

# **Verbesserung modellgetriebener Softwareentwicklung durch die semantische Integration verschiedener Sichten auf ein System**

Schäbige Bartour Seminar Halle

# Einordnung des Themas

- MDE wird immer bedeutender
- Grundlage: qualitativ hochwertige Modelle
- Entwicklung hin zu automatischer Generierung von Code
- Vollständige Automatisierung erfordert konsistente, formalisierbare Modelle

# Einordnung des Themas

- Modelle größtenteils nur Dokumentation, veraltet, wirkliche MDE findet selten statt
- Woran liegt das?
- Wie kann das geändert werden?
- Ist das mit der UML in der heutigen Form realisierbar?

# Wozu soll dieser Vortrag dienen?

- Vorstellung meines geplanten Themengebietes
- Feedback und Anregungen von euch
  
- Zusammenfassung ungefähr eines Monats: manche Gedanken sind noch nicht zu Ende gedacht und formalisiert

- Aktueller Stand MDE
- Motivation / Kritik an der UML
  - Grundlegendes
  - Komplexität, Abstraktion, Semantik, Konsistenz
- Ziele der Dissertation
  - Die Vision
  - Vorteile zur UML, Eigenschaften
- Wissenschaftlicher und Praktischer Beitrag
- Potentielle Gefahren
- TODO-Liste / Ungeordnete Fragen
- Konferenzen / Journals

# Warum überhaupt MDE?

- Steigende Komplexität im Software Engineering
- (Gute) Modelle erleichtern Verständnis, Kommunikation
- Modelle führen automatisch zu mehr Dokumentation
- Je eher Fehler behoben werden desto besser

# Aktueller Stand MDE: DSLs

- Domain Specific Languages - Formale Sprachen für bestimmte, abgeschlossene Domänen
- z.B. Sprache für UI, Geschäftsprozesse, Dialogsteuerung in Webanwendungen
- Codegenerierung aus dem Modell an sich möglich
- Benötigt eigene Modellierungswerkzeuge
  
- Realisierung mittels UML-Profilen
  - SysML, CORBA, Software Radio, etc.
- Entwicklung einer komplett neuen Sprache (SQL, Latex)

- Initiative von Microsoft und Partnern
- Allgemeines Modell der Softwareentwicklung, Beschreibung einer Vorgehensweise
- Kombination aus Methodenbibliothek und erweiterbarer Entwicklungsumgebung
  - Bibliothek besteht aus Frameworks, Patterns, DSLs, Designer, Generatoren
  - Erweiterbare Entwicklungsumgebung ist Visual Studio
  - Verbunden mit Anleitung zur Verwendung der Werkzeuge ergibt sich Software Factory

# Aktueller Stand MDE: MDA

- Model Driven Architecture - Spezifikation der OMG zur modellgetriebenen Softwareentwicklung
- Ziel: gesamten Entwicklungsprozess in Modellen abbilden
  - System selbst wird mittels Modelltransformationen generiert
- Schrittweise Verringerung der Abstraktion
  - Platform Independent Model (PIM) wird mittels Platform Description Model (PDM) in Platform Specific Model (PSM) überführt
- PIMs und PSMs werden über UML-Profile definiert
  
- Keine Festlegung auf UML, verwendete Modellierungssprache muss aber kompatibel zur Meta Object Facility (MOF) sein

# Kritik an der UML: Grundlegendes

- Zusammenfassung objektorientierter Modellierungsansätze
- Kompromisse bei der Entwicklung [RAM98]
  - Amigos nicht gerade „pflegeleicht“
  - Unnötige Komplexität
- Kein Vorgehensmodell, nur Artefakte
- Konsortium aus Industriepartnern, die wollen Tools verkaufen

# Kritik an der UML: Komplexität

- Verschärft durch fehlendes Vorgehensmodell
- Komplexität erschwert den Einstieg in die Modellierung
- Führt zu komplexen und unintuitiven Werkzeugen

# Kritik an der UML: Formale Semantik

- Spezifikation ist Kombination von semiformalen Diagrammen, Randbedingungen und informalem Text
  - Erschwert Finden und Beseitigen von Fehlern, Unvollständigkeiten und Inkonsistenzen in Modellen
  - Bsp. Abschnitt Semantik von Assoziationen: *If a composite is deleted, all of its parts are **normally** deleted with it.*
  - Führt zu unterschiedlichen Implementierungen in Werkzeugen
- Fehlende Semantik in Modellen muss im Generator / Compiler stecken
  - Brauchen wir für jede Software einen eigenen Compiler?
- Ansätze zur Semantik einzelner Diagramme
- Integrierte Semantik über Graphen, z.B. [KUS09]

# Kritik an der UML: Konsistenz

Title:  
Creator:FreeHEP Graphics2D Driver  
CreationDate:Donnerstag, 22. Oktober 2006  
LanguageLevel:3

Title:  
Creator:FreeHEP Graphics2D Driver  
CreationDate:Donnerstag, 22. Oktober 2006  
LanguageLevel:3

- Folge fehlender Semantik
- Bedingungen in natürlicher Sprache
  - [...]The signature must either refer to an Operation [...]
  - Formalisierung fehlt
- Auswirkungen auf die Modellqualität
- Implementierung in Tools nötig [EGY06]

# Kritik an der UML: Abstraktion

Title:

Creator:FreeHEP Graphics2D Driver

CreationDate:Donnerstag, 22. Oktober 2009 11:

LanguageLevel:3

```
public class SBTTeilnehmer
{
    private int fuellstand;

    public void befuellen()
    {}

    public void heimgehen()
    {}
}
```

- Klassen sind Rechtecke
- Abbildung der Diagramme auf Code und nicht auf Konzepte
  - Sequenzdiagramm: Nachricht ist Operation / Signal

# Kritik an der UML: weiteres

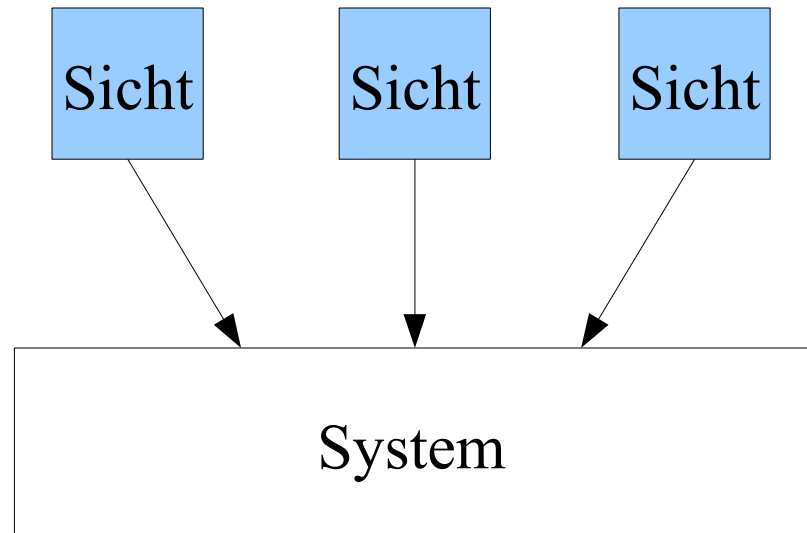
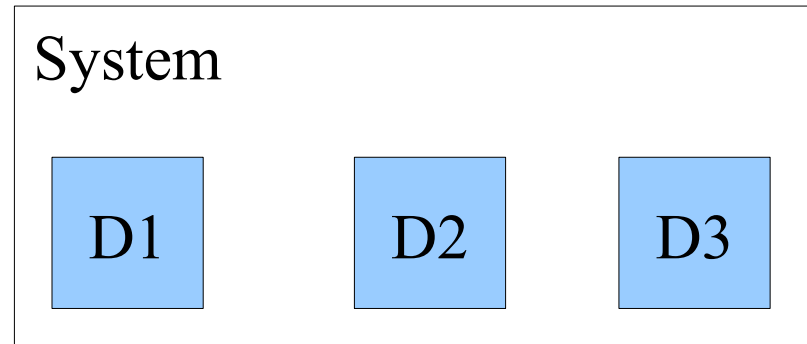
- Refactoring von Modellen
  - Bsp.: Welche Auswirkungen hat es auf das Modell, wenn ich eine Methode aus dem Klassendiagramm entferne?
  - Folge fehlender Semantik
- Integration weiterer Programmierparadigmen
  - UML als objektorientierte Modellierungssprache
  - Aspektorientierung, funktionale / prozedurale Programmierung
  - Unklar, ob das notwendig ist

# Ziele der Diss

- Definition einer in sich konsistenten und formalisierten Modellierungssprache
  - einzelne Sichten auf das System sind miteinander integriert und stehen in semantischer Relation zueinander
  - trotz Formalisierung soll Kreativität beim Modellieren möglich sein
- Angabe eines Vorgehensmodells mit der Sprache
  - Abbildung vorhandener Vorgehensmodelle (RUP, Agile, etc.)

# Die Vision

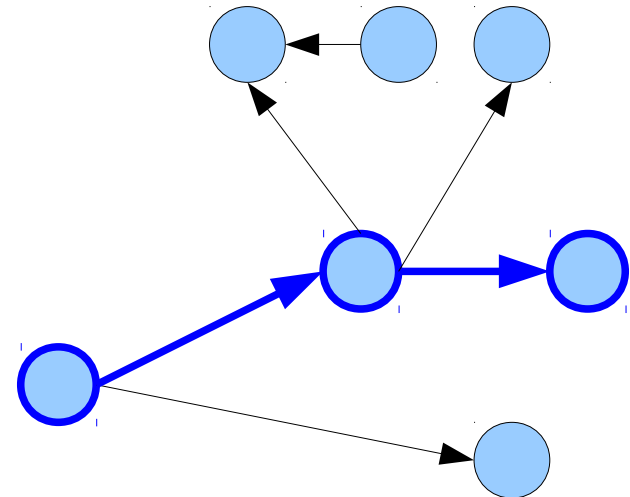
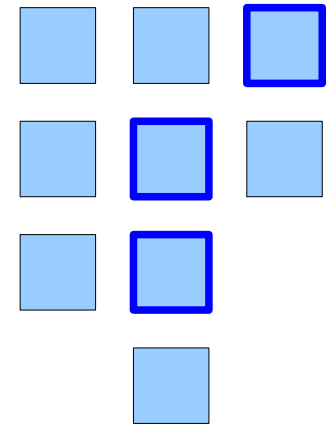
- Diagramme (Sichten) der UML beschreiben ein System, sind aber unabhängig voneinander
- Das System wird durch ein Modell beschrieben
- Diagramme sind wirklich nur Sichten



# Vorteile zur UML

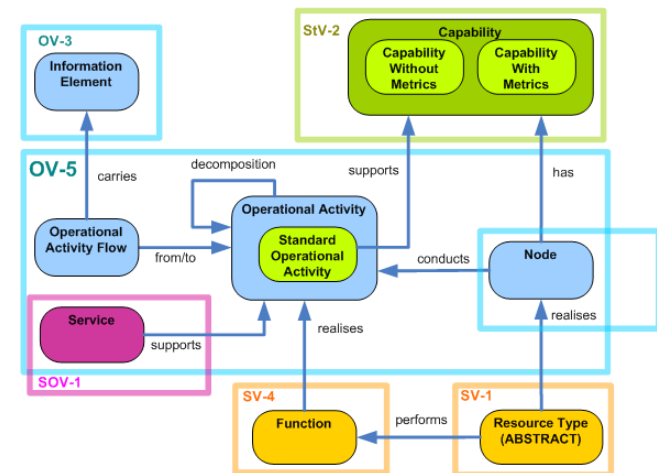
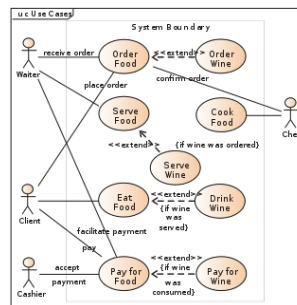
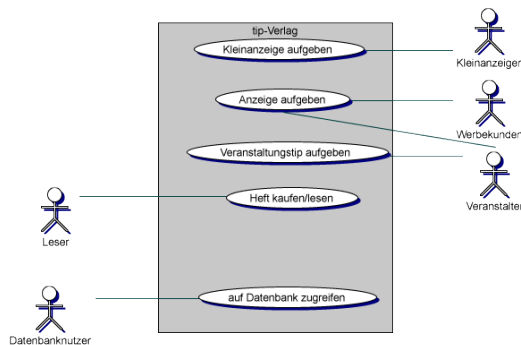
- Ergeben sich durch das zu Grunde liegende sichtübergreifende Datenmodell
- Leichte Informationsbeschaffung
  - Welcher Systemteil hat welche Aufgabe?
- Automatische Modellabstraktion
  - Stufenlose Modellverfeinerung
- Gleichzeitige Darstellung mehrerer Detailebenen

- Zeige die Modellteile, die für Logging zuständig sind.
- Ansätze
  - Annotationen am Modell
  - Aufteilung in Aspekte
  - ...



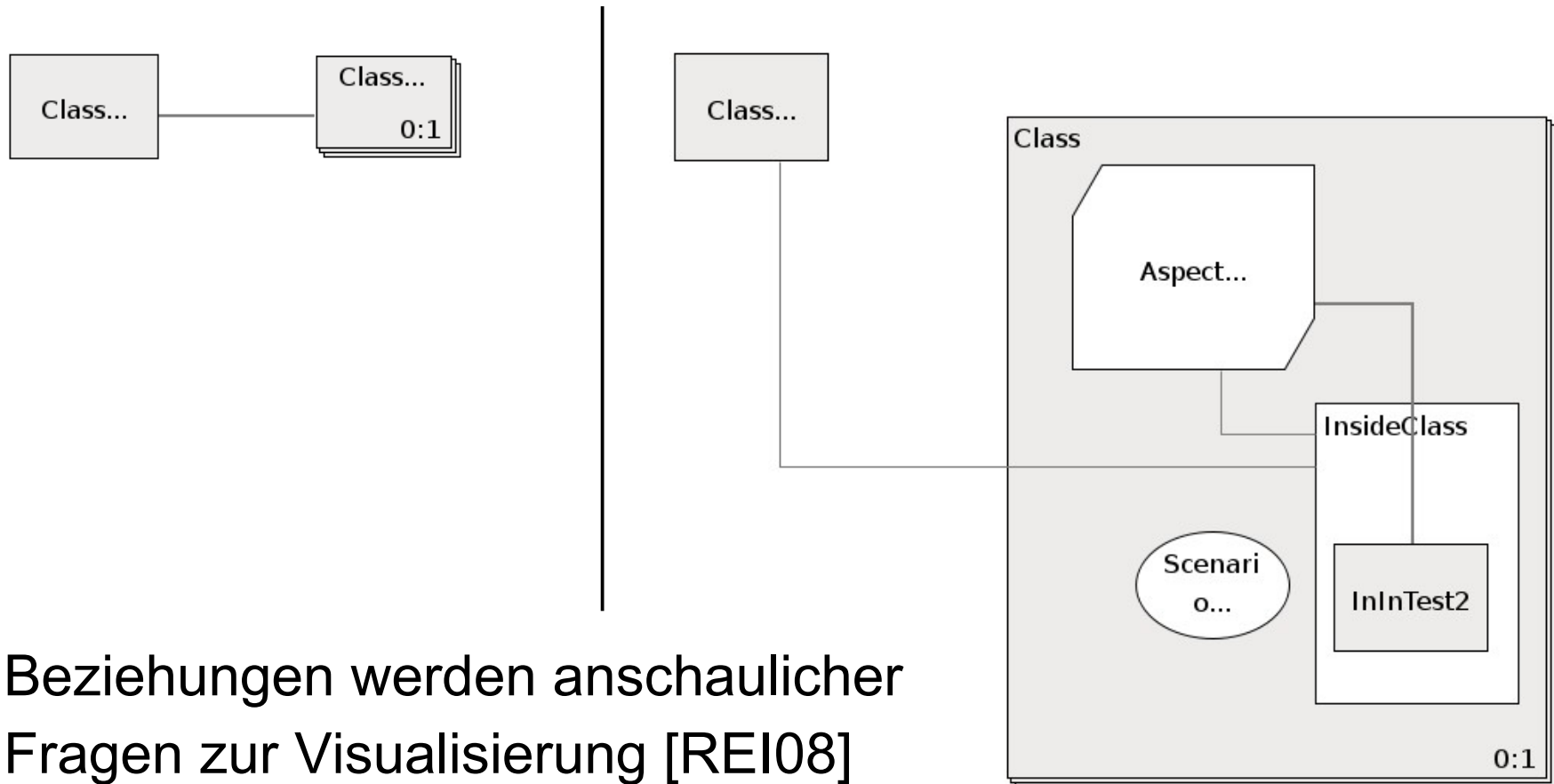
# Automatische Abstraktion

- Anzeige bestimmter Modellteile auf einer Abstraktionsebene
- Zusammenfassen von Modellteilen in höheren Abstraktionen
- Einteilung in Sichten: Management, Auftraggeber, Implementierer, etc.



Sinkender Abstraktionsgrad →

# Darstellung mehrerer Detailebenen



- Beziehungen werden anschaulicher
- Fragen zur Visualisierung [REI08]

- Ganzheitlicher Ansatz, der Konzepte der UML an sich hinterfragt
  - Aktuell viele Forschungen zur Verbesserung einzelner Diagramme
  - „Frickelei“ an UML erhöht deren Komplexität noch weiter
  - Entwicklung neuer Profile statt grundlegender Hinterfragung
- Verbreiterung der Forschungsansätze zu Modellierungssprachen
- Direkte Verwendung in der Praxis durch zugehöriges Vorgehensmodell
  - Voraussetzung: Werkzeuge vorhanden
- Bessere Unterstützung durch Werkzeuge möglich
  - Konsistenz erleichtert Modellevolution, Vergleich von Modellen
- Impulse für UML 3.0

- UML3 erscheint vor Ende der Diss mit den gleichen Intentionen
  - UML2 Request for Information Aug 99, finale Spezifikation Sep 04
  - UML 2.3 geplant für Anfang 2010
- Ansatz zu weit gefasst – GPL für Modellierung nicht möglich
  - Eingrenzung auf bestimmte Anwendungsbereiche
  - Wäre das die Erstellung eines UML-Profiles?

- Related Work
  - UML Spezifikation
  - Anforderungen MDE
  - Alternative Modellierungssprachen / Vorgehensweisen
  - Formale Beschreibungen
- Gedanken formalisieren und ordnen, Schlüssiges Konzept ausarbeiten
- Ende des Jahres: Vision als Paper / Proposal

# Ungeordnete Fragestellungen

- Ist es überhaupt möglich eine General Purpose Modellierungssprache zu erstellen?
- Welche Sichten sollen wie modelliert werden? Mit welchen Notationselementen?
- Wie lässt sich Formalisierung mit Kreativität vereinbaren?
- Welche alternativen (zur UML) Modellierungsansätze gibt es bereits? Warum haben die sich nicht durchgesetzt?
- Wie können Verbesserungen evaluiert werden?
- Wie lassen sich Software Factories einordnen?

# Konferenzen und Journals

- Fachtagung Modellierung (Mrz)
- Human Factors in Computing Systems (Apr)
- Workshop on Behavioural Modelling in MDA (Jun)
- Model Driven Engineering Languages and Systems (Okt)
- NASA Formal Methods Symposium (Apr)
  
- Software and Systems Modeling (Springer)

- [EGY06] Egyed. Instant consistency checking for the UML. ICSE 2006.
- [MER05] Mernik, Heering. When and how to develop domain-specific languages. ACM Computing Surveys, 37(4), 2005.
- [RAM98] Rambaugh, Jacobson Booch, UML Reference Manual. 1998
- [REI08] Reinhard et al. Tool support for the navigation in graphical models. ICSE 2008
- [KUS09] Kuske et al. Towards an integrated graph-based semantics for UML. SoSym, 8(3), 2009.