Papyrus:
Advent of an Open Source IME at Eclipse (Redux)

Kenn Hussey

Eclipse Modeling Day, Toronto
November 18, 2009
A “Perfect Storm” for Tools

- Core technologies like MOF™ and UML® are evolving
  - Microsoft a member of Object Management Group™ (OMG™); revision of MOF (SMOF); UML RFP and roadmap working group
- Vendors have largely failed to provide consumable tools
  - time to question the status quo (some already are)
- Software industry is shifting
  - enterprises preferring use over buy over build
- Opportunity to organize and provide industrial strength alternative to proprietary tools
Overview

- Introduction
- Past - How Did Papyrus Get Here?
- Present - Where is Papyrus?
- Future - Where is Papyrus Going?
- Next Steps
Papyrus I

- Based on Eclipse and the Eclipse UML2 project (now a subproject of MDT)
  - uses de facto reference implementation of OMG’s Unified Modeling Language™ (UML) metamodel at Eclipse

- Developed as an open source project, primarily by the LISE team of the Commissariat à l’Énergie Atomique (CEA) in France
  - to support development of real-time embedded systems
  - also led the definition of OMG’s MARTE profile

- Strong focus on customizability
  - powerful profile definition capability
  - support for key UML diagram types (using DI2 standard)
TOPCASED

- TOPCASED is Model Based Engineering Platform initiated by a consortium of 30 industrial and research institutions including Airbus, Continental, Thales, Telecom Paris, and Rockwell Collins.
- It targets safety-critical system design and is soundly based on Eclipse frameworks.

**TOPCASED**
- Model editors
- Simulator Engines
- Model to Model Transformations
- Model to Text Transformations
- Formal Checking

**Configuration, Change and Requirements management tools communication**

11/18/2009

*Papyrus (Redux) © 2009 by Kenn Hussey, Zeligsoft, Atos Origin, CEA LIST, and LIFL; made available under the EPL v1.0*
MOSKitt

- Valencian Ministry of Infrastructure and Transport
- Offers support for generation and use of CASE tools
- Composed of different modules, including UML diagram editors, model explorer, transformation manager, reports generator, form editors, etc.
- Based on Eclipse UML2 format and reuses/extends UML2 Tools diagram editors
- UML2 and model explorer modules to be replaced with Papyrus
Papyrus at Eclipse

- Consolidation of several free open source UML tooling initiatives - Papyrus I, TOPCASED editor, and MOSKitt
- Integrated as an official subproject of Eclipse MDT
  - leverages infrastructure from Modeling project
  - shares strategic direction and release planning with other MDT subprojects
  - expands the field of contributors to Eclipse
Model Development Tools (MDT)

- Model Development Tools is a Modeling sub-project at http://www.eclipse.org/modeling/mdt
- Inspired by the Eclipse community’s demand for more end user “tooling” from the Modeling project
- Purpose of MDT is to provide extensible frameworks and exemplary tools for the metamodels within the scope of the Modeling project
- Next release scheduled for June 2010 (Helios)
Papyrus as a Subproject of MDT

Papyrus

GMF

VF, MT, OCL, UML2, Compare, Search, Index

GEF

EMF

Platform

MST

Papyrus

Atos Origin
Integranova
LIFL
Prodevelop
XML Modeling
Other UML Efforts at Eclipse

- **UMLX**
  - prototype editors for a QVTd-based graphical transformation language

- **UML2 Tools**
  - UML diagram editors generated (almost entirely) with GMF

- **Tigerstripe**
  - custom “UML” implementation and domain-specific tools for telecommunications
Ensuring Project Success

- To be successful, an Eclipse project must provide both technology and business value
  - Technology
    - high-quality consumable frameworks and tools
  - Business
    - commercial benefit to developers, users, and vendors
- Diversity of contributors is also key to building a sustainable ecosystem
- Papyrus is among the most diverse projects at Eclipse and is poised to provide both kinds of value
Overview

- Introduction
- Past - How Did Papyrus Get Here?
- Present - Where is Papyrus?
- Future - Where is Papyrus Going?
- Next Steps
Milestones To Date — 0.7.0 M1

- Use case diagram (partial)
- Class diagram
- Preferences
- Model explorer
- Backbone
  - Multi-window manager
  - Multi-editor manager (tabs)
  - GMF editor adapter
Milestones To Date — 0.7.0 M2

- Sequence diagram (partial)
- Composite diagram (partial)
- Control command to split models (prototype)
- Hyperlinks
- Model explorer refactoring (to remove UML dependencies)
Milestones To Date — 0.7.0 M3

- Use case diagram (complete)
- Block definition diagram
- Backbone support for DSLs
DEMO TIME!
diagrams, model explorer, model/diagram synchronization, preferences
Overview

- Introduction
- Past - How Did Papyrus Get Here?
- Present - Where is Papyrus?
- Future - Where is Papyrus Going?
- Next Steps
Model-Based Engineering (MBE)

- An approach to system and software development in which models play an indispensable role; based on two time-proven concepts.

### (1) ABSTRACTION

```java
switch (state) {
    case '1': action1;
    newState('2');
    break;
    case '2': action2;
    newState('3');
    break;
    case '3': action3;
    newState('1');
    break;
}
```

### (2) AUTOMATION

```java
switch (state) {
    case '1': action1;
    newState('2');
    break;
    case '2': action2;
    newState('3');
    break;
    case '3': action3;
    newState('1');
    break;
}
```
Industry demand for open source MBE tools is growing

- prompted creation of Eclipse Modeling project and its subproject, Model Development Tools (MDT)

Papyrus is intended to respond to that demand by providing an industrial quality, integrated toolset (IME) that supports MBE

- key principles driving Papyrus development include customizability, extensibility, scalability, usability, interoperability, and interactivity
Customizability

- It should be possible to use Papyrus and its existing facilities in ways that are best suited to the task at hand
  - provide extensive preferences for user defined settings
  - support form-based editing via highly customizable Properties view
- Bugs 249777, 257049, 269660, 271057, 290237, 290257
Extensibility

- It should be possible to extend Papyrus with new capabilities (tools, processes, languages, methods), including those that may not have been anticipated during inception
  - provide extension points for key points of variability
  - support tool specialization via OMG SysML™, MARTE, and other profiles
  - facilitate the development of domain specific languages (DSLs)

- Bugs 269490, 269492, 269494, 277478
Scalability

- Papyrus should be able to efficiently cover a broad range of differently sized problems, scaling not only in the complexity of the system being developed, but also in the size of development teams and environments involved
  - leverage EMF CDO for more scalable Ecore, UML, notation models
  - use RESTful resources API from e4 to support arbitrary repositories
  - support binary EMF resource implementations
- Bugs 275666, 290937, 290939, 290941
Usability

- Papyrus should minimize the cognitive load on users; this should extend beyond good UI design to include aspects such as adapting to specific models and (individualized) modes of usage
  - provide comprehensive documentation for both users and developers
  - introduce custom widgets (e.g., ribbons and Chrome-like navigation bar)
  - apply UI styling from e4
  - leverage Eclipse UDC and/or heuristics to personalize UI
  - integrate with Mylyn to support task-focused modeling

- Bugs 273415, 273416, 273417, 273418, 273419, 282857, 288362, 290246, 290943, 290944, 290945, 290946
Interactivity

- Papyrus should provide users with the ability to interact and share artifacts in real time, as well as monitor each others’ progress via social media facilities
  - expose resource partitioning functionality provided by EMF
  - leverage EMF Compare for concurrent modeling
  - support CDO model repository as a backing store for models
  - utilize Google Wave protocol to facilitate real-time communication
- Bugs 275628, 277680, 290951, 290952, 290953
Interoperability

- Wherever possible, Papyrus should favor industry standards over custom solutions so as to facilitate interworking with external toolsets
  - provide full support for important UML and OMG SysML diagram types
  - maintain compliance with latest standards, i.e., OCL, UML, OMG SysML
  - participate in the model interchange working group at the OMG

- Bugs 281320, 290725, 290948, 290949, 290954
Papyrus Architecture

EMF editor or others

Papyrus diagram editors

GMF Connector

GMF-based editors

GEF Connectors

GEF Editors

Papyrus Backbone

Eclipse
Overview

- Introduction
- Past — How Did Papyrus Get Here?
- Present — Where is Papyrus?
- Future — Where is Papyrus Going?
- Next Steps
## Helios Development Plan (1/3)

<table>
<thead>
<tr>
<th>#ID</th>
<th>add diagram, control, open diagram, delete,</th>
<th>Scope</th>
<th>Details</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I can create sub models</td>
<td>Collaborative work</td>
<td>requires delete UML and DI dependencies from the backbone</td>
<td>Milestone 4</td>
</tr>
<tr>
<td>2</td>
<td>I can create a sequence diagram with main (common) features</td>
<td>Sequence</td>
<td>lifelines, combined Fragments, messages (partial), guard, interaction</td>
<td>Milestone 2</td>
</tr>
<tr>
<td>3</td>
<td>I can create a sequence diagram with advanced features</td>
<td>Sequence</td>
<td>part decomposition, gates...</td>
<td>Milestone 4</td>
</tr>
<tr>
<td>4</td>
<td>I can create a composite structure diagram with main features</td>
<td>Composite</td>
<td>Role-binding / Deletion Feedback</td>
<td>Milestone 4</td>
</tr>
<tr>
<td>5</td>
<td>I can create a state machine diagram and make the main features</td>
<td>State Machine</td>
<td>Pseudo state, region, transition</td>
<td>Milestone 4</td>
</tr>
<tr>
<td>6</td>
<td>I can create an activity diagram with main features</td>
<td>Activity</td>
<td></td>
<td>Milestone 5</td>
</tr>
<tr>
<td>7</td>
<td>I can create and edit a timing diagram</td>
<td>Timing diagram</td>
<td></td>
<td>Milestone 6</td>
</tr>
<tr>
<td>8</td>
<td>I can manage (customize) model properties in a dedicated view</td>
<td>EEF properties</td>
<td>for advanced users (administrators)</td>
<td>Milestone 5</td>
</tr>
</tbody>
</table>
# Helios Development Plan (2/3)

<table>
<thead>
<tr>
<th>#ID</th>
<th>add diagram, control, open diagram, delete,</th>
<th>Scope</th>
<th>Details</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>I can create a profile using the diagram editor</td>
<td>Profile</td>
<td></td>
<td>Milestone 4</td>
</tr>
<tr>
<td>10</td>
<td>I can navigate in a model tree view and interact with it</td>
<td>Model explorer / Outline</td>
<td>add diagram, control, delete, filter,</td>
<td>Milestone 4</td>
</tr>
<tr>
<td>11</td>
<td>I can create and manage SysML models</td>
<td>SysML</td>
<td>activate SysML diagrams and SysML profile</td>
<td>Milestone 2</td>
</tr>
<tr>
<td>12</td>
<td>I can create and edit properly all the features of a Class diagram</td>
<td>Class</td>
<td>fix critical or blocking bugs</td>
<td>Milestone 4</td>
</tr>
<tr>
<td>13</td>
<td>I can create and edit properly all the features of a Use Case diagram</td>
<td>Use case</td>
<td>fix critical or blocking bugs</td>
<td>Milestone 3</td>
</tr>
<tr>
<td>14</td>
<td>I can manage a Papyrus model with a single resource</td>
<td>model + diagrams</td>
<td>Encapsulation of DI, notation and EMF</td>
<td>Milestone 4</td>
</tr>
<tr>
<td>15</td>
<td>I can install Papyrus feature in an Eclipse Platform</td>
<td>continuous Integration</td>
<td>P2, update site</td>
<td>Milestone 2</td>
</tr>
<tr>
<td>16</td>
<td>I can create a DSL based on papyrus backbone</td>
<td>backbone</td>
<td>advanced user: requires delete UML and DI dependencies from the backbone</td>
<td>Milestone 3</td>
</tr>
</tbody>
</table>
## Helios Development Plan (1/3)

<table>
<thead>
<tr>
<th>Version</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>21/09/2009</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Milestone 1</th>
<th>Milestone 2</th>
<th>Milestone 3</th>
<th>Milestone 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>11-sept</td>
<td>16-oct</td>
<td>13-nov</td>
<td>11-déc</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Milestone 5</th>
<th>Milestone 6</th>
<th>Milestone 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>30-janv</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#ID</th>
<th>add diagram, control, open diagram, delete,</th>
<th>Scope</th>
<th>Details</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>I can create and edit Block Definition Diagram</td>
<td>BDD</td>
<td></td>
<td>Milestone 3</td>
</tr>
<tr>
<td>20</td>
<td>I can create and edit Internal Block Diagram</td>
<td>IBD</td>
<td></td>
<td>Milestone 4</td>
</tr>
<tr>
<td>21</td>
<td>I can create and edit a parametric diagram</td>
<td>PARA</td>
<td></td>
<td>Milestone 5</td>
</tr>
<tr>
<td>22</td>
<td>I can create and edit a requirement diagram</td>
<td>REQ</td>
<td></td>
<td>Milestone 6</td>
</tr>
<tr>
<td>23</td>
<td>I can edit my comment through a rich text editor</td>
<td>Specific view</td>
<td>from TOPCASED</td>
<td>Milestone 5</td>
</tr>
<tr>
<td>24</td>
<td>I can create a Deployment Diagram</td>
<td>Deployment</td>
<td></td>
<td>Milestone 5</td>
</tr>
<tr>
<td>25</td>
<td>I can create an Overview diagram</td>
<td>Overview</td>
<td></td>
<td>Milestone 5</td>
</tr>
<tr>
<td>26</td>
<td>Performance measures and optimization</td>
<td>All diagrams on different models with different sizes</td>
<td>load, edit, delete</td>
<td>Milestone 6</td>
</tr>
<tr>
<td>27</td>
<td>usability</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
After Helios...

The diagram illustrates the transitioning of tools and frameworks from Eclipse RCP 3.5 (Galileo) to Eclipse RCP 3.6 (Helios).

**Ecore Editor**
- UML2 Editor
- SysML Editor
- SAM Editor
- xxx Editor

**Simulation Engine**
- ATL/QVT
- Acceleo/oAW

**Rules Checker**
- Ecore Editor

**Documentation Generator**
- code generator

**Traceability Engine**
- Eclipse Modeling Framework
- Graphical Editor Framework

**TOPCASED SDK**
- Templates

**Eclipse RCP 3.5 (Galileo)**

**MDT Papyrus backbone**
- GEF

**Eclipse RCP 3.6 (Helios)**

**Formal Proof Tools**
- safety

**TOPCASED SDK and templates**
- UML2 search compare

**Acceleo/oAW ATL/QVT**

**Comparison**
- compare
- UML2

**Eclipse Modeling Framework**

**Papyrus (Redux) © 2009 by Kenn Hussey, Zeligsoft, Atos Origin, CEA LIST, and LIFL; made available under the EPL v1.0 11/18/2009**
This Just In

- Sessions well-attended at Eclipse Summit Europe; further evidence of “perfect storm”
  - “Integrated Modeling Toolset” BoF
  - Papyrus long talk
- New initiatives underway
  - Sphinx project proposal
  - Consumer interest groups
- Papyrus will be aligned with these (and other) initiatives to drive additional technology and business value
How Can You Help?

- **As a developer...**
  - write documentation
  - participate in mailing list discussions
  - become a Papyrus contributor!

- **As a vendor...**
  - build extensions for Papyrus
  - participate in *industry working groups*

- **As a user...**
  - use Papyrus
  - participate in forum discussions
  - report bugs
More Information

- For developers...
  - http://dev.eclipse.org/mailman/listinfo/mdt-papyrus.dev

- For vendors...
  - http://www.eclipse.org/papyrus

- For users...
  - news://news.eclipse.org/eclipse.papyrus
Legal Notices

- **MOF, Object Management Group, OMG, OMG SysML, and Unified Modeling Language** are trademarks of the Object Management Group.
- **UML** is a registered trademark of the Object Management Group.
- **Java** and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others.