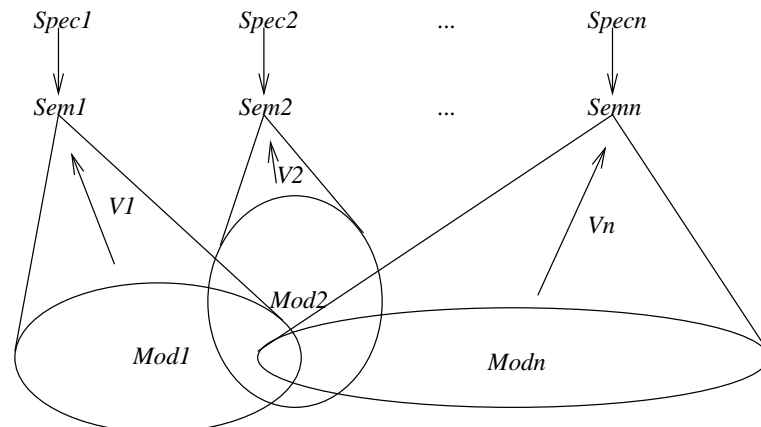
TFS-Forschungskolloquium, 12.6.2003

Übersicht

- Motivation
- Algebra-Transformations-Systeme
- Interpretation von UML-Diagrammen
- Konsistenz von UML-Spezifikationen
- Komponenten für UML-Spezifikationen
- Mögliche Anwendungen

Motivation

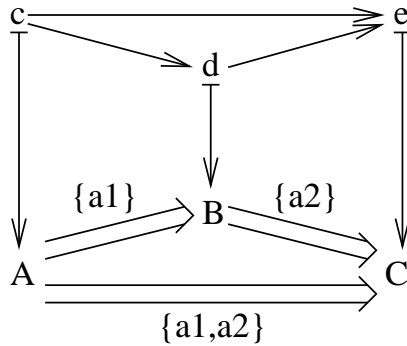
- Gemeinsamer semantischer Bereich für verschiedene **Viewpoints**



- Formale Untersuchung der **Konsistenz** von Spezifikationen
- “**Semantics First**”-Konzepte für **Beziehungen** zwischen Spezifikationen und **Komponenten**
- Präzise Begriffsbildung erleichtert und ermöglicht **Tool-Unterstützung** und **Code-Generierung**
- Untersuchung der Beziehungen zwischen Sichten, Schichten und Ebenen des **ODEMA-Würfels**

Algebra-Transformations-Systeme [Gro '01b]

- **Algebra-Transformations-System** $TS = (TG, m)$ zu einer Signatur $D\Sigma$
 - **Kontroll-Graph** TG
 - **Markierungs-Morphismus** $m: TG \rightarrow \mathbb{D}_{D\Sigma}$



- **Daten-Zustands-Signatur** $D\Sigma = (\Sigma, AC)$
 - **Algebraische Signatur** $\Sigma = (S, OP)$ mit Sorten S und Operationen $OP = OP(w \rightarrow s)_{w \in S^*, s \in S \cup \{\lambda\}}$
 - **Aktionen** $AC = AC(w, w')_{w, w' \in S^*}$
- **Daten-Zustands-Raum** $\mathbb{D}_{D\Sigma} = (|PAlg(\Sigma)|, T, \perp, \emptyset)$
 - **Partielle Algebren** $|PAlg(\Sigma)|$ als Daten-Zustände
 - **Aktions-Mengen** $T(Al, Al') = \mathcal{P}(\{ac(in, out) | ac \in AC(w, w'), in \in Al(w), out \in Al'(w')\})$ als Transitionen
 - **Komplett undefinierte Algebra** $\perp \in |PAlg(\Sigma)|$ als Initialisierungs- und Finalisierungs-Zustand
 - **Leere Mengen** $\emptyset \in T(Al, Al')$ als Idle-Transitionen

- **Transitions-Graph** $TG = (CS, T, in, id)$
 - **Kontrollzustände** CS
 - **Transitionen** $T = T(c, d)_{c, d \in CS}$
 - **Initialisierungs- und Finalisierungs-Zustand** $in \in CS$
 - **Idle-Transitionen** $id(c) \in T(c, c)$
- **Transitions-Graph-Morphismus** $m: TG \rightarrow TG'$
 - **Zustands-Abbildung** $m_{CS}: CS \rightarrow CS'$ mit $m_{CS}(in) = in'$
 - **Transitionen-Abbildungen**

$$m_{T(c,d)}: T(c, d) \rightarrow T(m_{CS}(c), m_{CS}(d))$$
mit $m_{T(c,c)}(id(c)) = id'(m_{CS}(c))$

Interpretation von UML-Diagrammen

- **Rollen für UML-Diagramme:**
 - Klassen-Diagramme \rightarrow Signatur
 - Statecharts \rightarrow Objekt-Verhalten
 - Action Language \rightarrow Daten-Transformation
 - OCL \rightarrow Daten-Eigenschaften
 - Sequenz-Diagramme \rightarrow Kommunikations-Eigenschaften

- Daten-Zustände: **Objekt-Konfigurationen** [Par '01, Ten '01]
Aktionen: Atomare Operationen, Ereignisse ???
- Semantik für Statecharts und Sequenz-Diagramme auf Basis von **Labelled Transition Systems** als Kontroll-Graphen [Özh '01]
- OCL und Action Language **open**

- **Entsprechende Signatur:**

$$\text{Sig}(CD) =$$

sorts : $\text{Integer}, \text{Class1}, \text{Class2}$

opns : $\text{att1} : \text{Class1} \rightarrow \text{Integer}$

$\text{att2} : \text{Class2} \rightarrow \text{Integer}$

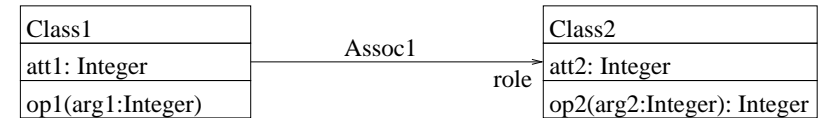
$\text{role} : \text{Class1} \rightarrow \text{Class2}$

acts : $\text{op1} : \text{Class1 Integer}; \lambda$

$\text{op2} : \text{Class2 Integer}; \text{Integer}$

Beispiel

- **Klassendiagramm:**

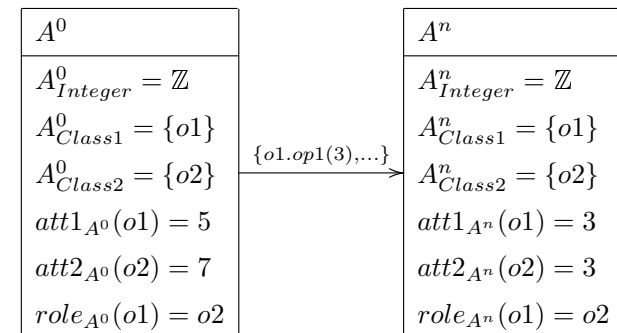


- **Action-Spezifikationen:**

```

action Class1.op1(arg1):
    att1 := role.op2(arg1);
action Class2.op2(arg2):
    att2 := arg2;
    return att2;
  
```

- **Teil des Daten-Zustands-Raumes:**



- **Verschiedene Möglichkeiten für Transitionen:**

$$A^0 \xrightarrow{\{o1.op1(3)_{call}\}} A^1 \xrightarrow{\{o2.op2(3)_{call}\}} A^2 \xrightarrow{\{o2.op2(3)_{ret}\}} A^4 \xrightarrow{\{o1.op1(3)_{ret}\}} A^n$$

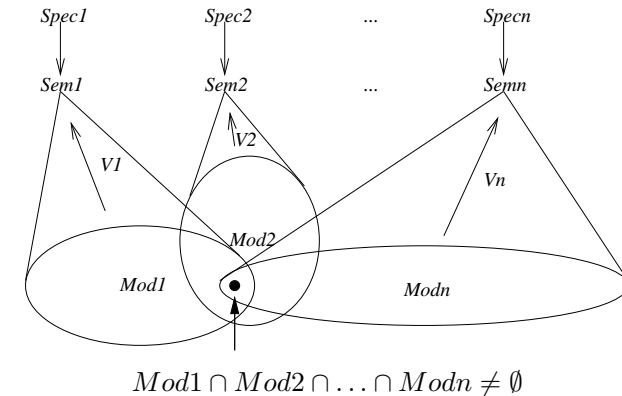
$$A^0 \xrightarrow{\{o1.op1(3), o2.op2(3)\}} A^n$$

$$A^0 \xrightarrow{\{o1.op1(3)_{call}\}} A^1 \xrightarrow{\{o2.op2(3)_{call}, o2.op2(3)_{ret}\}} A_4 \xrightarrow{\{o1.op1(3)_{ret}\}} A^n$$

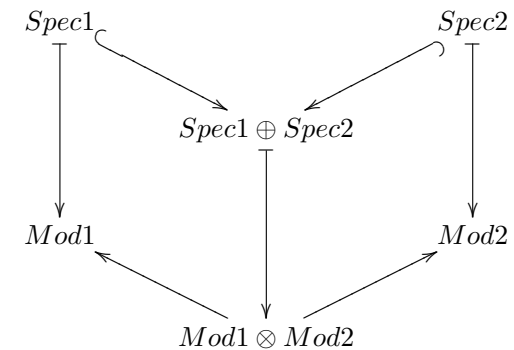
- **Syntaktische Konsistenz** aller anderen Diagramme mit dem Klassendiagramm
- **Konstruktion** eines minimalen Modells aus Statecharts und Action Language (konstruktive Techniken)
- **Verifikation** von OCL-Constraints und Sequenz-Diagrammen an diesem Modell (deskriptive Techniken)
Für Sequenz-Diagramme und Statecharts in [Özh '01, Tsi '01]
- **Widerspruchsfreiheit** der OCL-Constraints und Sequenz-Diagramme untereinander

Konsistenz von UML-Spezifikationen

- **Definition: Konsistenz** einer Spezifikation, wenn **mindestens ein Modell** existiert

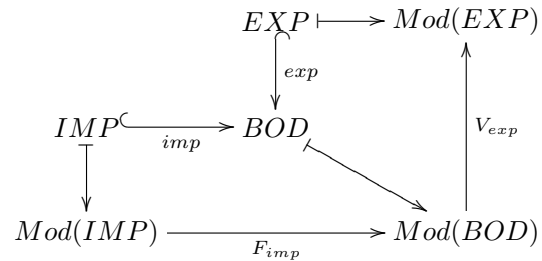


- **Kompositionalität** konsistenter Spezifikationen



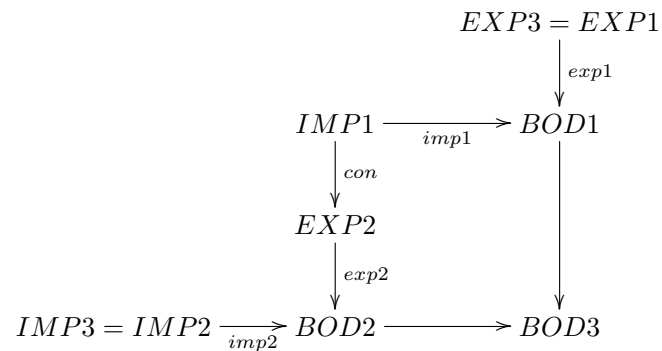
Komponenten für UML-Spezifikationen

- **Wiederverwendbarkeit** und **Austauschbarkeit** durch **Kapselung des Body** hinter **Import** und **Export**

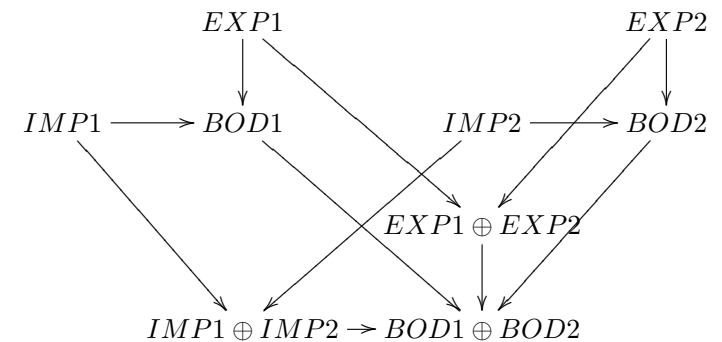


- **Korrektheit** einer Komponente, wenn der Body den Export mit Hilfe des Imports **realisiert**
- *exp* ist eine **Verfeinerung**
- *imp* ist eine **Konstruktion**
- Im- und Export-Konnektoren können aus **Regel-Anwendungen** zusammengesetzt werden [EO+ '02, Kle '03]
- Syntaktische Komponenten für UML in [Pii '03]

- **Sequentielle Komposition** von Komponenten



- **Vereinigung** von Komponenten



Mögliche Anwendungen

- Tool zur Konsistenzprüfung von UML-Modellen
- UML-Profil und Korrektheitsprüfung für Komponenten
- Regelbasierter Editor für Komponenten

UML-Integration:

[Par '01] Daniel Parnitze: On Formal Semantics of Object Systems with Data and Object Attributes

Bericht Nr. 2001/05, Fak. IV, TU Berlin

[Ten '01] Jennifer Tenzer: A Formal Semantics of UML Class Diagrams based on Transformation Systems

Bericht Nr. 2001/09, Fak. IV, TU Berlin

Konsistenzanalyse:

[Tsi '01] Aliko Tsiolakis: Semantic Analysis and Consistency Checking of UML Sequence Diagrams

Bericht Nr. 2001/06, Fak. IV, TU Berlin

[Özh '01] Mesut Özhan: Semantische Konsistenzanalyse von UML Sequenz- und Zustandsdiagrammen

Bericht Nr. 2001/07, Fak. IV, TU Berlin

IOSIP-Publikationen

Überblickspapiere:

[Gro '00] Martin Große-Rhode: Using a formal reference model for consistency checking and integration of UML diagrams

IDPT 2000

[Gro '01a] Martin Große-Rhode: Formal concepts for an integrated internal model of the UML

UniGra 2001, ENTCS Vol. 44 Issue 4

Algebra-Transformations-Systeme:

[Gro '01b] Martin Große-Rhode: Semantic Integration of Heterogeneous Formal Specifications via Transformation Systems

Bericht Nr. 2001/13, Fak. IV, TU Berlin

Komponenten:

[EO+ '02] Hartmut Ehrig, Fernando Orejas et al.: A Transformation-Based Component Framework for a Generic Integrated Modeling Technique

IDPT 2002, Bericht Nr. 2002/02, Fak. IV, TU Berlin

[Kle '03] Markus Klein: A Component Concept for System Modeling Based on High-Level Replacement Transformations

Bericht Nr. 2003/09, Fak. IV, TU Berlin

[Pii '03] Martti Piirainen: Applications of a Generic Component Framework to a UML Case Study in Production Automation

Diplomarbeit, Fak. IV, TU Berlin