

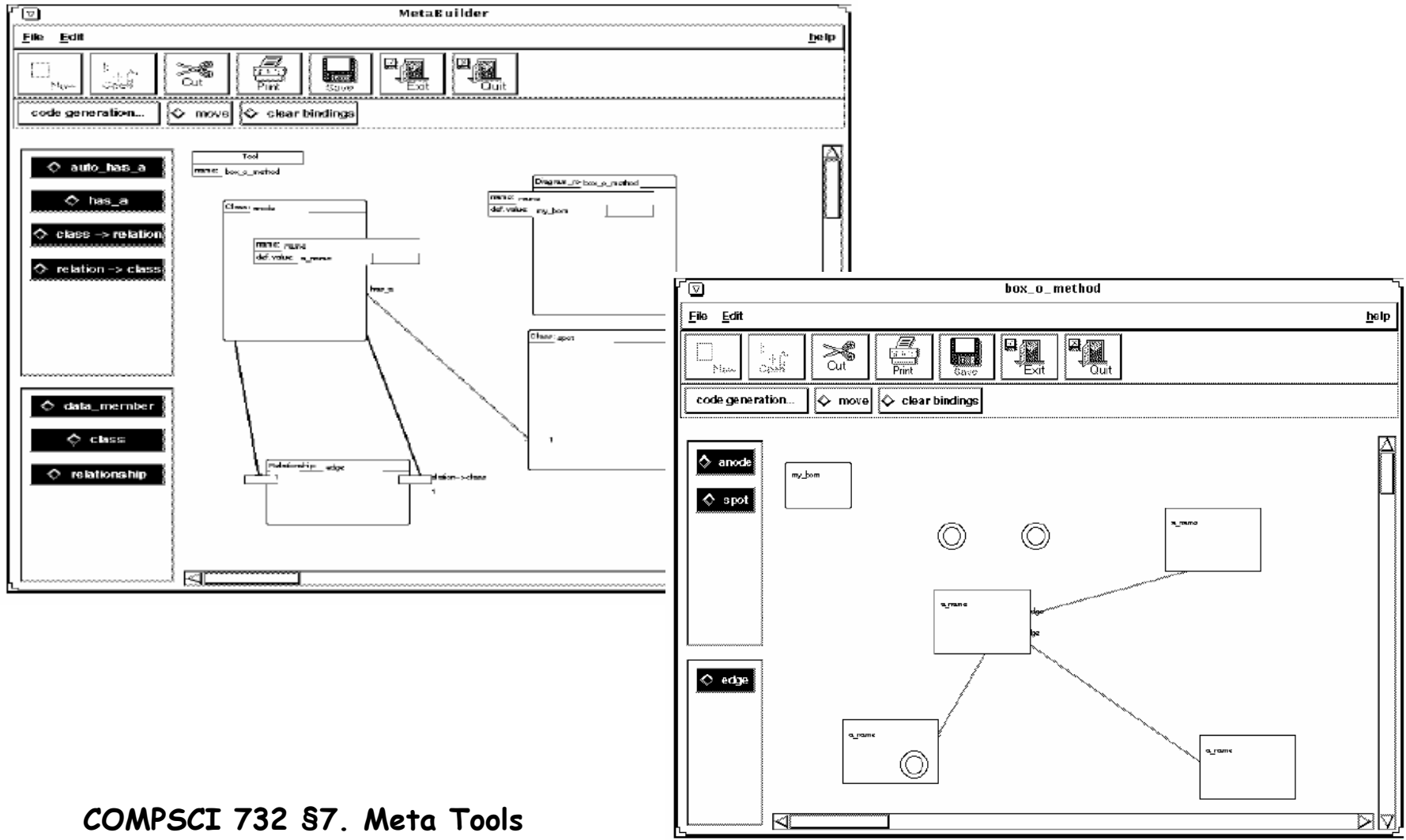
Other Meta Tools

- **Aim of section:**
 - Examine other meta tools compare and contrast to Marama/Pounamu
- **Contents**
 - MetaBuilder
 - MetaEdit+
 - GME
 - Microsoft DSL Tools
 - IPSEN
 - Comparison

MetaBuilder

- Ferguson et al, U of Sunderland (now @ Strathclyde)
- Visual constructor tool for the MetaMOOSE framework
- Metamodel tool
 - Classes, with data and function members
 - Function members provide behaviour (using Itcl)
 - Relations
 - Source, sink, cardinality constraints
 - Has-a for aggregation, Inheritance
 - CF Pounamu Entities and relationships
- Symbol editor
 - Widgets of various sorts, implemented in Itcl

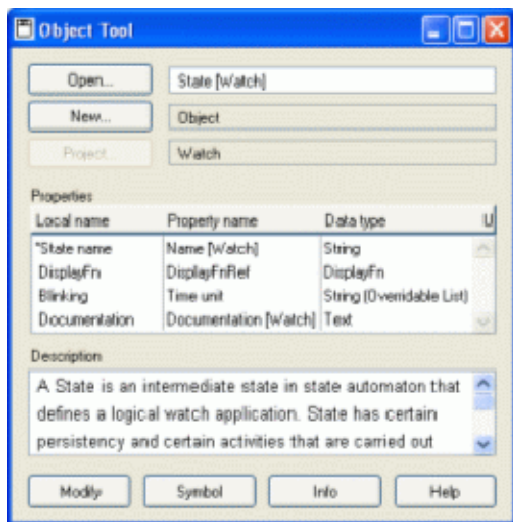
MetaBuilder



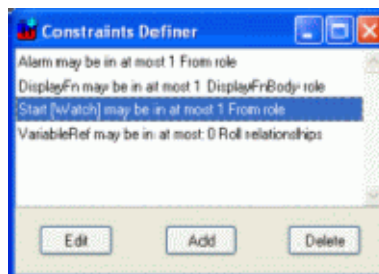
MetaEdit+

- Commercial system from MetaCASE (cost E11,500) www.metacase.com
 - (ex MetaEdit from U Jyvaskyla Finland)
- Variety of text/form based tools to specify meta model
 - Objects
 - Properties (attributes)
 - Relationships and Roles (endpoints)
 - Ports (constraints on connection points)
 - Graph (like Pounamu view tool)
- Symbol and Dialog Box Editors
- Reports and generators (walk data structures to generate reports, code)
- External interfaces
- Model editors include diagrams, matrices, tables, browsers

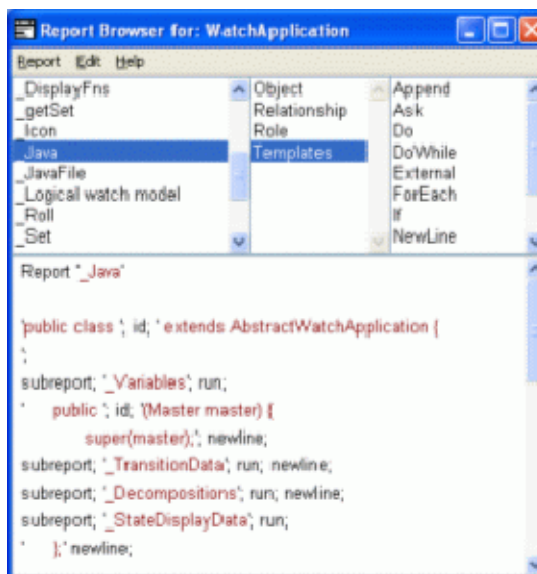
MetaEdit+



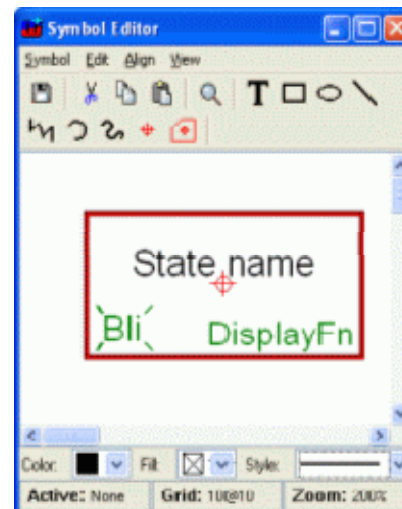
Object Tool



Constraints



Symbol editor



Generator

MetaEdit+ Generated System

The screenshot shows the MetaEdit+ interface for a stopwatch application. The state machine diagram includes states: Stopped, Running, and Up. Transitions are labeled 'Down' and 'Up'. The diagram also shows variables: startTime, stopTime, and sysTime. The generated code in the 'Report Output' window is as follows:

```
public Stopwatch(Master master) {  
    super(master);  
    addTransition ("Start [Watch]", "", 0, "Stopped");  
    addTransition ("Running", "Up", a22_718, "Stopped");  
    addTransition ("Stopped", "Mode", 0, "Stop [Watch]");  
    addTransition ("Stopped", "Up", a22_1077, "Running");  
    addTransition ("Running", "Mode", 0, "Stop [Watch]");  
}
```

The variable declaration table in the 'C:\MetaEdit\MetaEdit\MWB 3.0\Reports\Stop...' window is as follows:

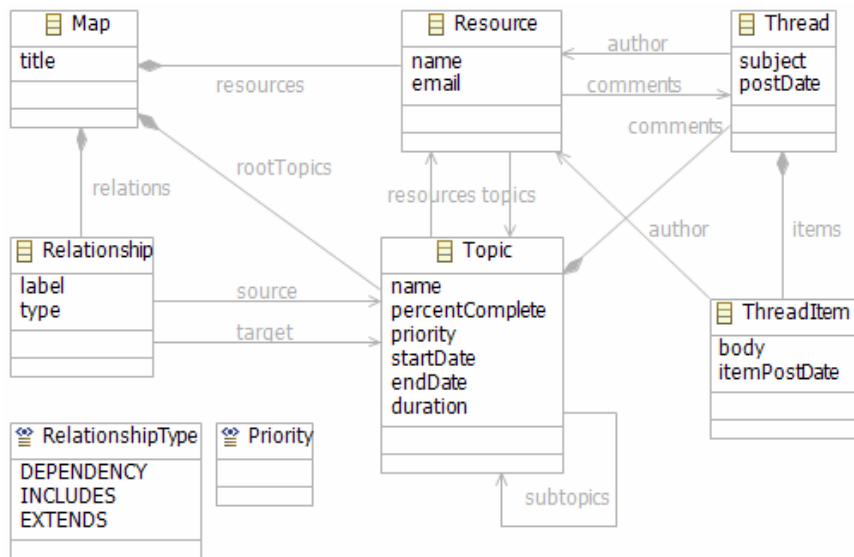
<u>startTime</u>	Variable	Variable that stores the start time of the running stopwatch.
<u>stopTime</u>	Variable	Variable that stores the current stop time: how many seconds had elapsed when the stopwatch was stopped.

GME

- **Generic Modelling Environment, Ledeczi et al, Vanderbilt**
- **<http://www.isis.vanderbilt.edu/Projects/gme/default.html>**
- **Visual MetaModel composed of several parts**
 - **Class diagram with stereotypes representing metatype**
 - **Metatypes defined by MetaGME meta model**
 - **Atoms, connections, models**
 - **Attributes, constraints**
 - **Constraints represented using OCL**
 - **Visualization**
 - **Like Pounamu view definer - defines *aspects***
 - **Symbols from simple built-in symbols or bitmaps + code for more complex symbols**
- **Extensibility via COM interfaces and XML import/export**

Eclipse GMF

- **GMF = Graphical Modelling Framework**
 - <http://www.eclipse.org/gmf/>
 - part of the Eclipse Model Project
- **Goals very similar to Marama**
- **Provides:**
 - **EMF modelling tool (textual or graphical) for meta model**
 - **Visual notation specification tool (wizard based for simple notations)**
 - **Palette etc specification tool (ditto)**
 - **Mapping tool (textual - like the old Pounamu view specification)**
 - **Code generation specn**



New

Graphical Definition
Specify basic graphical definition of the domain model.

Domain model elements to process:

Element	checkbox	checkbox	checkbox
ThreadItem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
body : String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
itemPostDate : Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
author : Resource	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Topic	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
duration : Float	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
endDate : Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
name : String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
percentComplete : Float	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

GMF Tutorial

- Introduction
- Create a New Project
- The Domain Model
- Create a Graphical Definition

In a graphical definition, you will define figures, nodes, compartments, connections, etc.

Using the wizard, we can get started on our graphical definition by examining our domain model. Use the "Click to Perform" button to get started.

The following steps are required:

- In the wizard, select the

New

Tooling Definition
Specify basic tooling definition of the domain model.

Domain model elements to process:

Element	checkbox	checkbox	checkbox
Relationship	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
source : Topic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
target : Topic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Resource	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
comments : Thread	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
topics : Topic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Thread	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
items : ThreadItem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
author : Resource	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ThreadItem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

GMF Tutorial

- Introduction
- Create a New Project
- The Domain Model
- Create a Graphical Definition
- Create a Tooling Definition**

Our graphical definition for our editor will need tooling to be useful. We will again use a wizard to get started by examining the domain model once more.

The following steps are required:

Resource - mindmap/default.mindmap_diagram - Eclipse SDK

Lucida Grande

Project Expl... Navigator

- mindmap
 - default.mindmap
 - default.mindmap_diagram
 - Simple
 - A Subtopic
 - A topic
 - links

default.mindmap_diagram

Palette

- Select
- Zoom
- Note
- Topic
- TopicSubto...

Tasks Properties

Core

Appearance

Property	Value
Duration	0.0
End Date	
Name	A topic
Percent Complete	0.0
Priority	ZERO
Resources	

Create GMFMap model

Mapping
Map domain model elements

Nodes

Topic (Topic, rootTopics)

Links

subtopics : Topic (TopicSubtopics, <unspecified>)

As node <-->

As link -->

Remove

Restore...

Structure

Element: Topic

Containment: rootTopics

Target Feature:

Visual

Diagram Element: Topic

Edit

Change...

Constraints

Specialization: Initializer:

Click to Skip

Code Generation

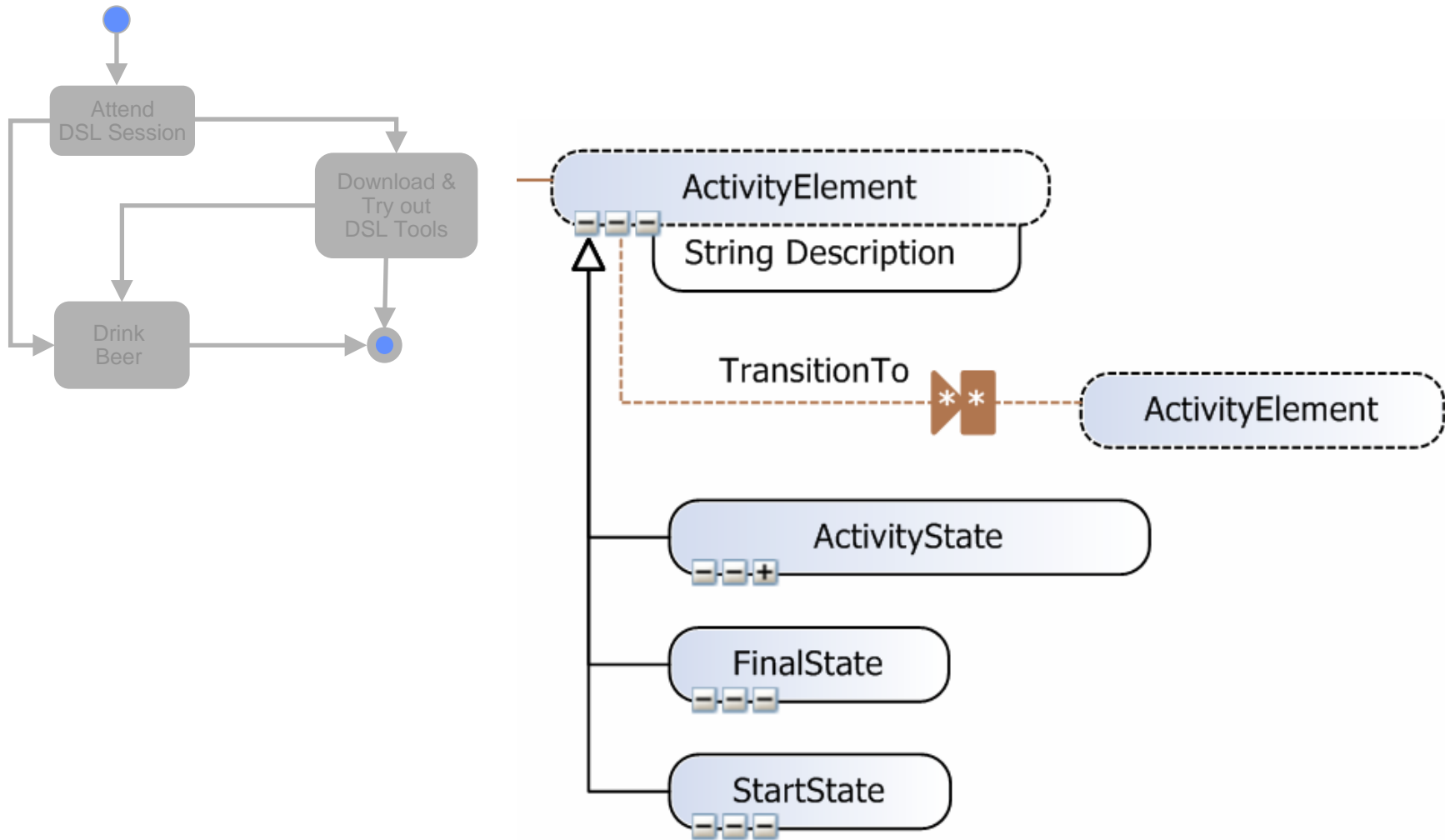
Running the Diagram

< Back Next > Finish Cancel

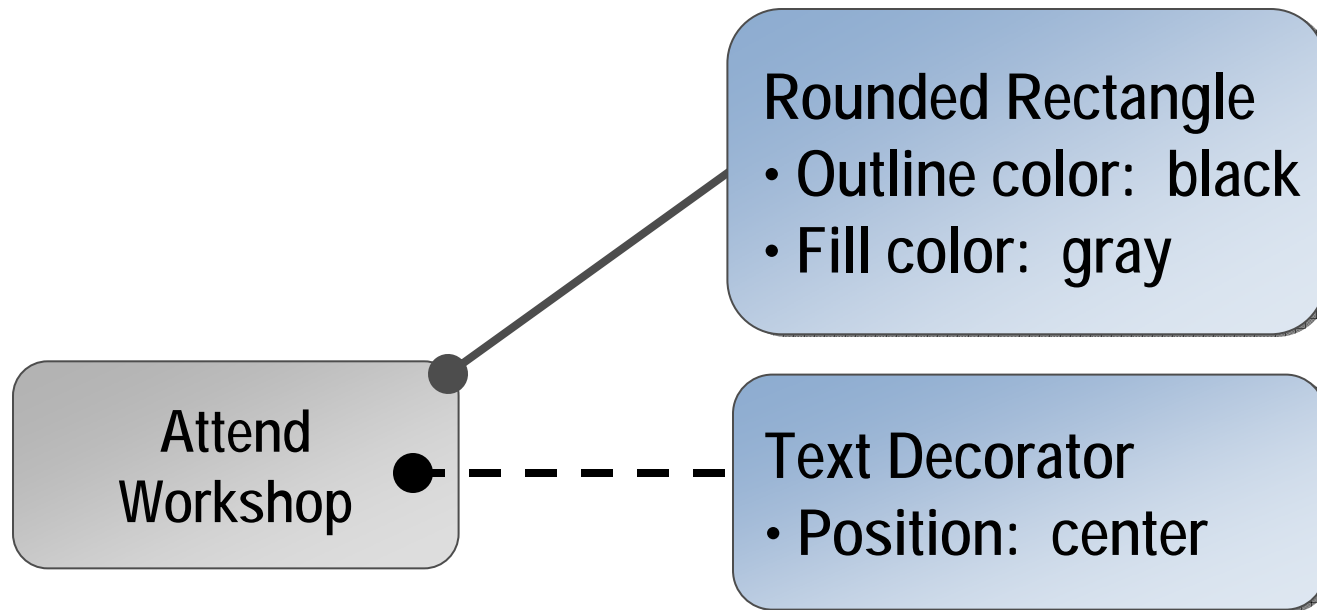
Microsoft DSL Tools

- Extension to Visual Studio 2005
- Provides tools/notations for:
 - Meta model specification (visual)
 - Shape specification (textual but visual definer being developed)
 - Mapping from shapes to model (textual)
 - Code gen using templates (cf Jet etc)
- Multiple views can only be done very awkwardly
- Slide elements from Aali Alikoski, Microsoft Finland, presentation to XP2006

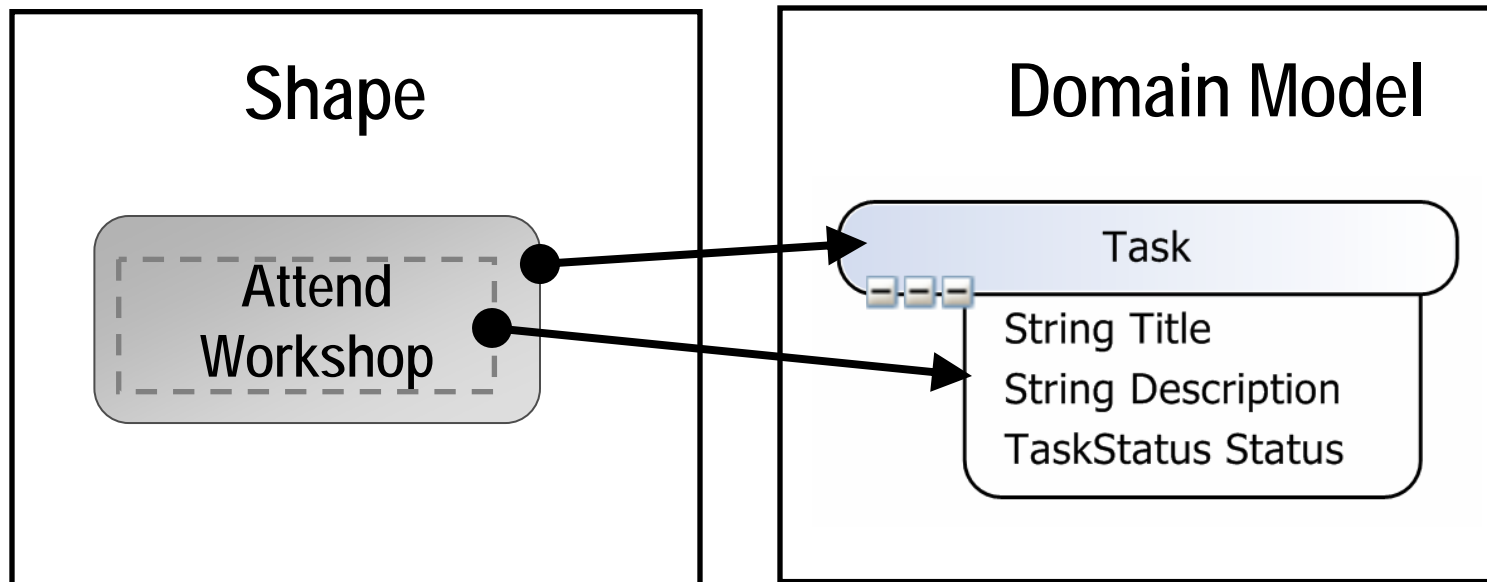
Microsoft DSL Tools - Metamodel



Microsoft DSL Tools - Shapes



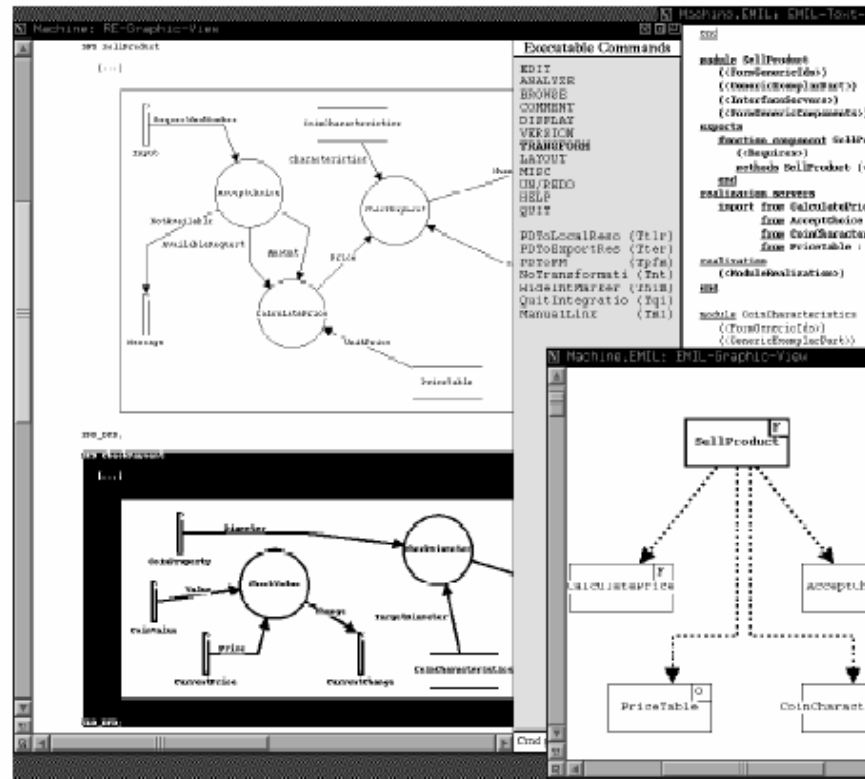
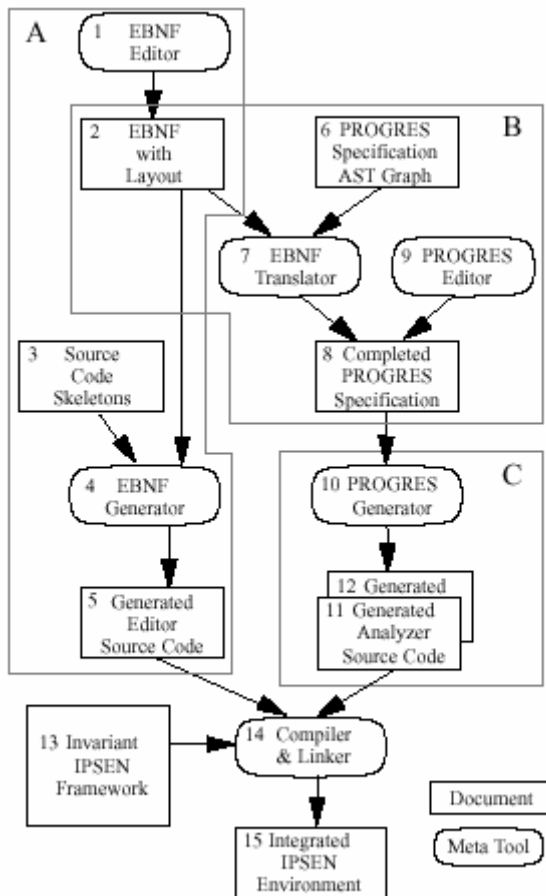
Microsoft DSL Tools - Mapping



IPSEN

- Klein and Schurr, Aachen (Schurr now @ Darmstadt)
 - See SEE'97 paper
- Quite different approach to the other tools
 - Context free grammars used to specify syntax and layout of languages
 - Graph rewriting rules (PROGRESS) used for specifying semantics
 - Both mechanisms use textual specification to generate syntax directed visual editor

IPSEN



Comparison

Tool	MetaModel Paradigm	Meta Model Specn	Visual Elmt Specfn	Behaviour Specfn
MetaEdit+	Unkown (MetaEdit was MOF)	Tabular/ Form based	Symbol Editor	Constraints
Meta Builder	EER/OO based on MOOSE	Visual Editor	Primitives, bitmaps, code	Code
GME	OO based on MetaGME	Visual - several editors	Bitmaps, simple shapes	OCL constraints
GMF	Graph based - EMF	EMF or XSD using text or visual	Wizard	Code
DSL Tools	Graph expressed as tree	Visual Designer	Textual code (visual designer soon)	Code
IPSEN	EBNF and graph grammars	Text	EBNF	Graph Grammars
Marama	Entity Relationship	Visual	Shape & Connector tools	Formulae or Event handlers or Kaitiaki

Comparison

Tool	Storage	Code gen support	Integration API	Multi paradigm
MetaEdit+	Custom DB	Custom scripting language	SOAP	Partially
Meta Builder	OODB	Unknown	Unknown	Unclear
GME	Variety - customisable	Model interpreters	COM interfaces	Yes, aspects
GMF	XML files (XMI based - ex Eclipse)	Jet, etc	Eclipse plugin	No
DSL Tools	Uses Visual Studio	Template based	Has custom API	No
IPSEN	Graph based database	Graph grammars	Unknown	No
Marama	XML files (XMI based - ex Eclipse)	Jet, Marama VMLPlus, XML tools	Eclipse plugin, SOAP, RMI	Yes, view definer

Comparison

Tool	Multiuser tools	Liveness	Portability	Thin client support	Cost
MetaEdit+	Yes	Yes	Multi-platform	No	High
Meta Builder	No	No compile Cycle	No	No	Academic
GME	Unclear	Versioning support	Java based	No	Free
GMF	Yes, via Eclipse	No - compile cycle	Needs Java & Eclipse	No	Free
DSL Tools	Possibly	No- compile cycle	Needs Visual Studio o	No	Free if you own VS 😊
IPSEN	No	No- compile cycle	N	No	Free
Marama	Yes for generated tools	Yes, must close/open window	Needs Java & Eclipse	Yes	Free for ac use

Exercise

- **Consider how easy it would be to construct your Assignment 1 tool using the other five systems**
 - **Strengths in each case**
 - **Weaknesses**
- **Will need to explore websites/papers to get a good feel for capabilities of the other tools.**