CSSE 490 Model-Based Software Engineering: More Software Factories

Shawn Bohner

Office: Moench Room F212

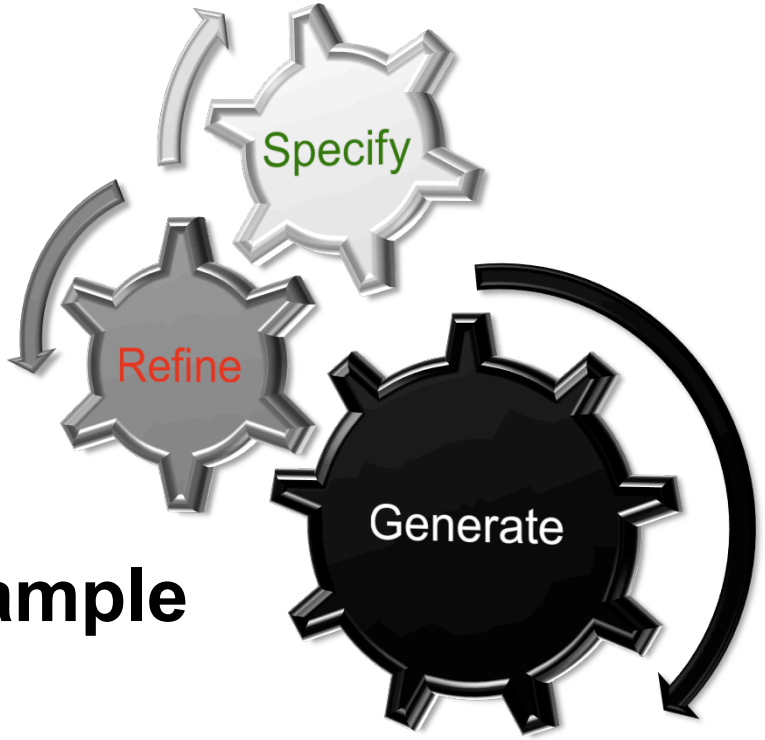
Phone: (812) 877-8685

Email: bohner@rose-hulman.edu

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Learning Outcomes: MBE Discipline

Relate Model-Based Engineering as an engineering discipline.



- Finish Software Factories
- Examine Executable UML
- Short Action Language Example
- Recipe Framework for Manual Code Inclusion

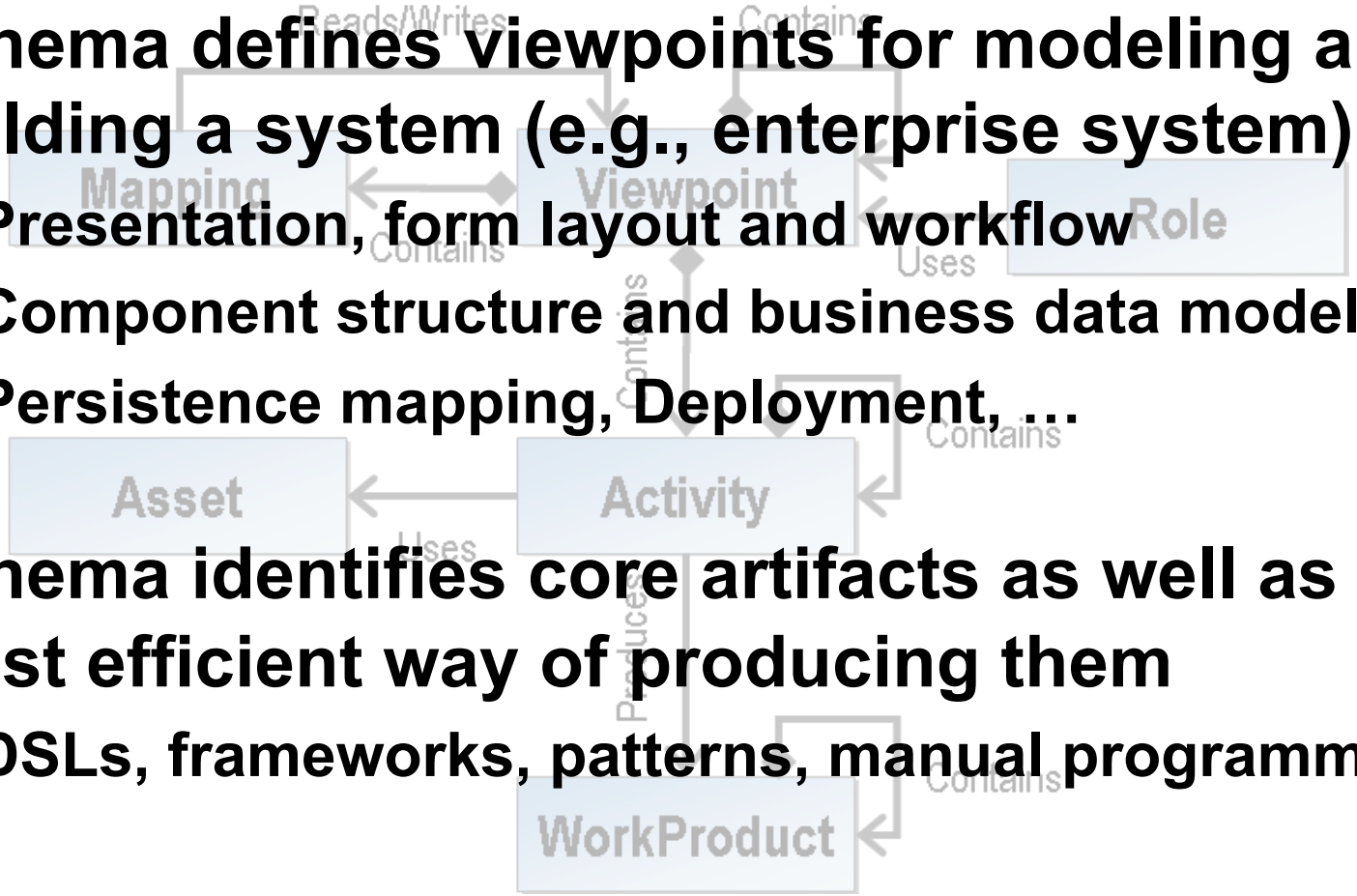
Recall: Software Factories

- 1990's Software Factories emerged as the new automated programming
- Faced an untrained community coupled with limitations in computing capabilities
  - The Virtual Software Factory
  - Software Templates
  - Software Refinery

- Devolved into IDEs configured for efficient development of Domain applications (led by Microsoft these days)

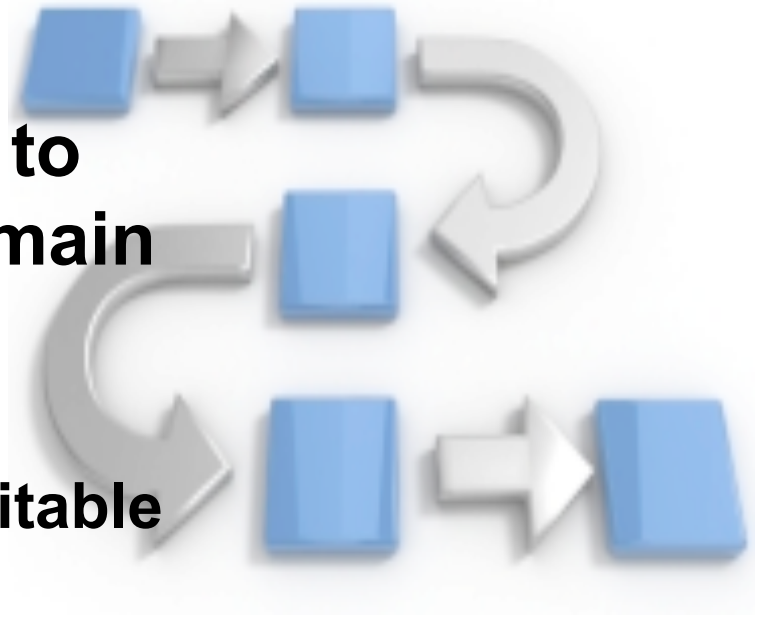
Recall: Software Factories Schema

- Schema defines viewpoints for modeling and building a system (e.g., enterprise system):
  - Presentation, form layout and workflow
  - Component structure and business data model
  - Persistence mapping, Deployment, …
- Schema identifies core artifacts as well as the most efficient way of producing them
  - DSLs, frameworks, patterns, manual programming
- Schema identifies commonalities and differences among applications in the domain



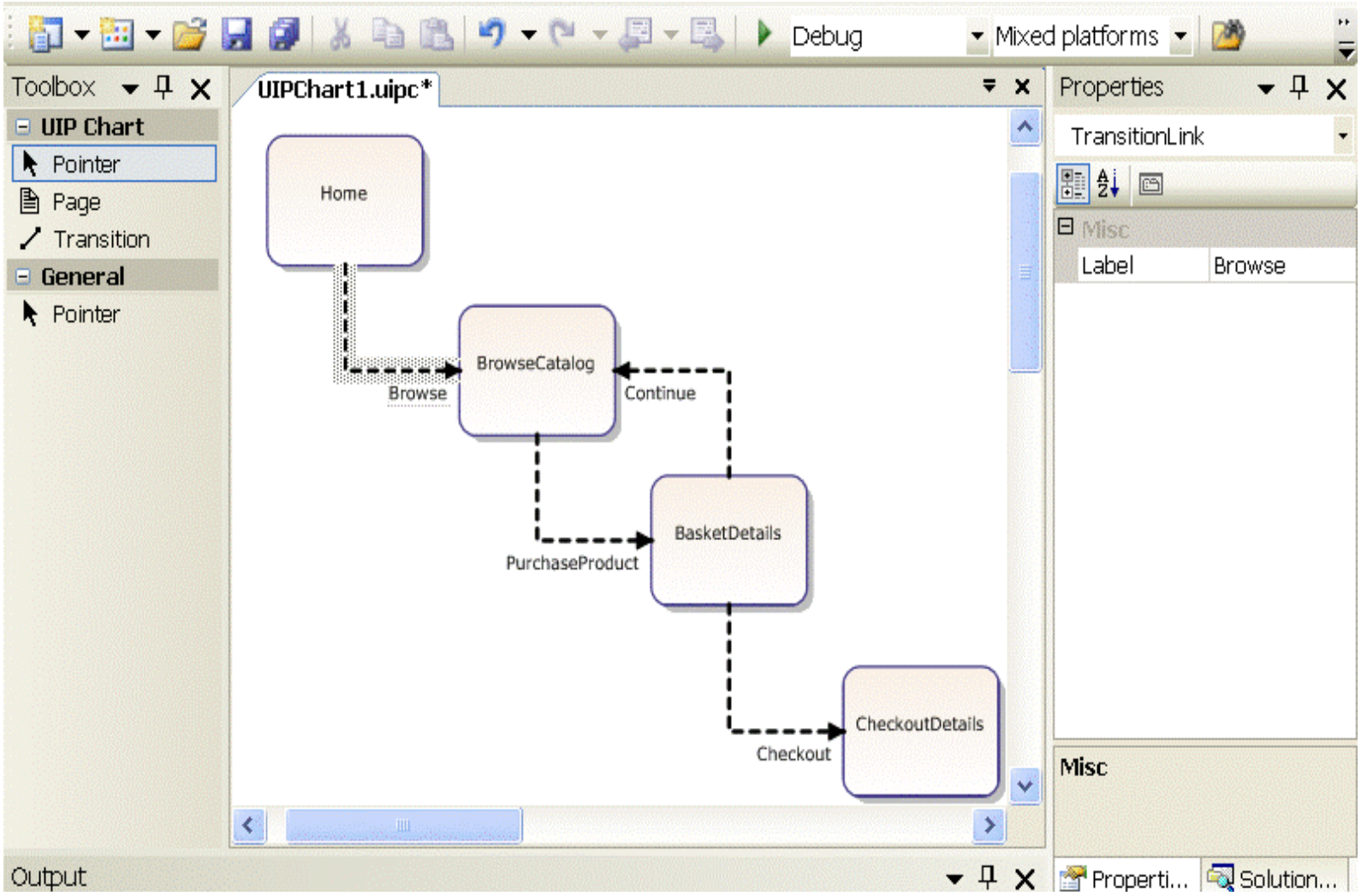
Software Factories Templates

- Makes the Schema usable



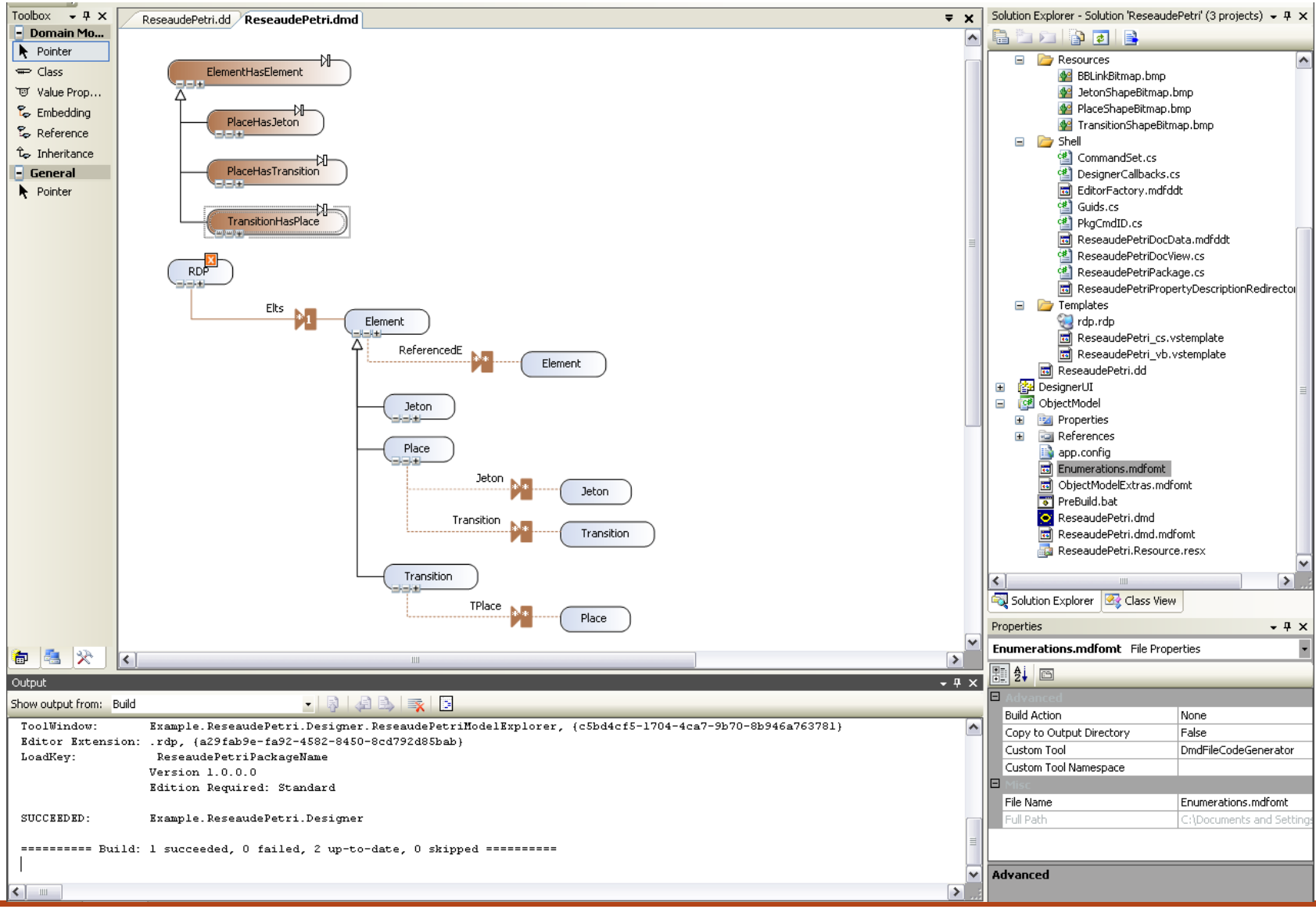
- Load SF Template into IDE to configure it for specific domain
  - Provides the necessary frameworks or libraries
  - Contributes project types suitable for the factory
  - Delivers build scripts
  - Extends IDE with DSL editors and transformations

MS DSL Tools Example



Source: http://galilee.microsoft.fr/(ncqibzbvkp2ezr45aevbcqjk)/a17fdcfb90f14a7592045f1c0fc5e97f/

Defining a Metamodel



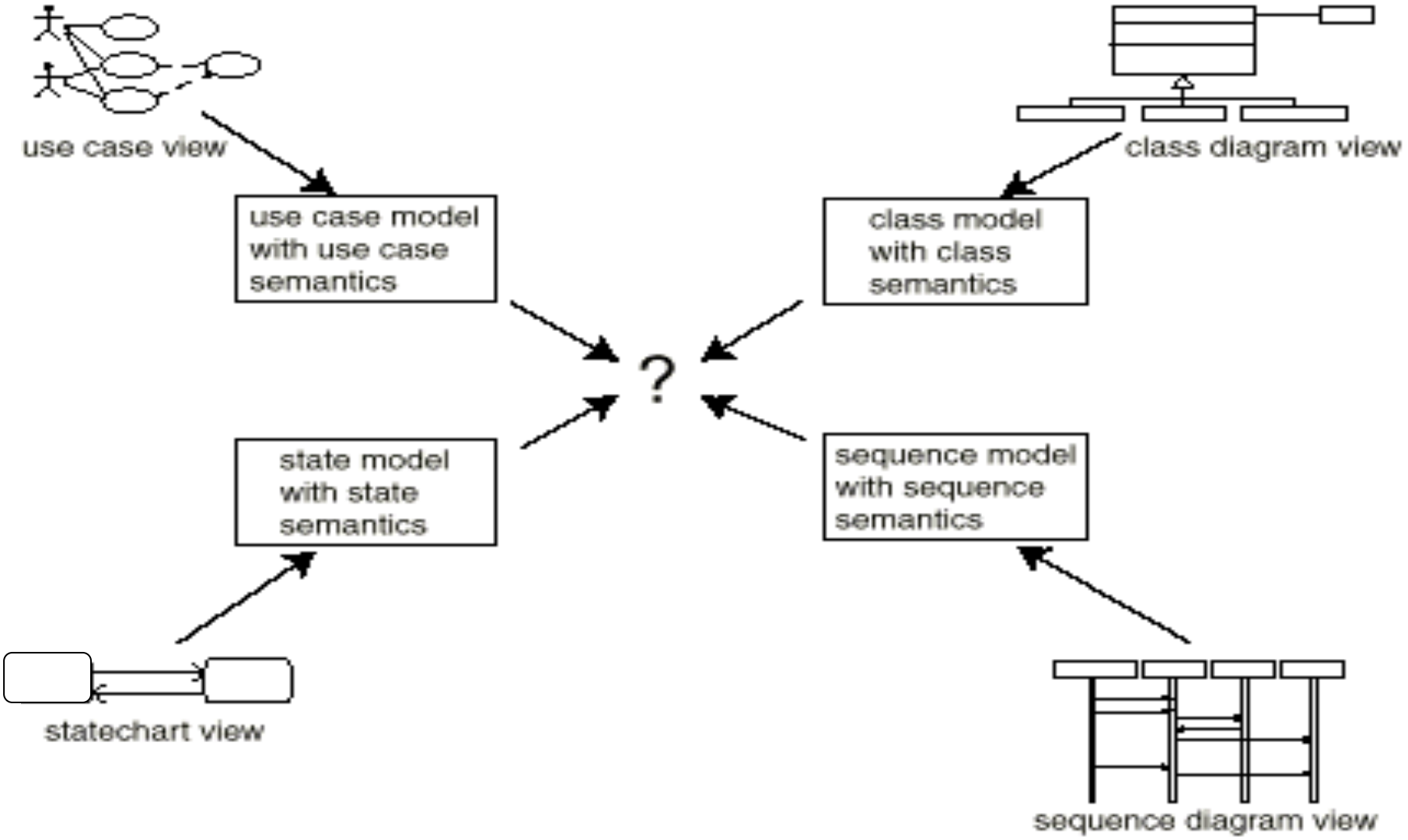
ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Source: http://galilee.microsoft.fr/(ncqibzbvkp2ezr45aevbcqjk)/a17fdcfb90f14a7592045f1c0fc5e97f/

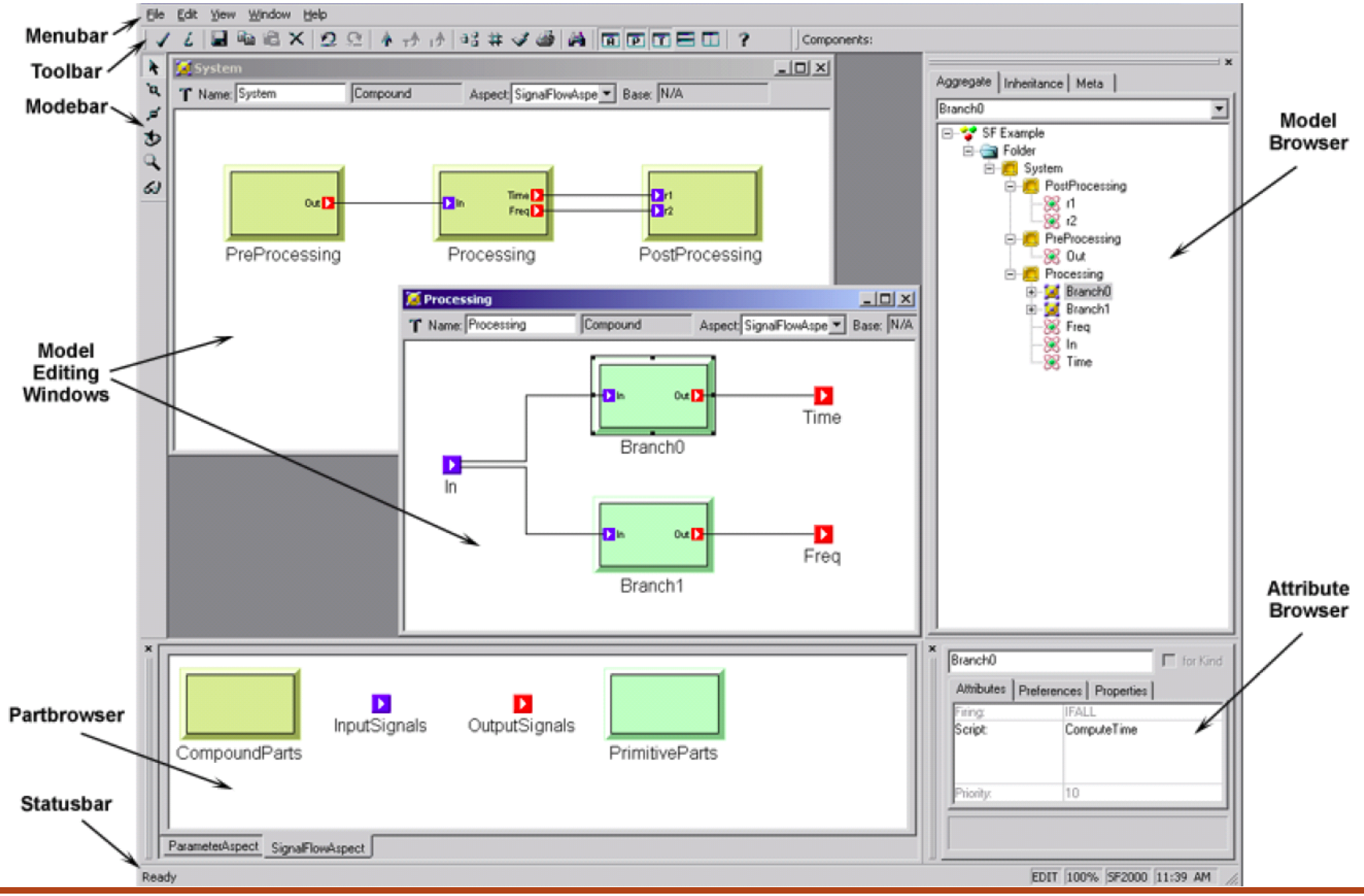
Software Factory and MBSE's

- SFs use model-based concepts without major changes
  - DSLs are used to build models, Languages often graphical
  - Some provide tooling to define the metamodels as well as concrete syntax and editors
- SFs seldom use OMG standards for their infrastructure
  - DSLs are not UML based
  - Metamodels are not based on the MOF, and not QVT
- Application developer's perspective
  - Models are first class artifacts in development projects
  - Editors and transformations integrate seamlessly with the IDE
- Infrastructure developer's perspective
  - Metamodels, editor definitions and transformations are first class artifacts
  - Tools to build them are seamlessly integrated into the IDE

Same Semantics for Different Views

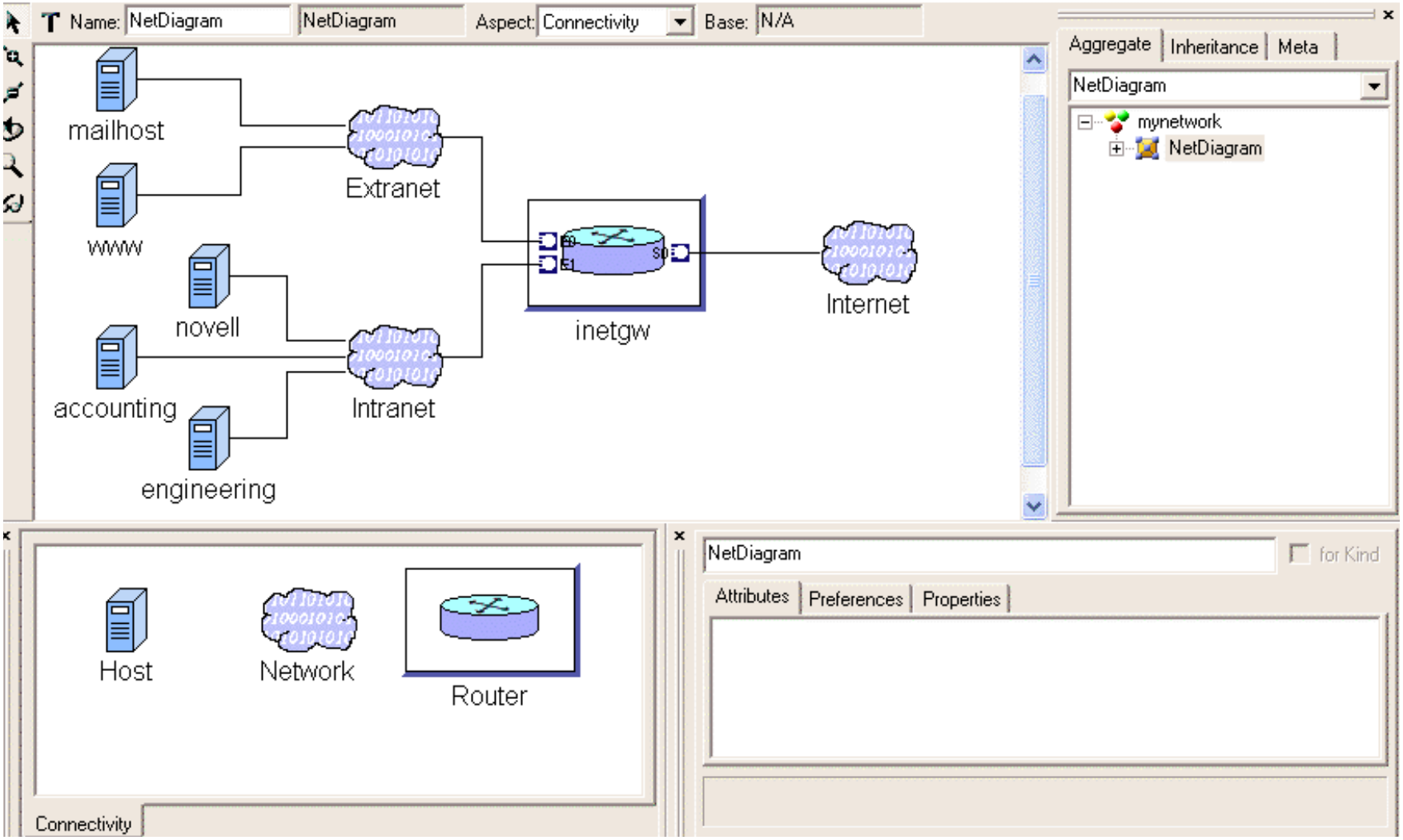


Graphical Model Editor

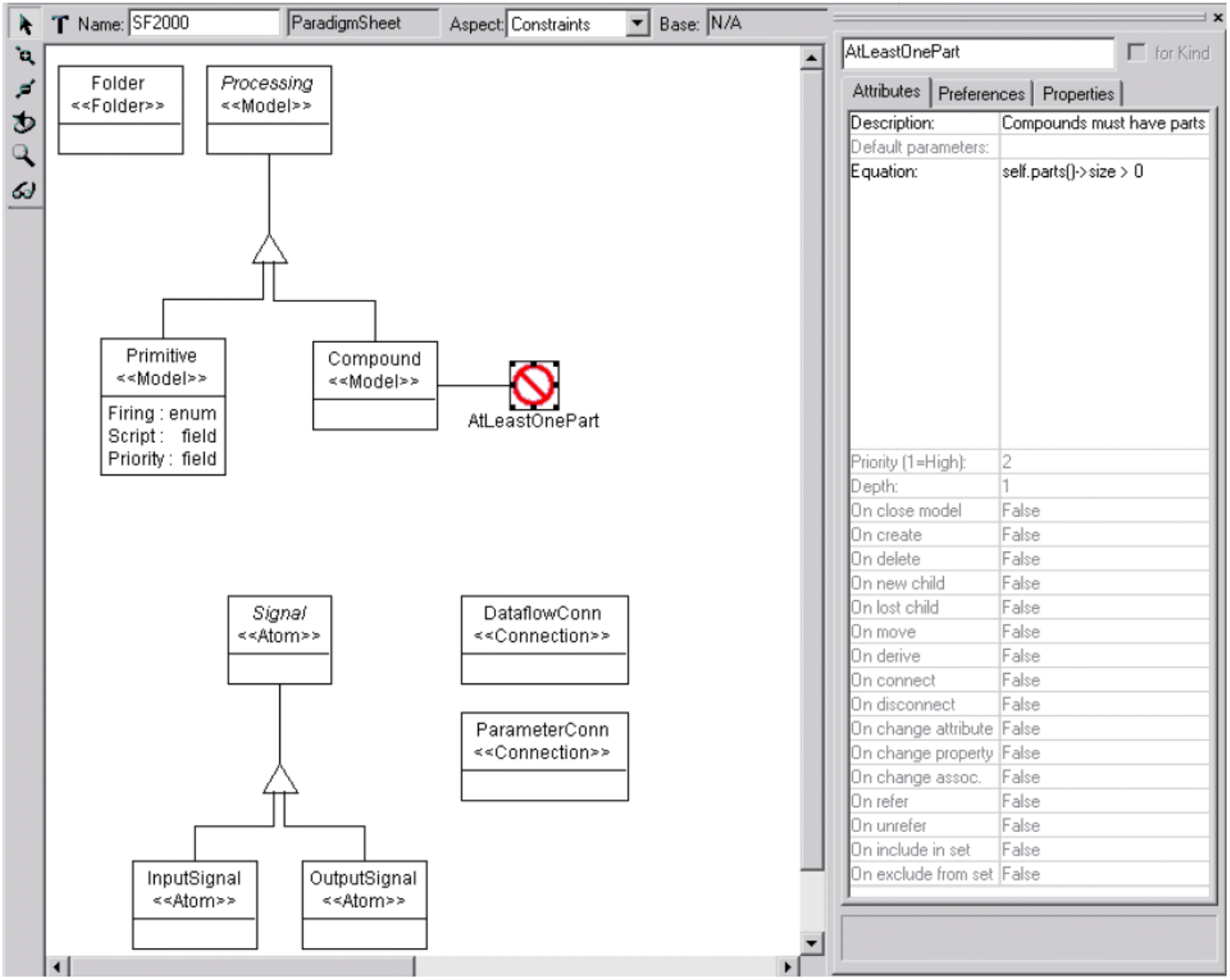


ROSE-HULMAN INSTITUTE OF TECHNOLOGY

GME: Modeling based on previously defined Metamodel

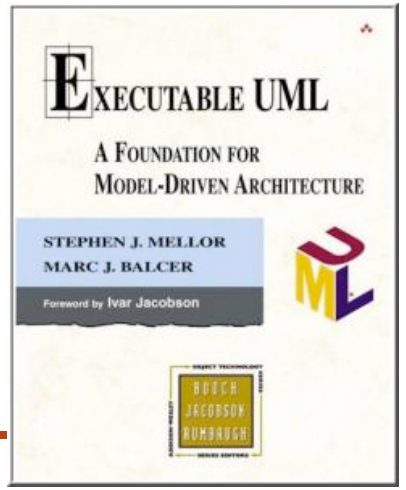


GME: OCL Constraint Validation



Executable UML (xUML) Concepts

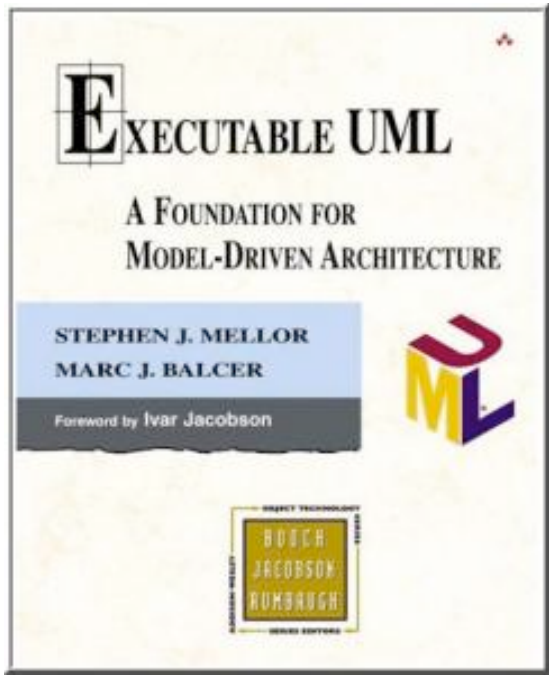
- Executable UML is not a formal standard, but a goal for a UML-based programming language
- Must eliminate redundancy and ambiguities, to increase executability of UML
- Action language needed to define complete implementations of software systems
- Not a DSL, but rather a universal, UML-based programming language



Executable UML → Action Semantics

- Hard to model a complete system today via UML or even MOF-based languages

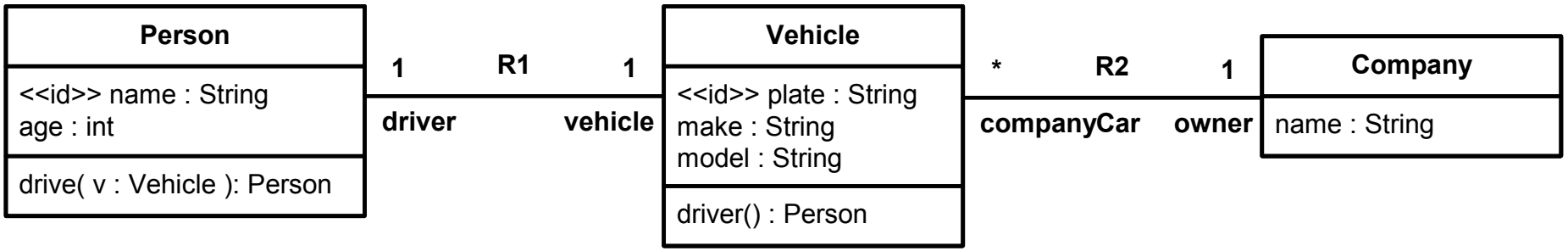
- Action semantics do not contain structural constructs (classes, attributes & relationships)
  - Already defined in the structural part of the model
  - Merely define behavioral building blocks



Action Semantics in UML 2.0

- Models procedural behavior via abstract syntax
- Variables for assigning/reading (sets, bags…)
- Arithmetic and logical operations
- Control flow (if-then-else, case, block…)
- Class extents that be queried (SQL-like)
- Creation, deletion, and navigation of associations
- Generation of signals and timers
- Definition of functions

Action Languages Example 1/3




```
myJeep = create Vehicle with plate = "IYQ2"
myJeep.make = "Chrysler Jeep"
myJeep.model = "Liberty CRD"
```

Action Languages Example 2/3

```
shawn = create Person with name = “Shawn”
```

We can now call the operation drive() to let the driver drive the vehicle.

```
[actualDriver] = drive[aVehicle] on shawn
```

What is still missing, of course, is the implementation of the operation drive(). The least it must do is to instantiate the association R1 (that is, to create a link between the two concerned objects).

```
link this R1 aVehicle
```

Action Languages Example 3/3

```
theCurrentDriver = this.R1.“driver“
```

Let’s assume we want to find all people in the system:

```
{allPersons} = find-all Person
```

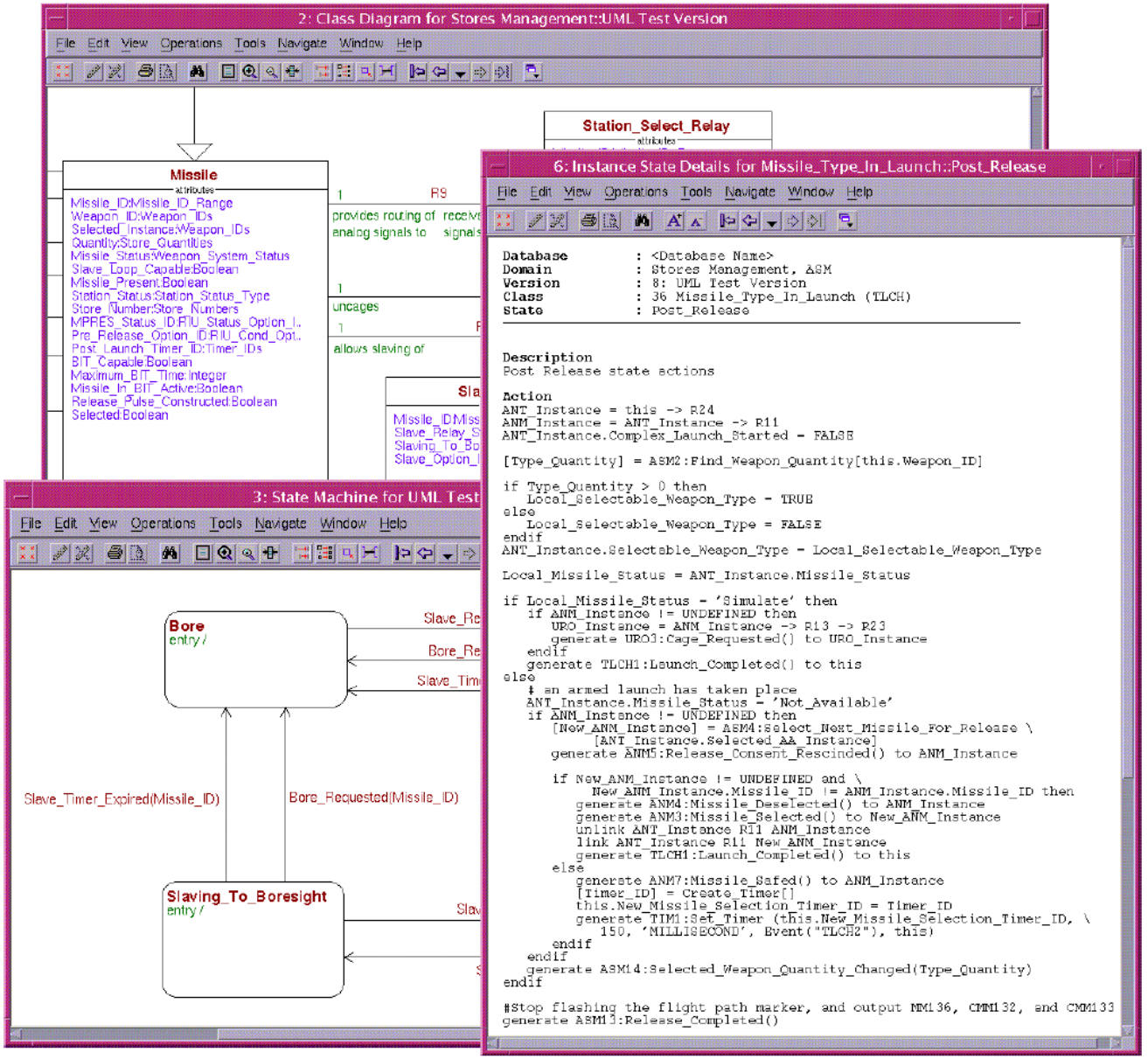
The braces state that allPersons is a set of objects instead of just one.

One can also limit such a search. For example, all vehicles of the brand Jeep can be looked for.

```
{Jeeps} = find Vehicle where make = “Jeep”
```

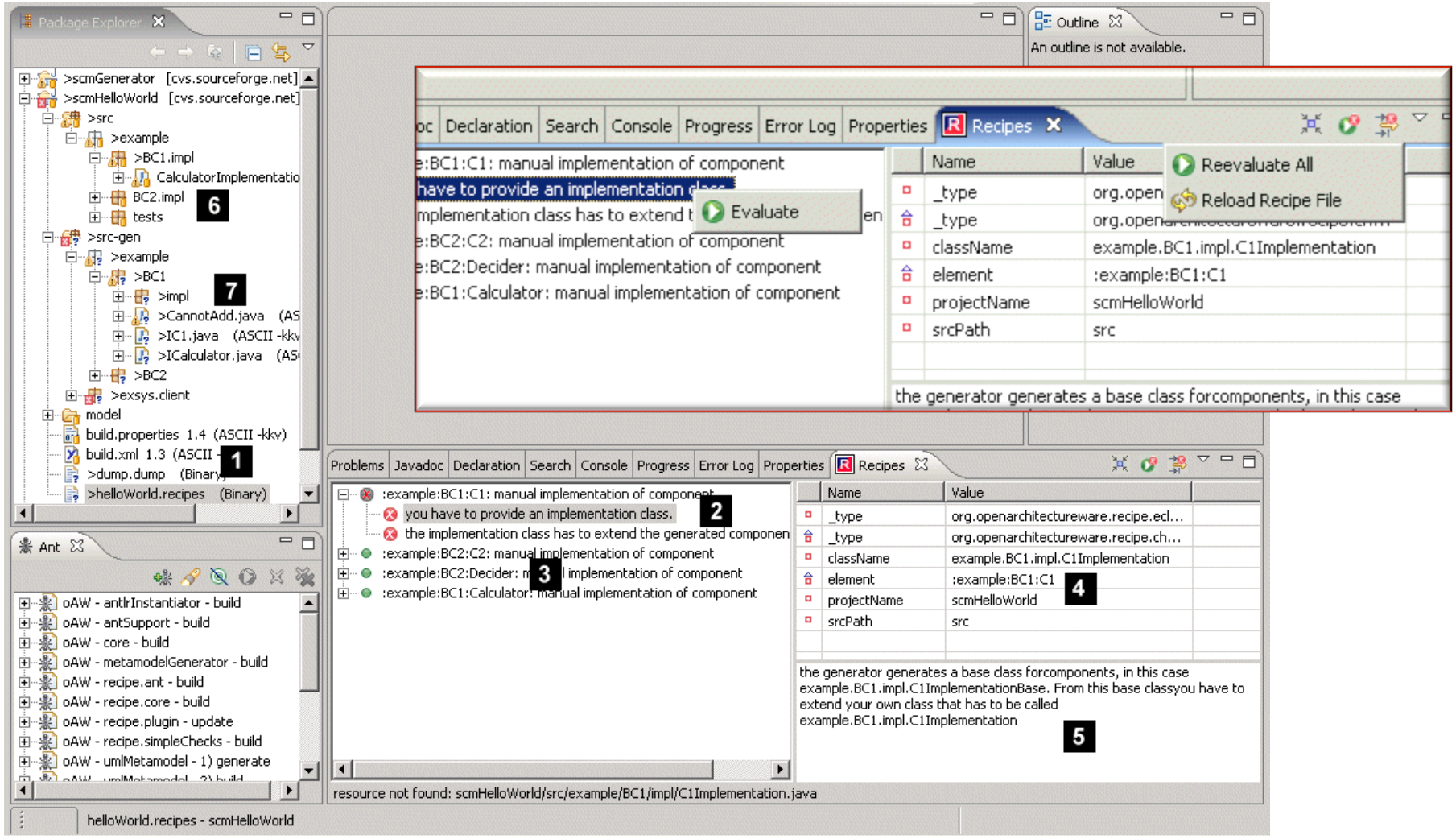
Example: Kennedy Carter's iUML

- Model Diagrams
- Code
- Integration



ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Recipe Framework for Integrating Manually Developed Code



ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Homework and Milestone Reminders

- Milestone 3/4: Final MBSE Environment (see Milestone 3/4 assignments)
  - Due by 11:55pm, Friday, May 13th, 2011.
- Term Paper and Presentation
  - Paper Due by 11:55pm, Tuesday, May 17th, 2011.
  - Presentation Due by 1:35pm, Thursday, May 19th, 2011.