25. Meta-CASE-Werkzeuge

Prof. Dr. Uwe Aßmann
Technische Universität Dresden
Institut für Software- und Multimediatechnik
http://st.inf.tu-dresden.de
Version 11-1.0, 08.12.11

1) Meta-CASE-Werkzeuge
2) MetaEdit+
3) MOFLON Meta-CASE-Werkzeug
    1) MOFLON Architektur (ext.)
4) FlowR ScreenFlow-Umgebung (ext.)

Obligatory Reading

- MetaCase. Domain-Specif c Modeling With Metaedit+: 10 Times Faster Than UML. White paper. http://www.metacase.com/papers/Domain-specif c_modeling_10X_faster_than_UML.pdf
- MetaCase. Abc To Metacase Technology. http://www.metacase.com/papers/ABC_to_metaCASE.pdf

Literatur

- [Nill] C. Nill. Analysis and Design Modeling Using Metaphorical Modeling Entities. A Modeling Language for the Tools and Materials Approach. Diplomarbeit Technische Universität Dresden, 2006.
- http://www.metacase.com/support/45/manuals/index.html

Literatur

- [Nill] C. Nill. Analysis and Design Modeling Using Metaphorical Modeling Entities. A Modeling Language for the Tools and Materials Approach. Diplomarbeit Technische Universität Dresden, 2006.
- http://www.metacase.com/support/45/manuals/index.html

25.1 Meta-CASE-Werkzeuge

Nutzung von Meta-CASE

- Ein **Meta-CASE-Werkzeug** ist eine Entwicklungsumgebung für den Entwurf von SEU und Softwarewerkzeugen
  - Herstellung einer individuell angepassten Umgebung aus einem Guss
    - Generierung von Repositorien mit Frontend- und Backend-Tools für Austauschformate
    - Generierung von Editoren und Kompositionswerkzeugen für Artefakte
    - Kompositionssysteme zur Komposition von Werkzeugen
  - Modellierung von textuellen und graphischen Sprachen
  - Modellierung von domänenspezifischen Sprachen und ihren Werkzeugen (domain-specific languages, DSL)
- Speziell an die Domäne angepasste Entwurfsmethoden verbessern die Produktivität des Teams
  - An den Anwendungsfall angepasste Software-Entwicklungswerkzeuge bringen eine höhere Effizienz
    - Domänenspezifische Methoden sind 5 bis 10 mal schneller als die sonst übliche (UML-)Notation (MetaCase erzielte bei Nokia 10-fache Produktivitätssteigerung)

Quelle: Domain-Specif c Modeling: 10 Times Faster Than UML; Whitepaper MetaCase 2005;

URL: http://www.metacase.com/de/

Weitere Beispiele zu Meta-CASE

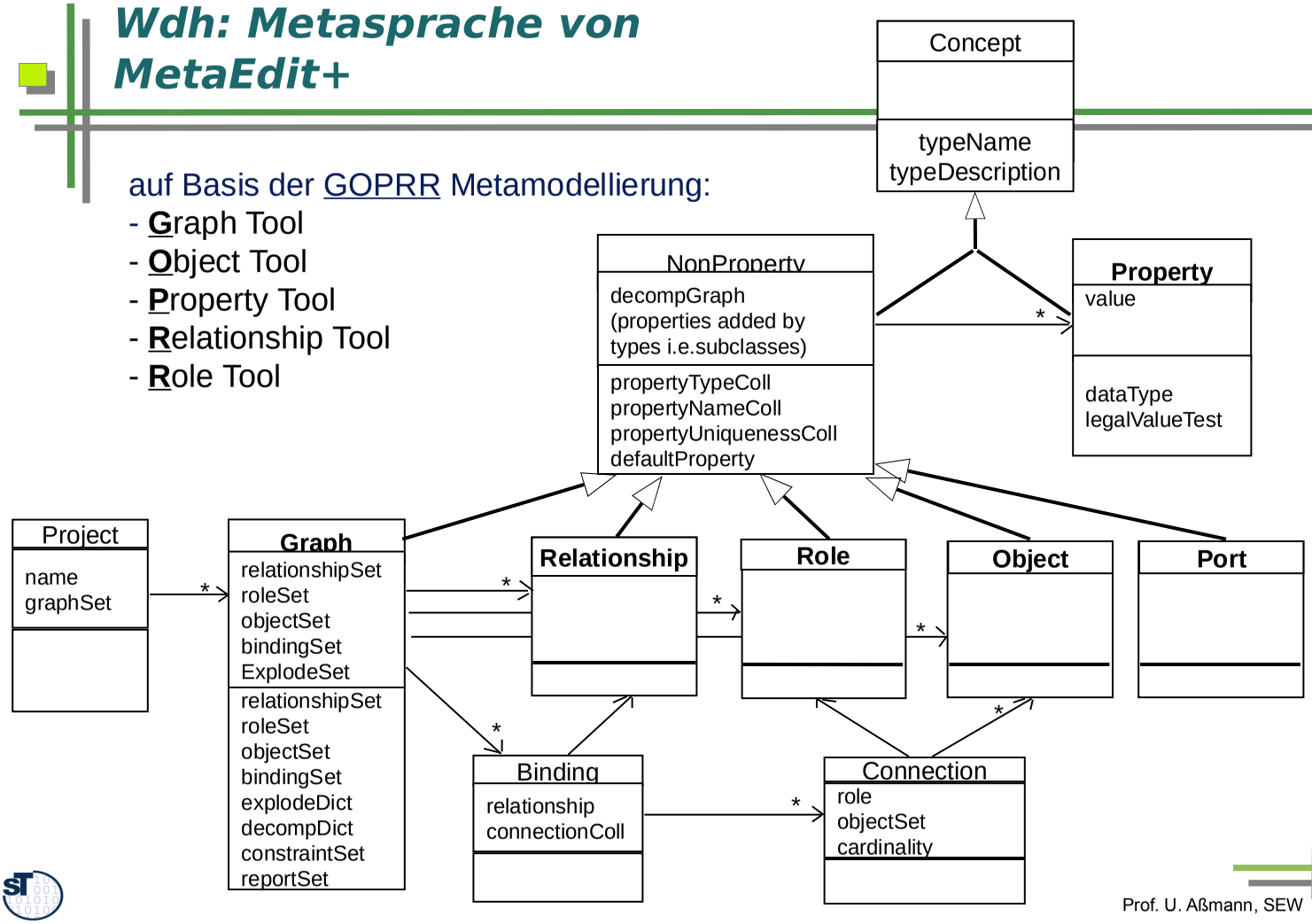
- KOGGE, JKOGGE: Generator für grafische Entwurfsumgebungen
  - KOGGE basiert auf einer formalen Meta-Tool-Beschreibung und einem Interpreter (Prof. Ebert, Uni Koblenz)
    - http://www.uni-koblenz-landau.de/koblenz/fb4/institute/IST/AGEbert/MainResearch
- MetaEdit+: Parametrisierbares CASE-Tool mit
  - Editor für Metamodelle (MetaEdit+ Metasprache)
  - Generator für die Erstellung der Methodenbeschreibung
- Eclipse Modeling Facility (EMOF):
  - Benutzt eine Teilmenge von MOF
- OpenArchitectureWare (EMOF)
- Netbeans: IDE basierend auf MOF
- MOFLON: IDE basierend auf MOF, mit Storyboards und TGG
- FlowR: ScreenFlows

25.2 MetaEdit+ von MetaCase

- http://www.metacase.com/download/ Evaluation version
- http://www.metacase.com/cases/dsm_examples.html Many more DSL examples
- http://www.metacase.com/resources.html Articles and handbooks

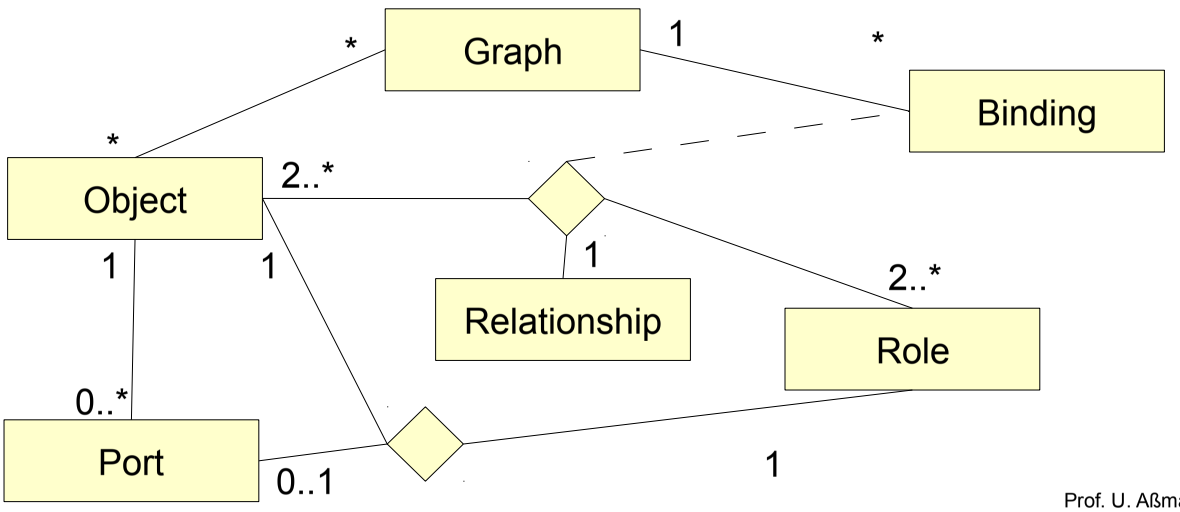
Wdh: Metasprache von MetaEdit+

auf Basis der GOPRR Metamodellierung:
- **G**raph Tool
- **O**bject Tool
- **P**roperty Tool
- **R**elationship Tool
- **R**ole Tool



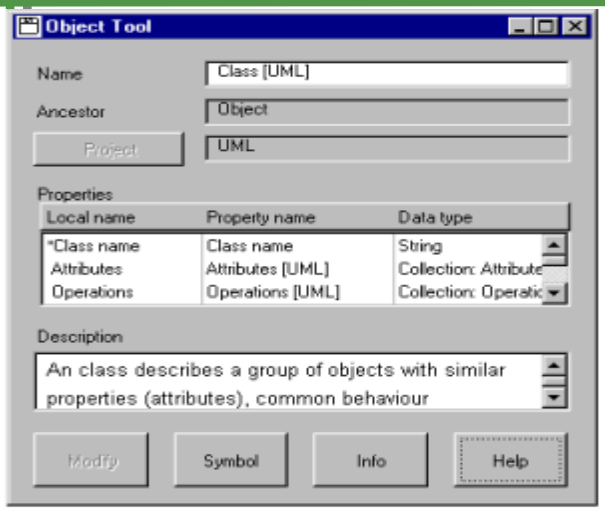
Wdh: Graph Types in MetaEdit+

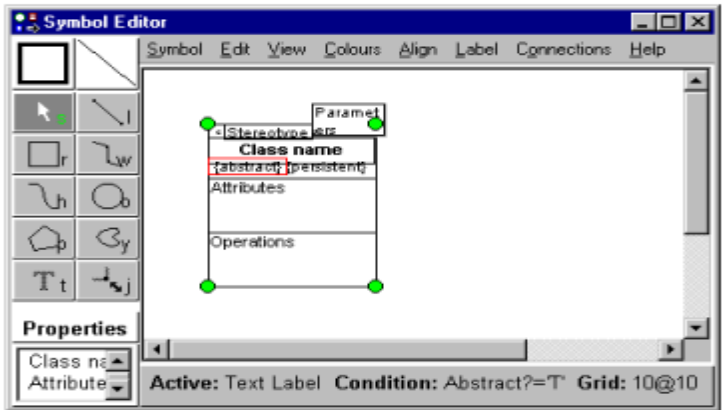
- A **graph type (diagram)** defines:
  - Objects
  - Roles
  - Relationships
  - Allowed Bindings between all entities:
    - a binding consists of a relationship with roles and playing objects



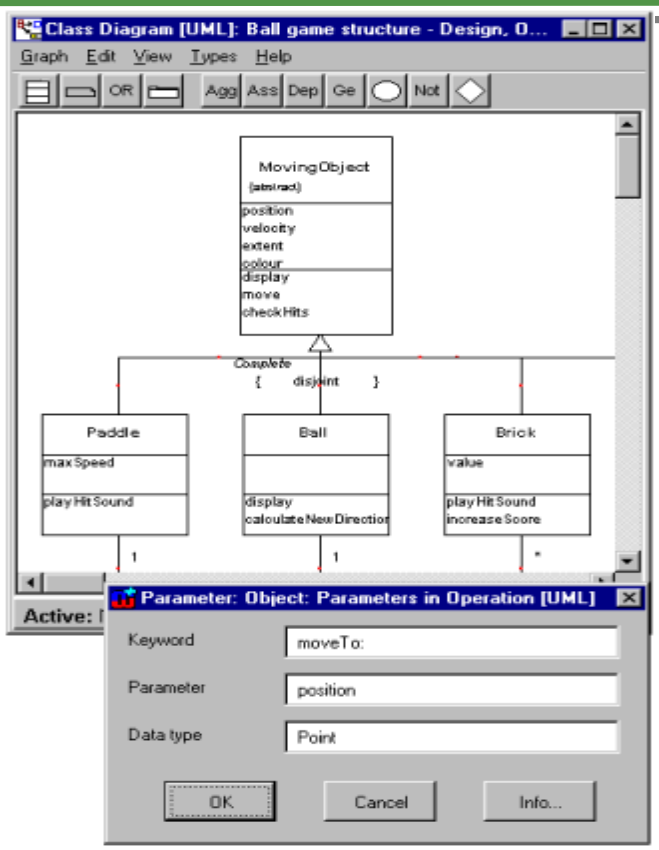
Erstellen eines eigenen CASE-Tools mit MetaEdit+

Entwurf der eigenen Methode



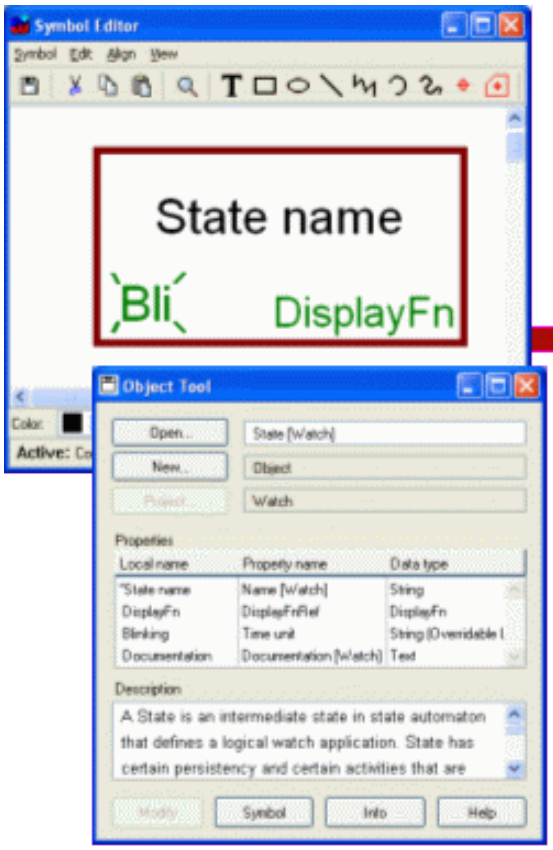


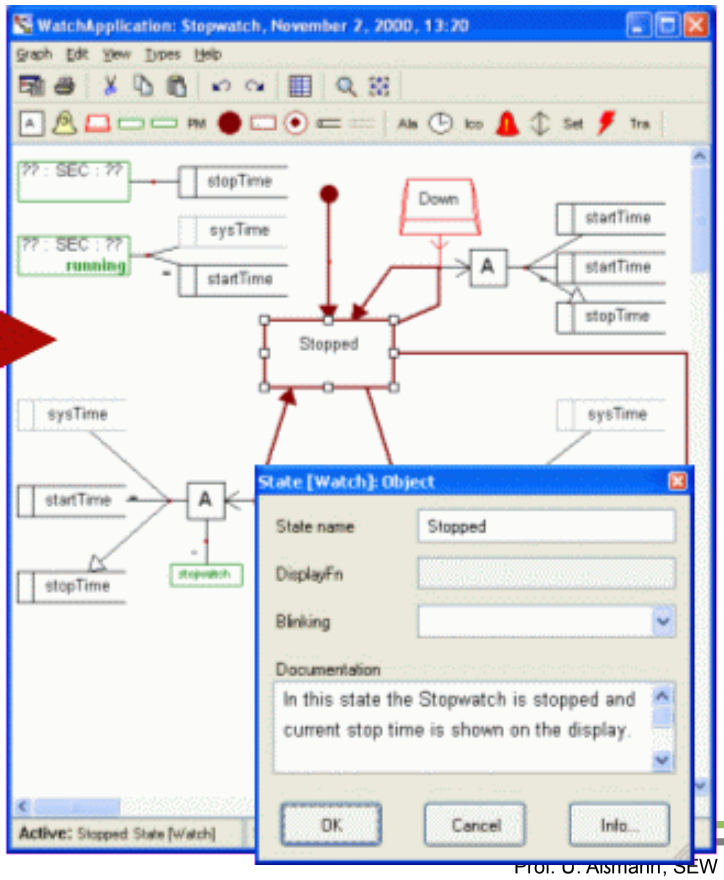
Benutzen der eigenen Methode



Quelle: http://www.metacase.com/mwb30index.html

MetaEdit+ Workbench für ein State Diagram (STD)

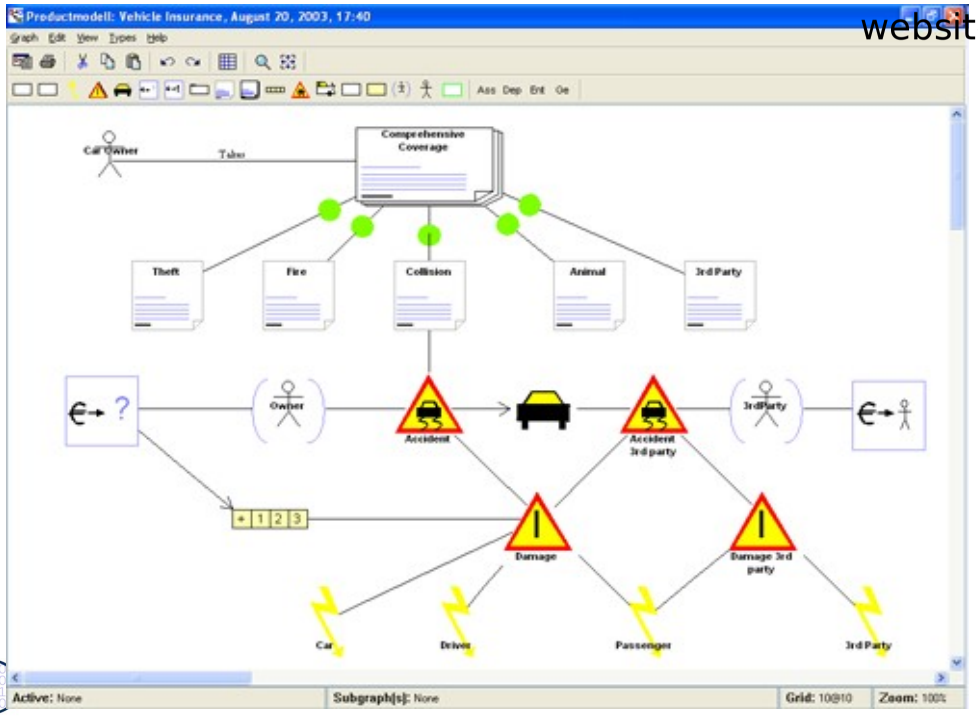




Quelle: http://www.metacase.com/mwb30index.html

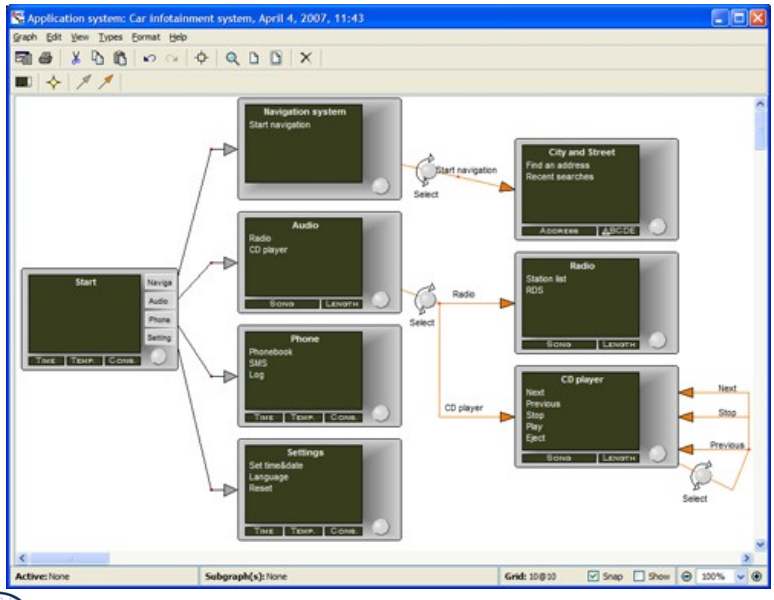
Insurance DSL

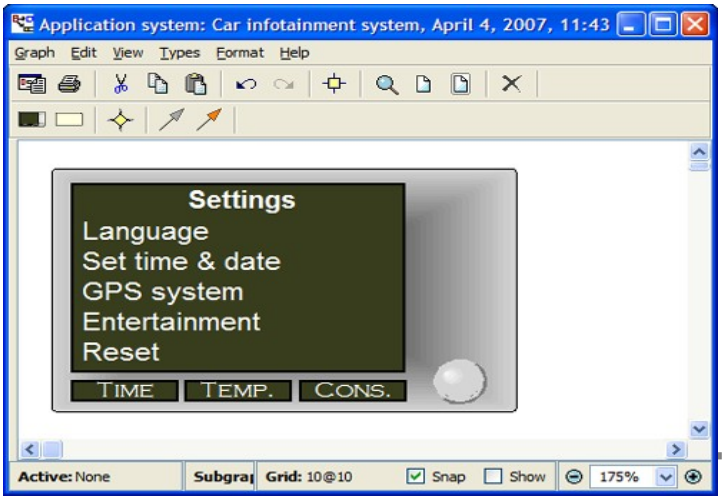
- For modeling of insurance products
- Generators produce the required insurance data and code for a J2EE website



Automotive Entertainment DSL

- Domain: car infotainment system and user interface elements
- Design of the logic and flow via connecting the modeling concepts between GUI and application concept metamodel editor





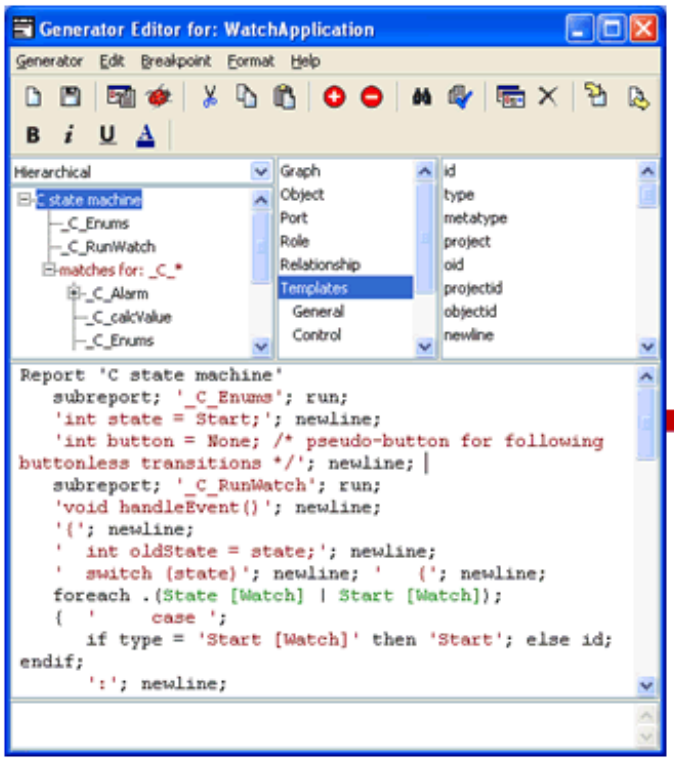
http://www.metacase.com/cases/autoinfo.html

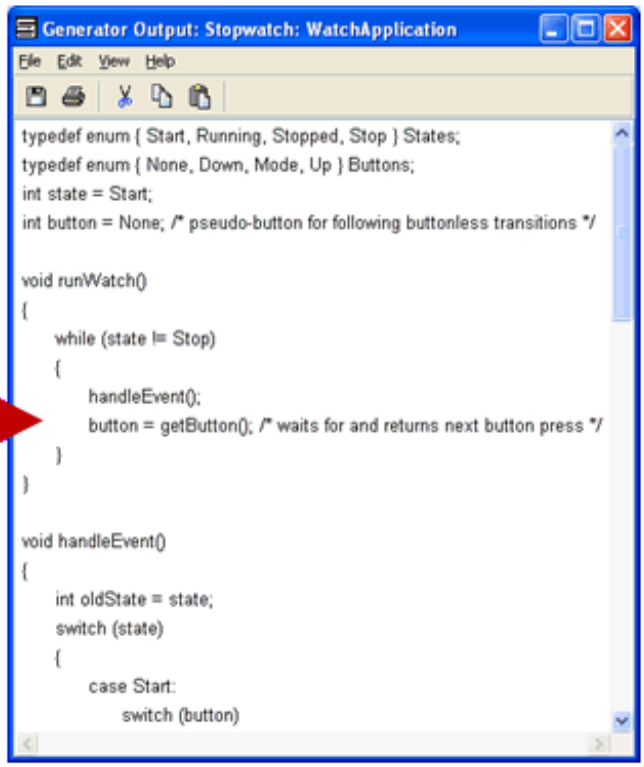
Werkzeuge in MetaEdit+

- Report Generator:
  - Skriptgesteuert, zur Erzeugung von Texten und Code
- API (API-Server):
  - MetaEdit+ ist in Smalltalk implementiert
  - Zugreifbar über Web Server (SOAP mit WSDL)

```
Report 'ExportToolUIModel'
'<?xml version="1.0" encoding="UTF-8"?>'newline;
'<model>'newline;
foreach .Graph {
  do :Graph {
    if type; = 'Tools UIs Model' then
       subreport; 'ToolUI_XML' run;
    else
       subreport; 'structureXML' run;
    endif
  }
}
'</model>'newline;
endreport
```

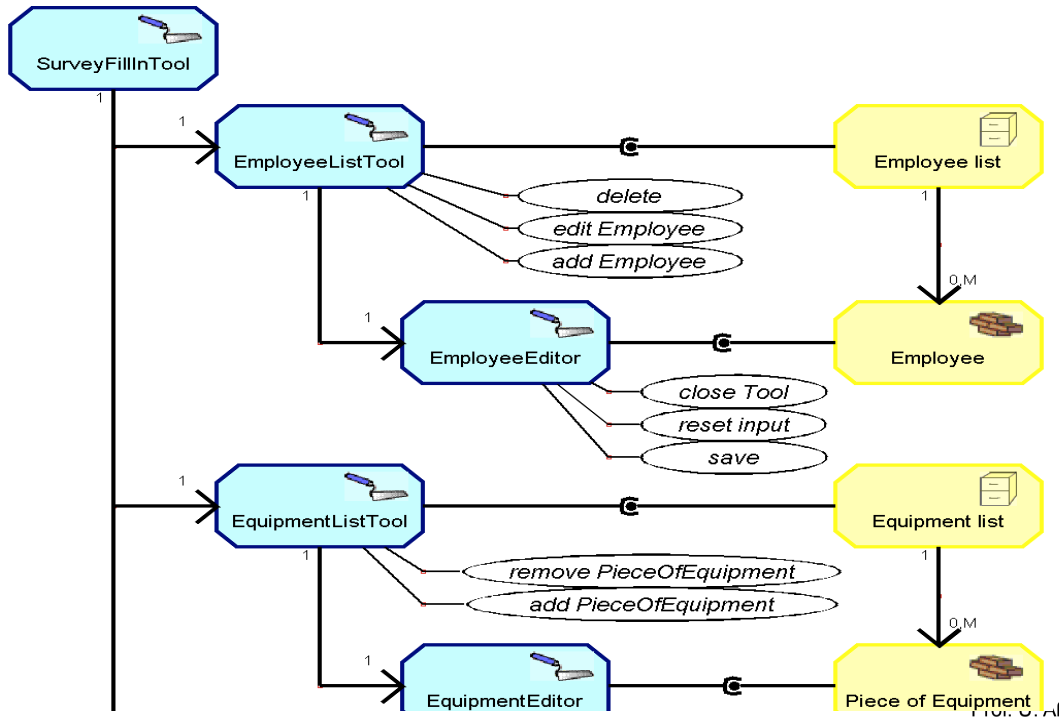
[Nill]



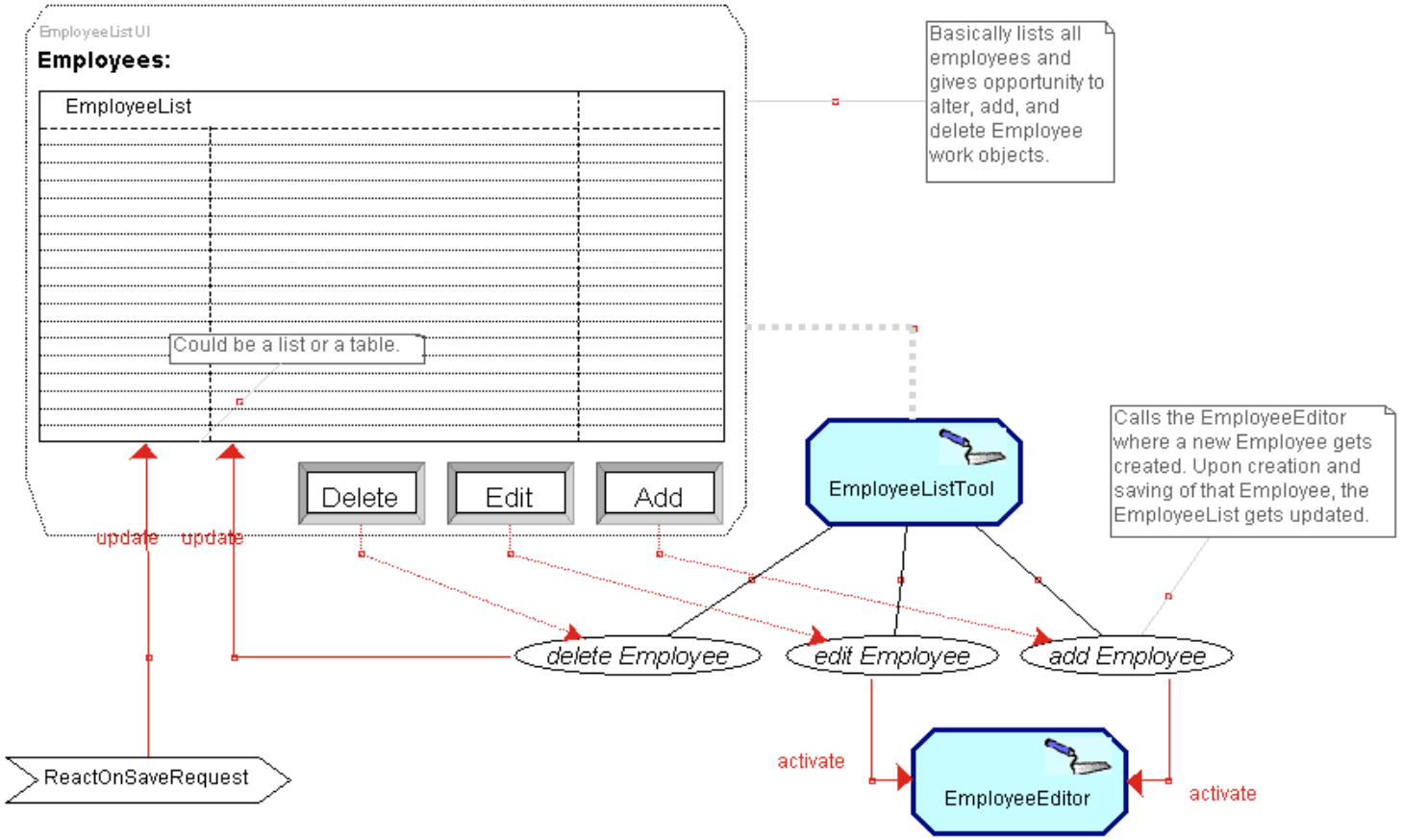


Tool/Material DSL, Modeled in MetaEdit+

- [Nill] präsentiert eine TAM-DSL, modelliert in MetaEdit+
- Editor erlaubt generische Darstellung der Konzepte der DSL



Verbindung GUI – Tool/Material DSL



25.3 Das MOFLON MetaCase-Werkzeug

Courtesy Florian Heidenreich

MOFLON Website

http://www.moflon.org

MOFLON Training

http://moflon.org/documentation/links.html

MOFLON Tutorial

http://moflon.org/documentation/tutorial.html

Einführung

MOFLON ist ein Metamodellierungswerkzeug entwickelt an der TU Darmstadt in der Fachgruppe Echtzeitsysteme von Prof. Andy Schürr

Es unterstützt

- MOF 2.0
- OCL 2.0
- JMI 1.4
- XMI 2.1

MOF Meta-Object Facility

MOFLON

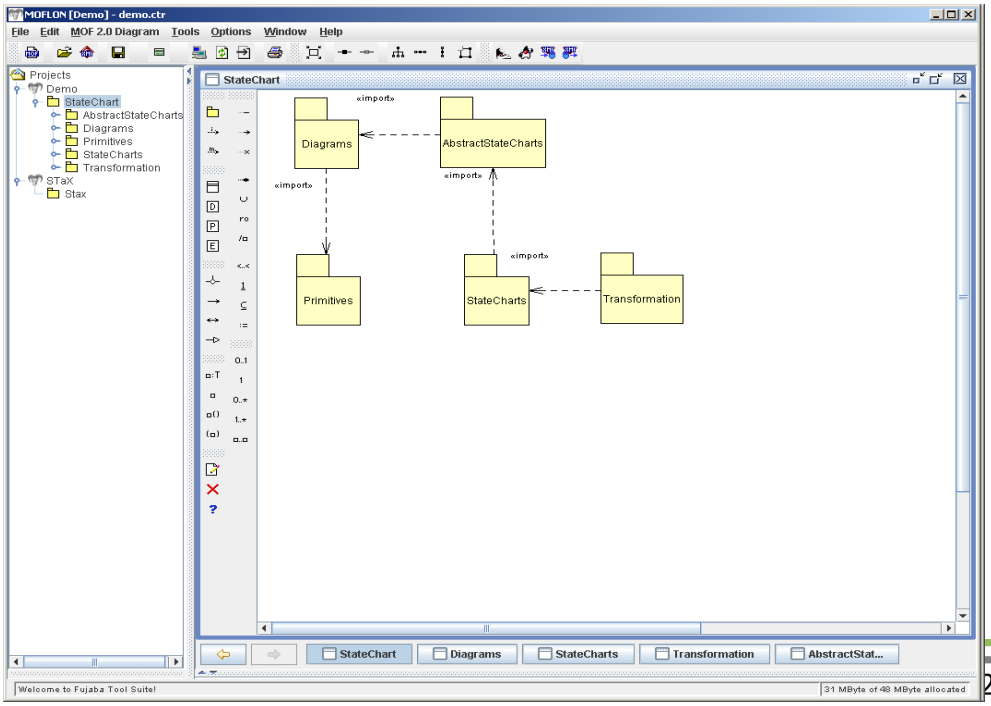
MOFLON 1.0.0 basiert auf der www.fujaba.de tool suite

FUJABA Tool Suite

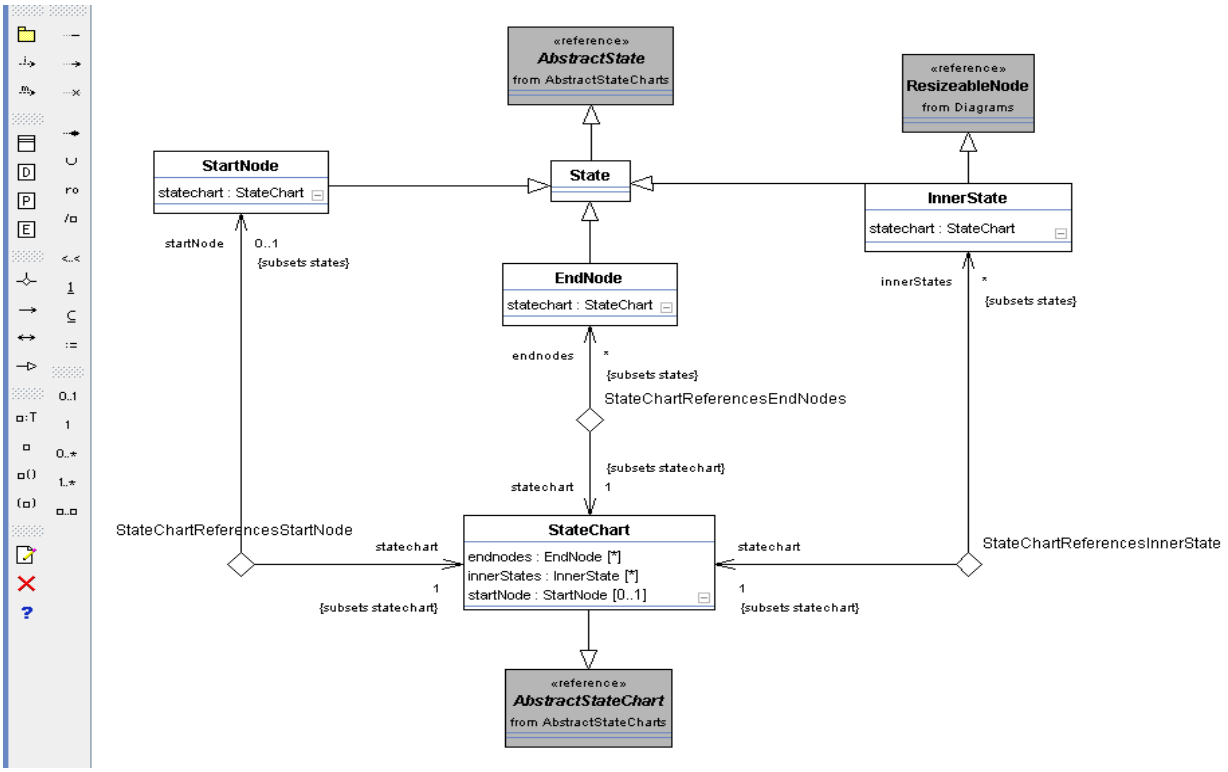
Beispiel: Metamodell für Statecharts: Vorgehensweise

1) Metamodell erstellen
2) Code generieren
3) Code über JMI-Schnittstellen verwenden

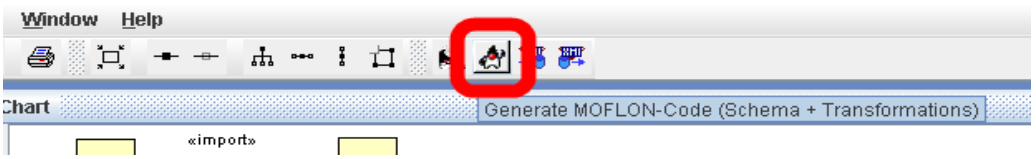
Metamodell für Statecharts



Beispiel: Erstellung eines Metamodells für Statecharts



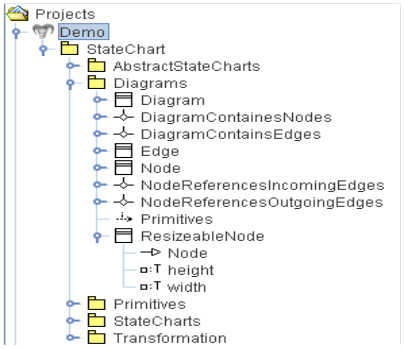
Beispiel: Codegenerierung aus Metamodell für Statechart-Modelle

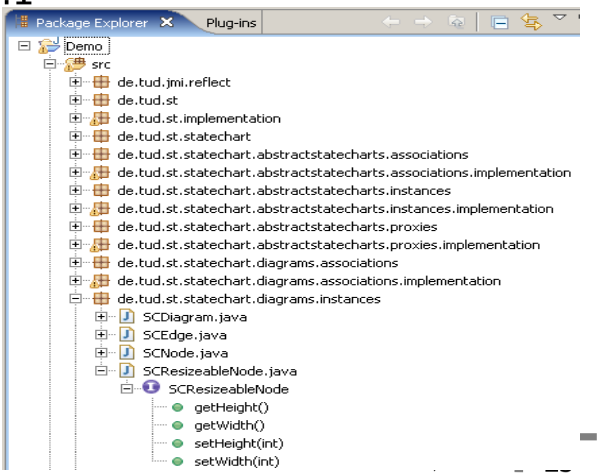


Erzeugt JMI-Schnittstellen zum Metamodell (metamodellgesteuertes Repositorium)

Generiert Code für alle als Story-Diagramm (Fujaba) modellierten Methoden

Codegenerator verwendet Velocity und XSLT 1.1





Beispiel: Codegenerierung aus Metamodell für Statechart-Modelle

Code generieren

Pro Package

- Java Paket
- Schnittstelle
- Implementierung

de.tud.st.statechart
SCStateChartPackage.java
SCStateChartPackageImpl.java

Pro Klasse

- Schnittstelle
- Implementierung
- Proxy Schnittstelle
- Proxy Implementierung

SCNode.java
SCNodeImpl.java
SCNodeClass.java
SCNodeClassImpl.java

Pro Assoziation

- Schnittstelle
- Implementierung

SCDiagramContainsEdges.java
SCDiagramContainsEdgesImpl.java

Beispiel: Codeverwendung von Statechart-Modellen

Code verwenden

- Wurzelpaket instanzieren

```
SCStateChartPackage root = new SCStateChartPackageImpl();
```

- Proxy anfordern

```
root.getSCDiagramsPackage().getSCNode();
```

- Über den Proxy Instanzen erzeugen

```
SCNode node = root.getSCDiagramsPackage().getSCNode().createSCNode();
```

The End