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 £11,500) 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+ Generated System

COMPSCI 732 §7. Meta Tools
**GME**

- *Generic Modelling Environment*, Ledeczi et al, Vanderbuilt
- [http://www.isis.vanderbilt.edu/Projects/gme/default.html](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
GME Example
Eclipse GMF

- **GMF = Graphical Modelling Framework**
  - 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
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

Diagram showing the relationships between different elements of a metamodel for DSL tools, including:
- ActivityElement
  - String Description
  - TransitionTo
- ActivityState
  - FinalState
  - StartState
Microsoft DSL Tools - Shapes

- Rounded Rectangle
  - Outline color: black
  - Fill color: gray

- Text Decorator
  - Position: center

Attend Workshop
Microsoft DSL Tools - Mapping

Shape

Domain Model

Attend Workshop

Task

String Title

String Description

TaskStatus Status
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

COMPSCI 732 §7. Meta Tools
## Comparison

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

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

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