Sphinx

An Industrial Strength Tool Platform Fostering Model-driven Development of Embedded Systems

Dr. Stephan Eberle

2010-06-24
About

■ Stephan Eberle
  ◆ Development manager at Geensys in Paris, France
  ◆ Lead of Artop Core and Validation
  ◆ Coming soon: Sphinx project co-lead
  ◆ Frequent speaker at conferences and events
Outline

- About Embedded System Design Tooling
  - Can Eclipse help?
  - Upcoming: Sphinx
  - Wrap-up
Increasing Complexity & Expectations

Compliance with quality and safety standards:

- DO 178B
- CMMI
- IEC 61508
- DO 254
- ISO 26262
- ...
Key role: Integrated Full Lifecycle Tool Support

Objectives:
- Continuous design flow
- Automatic/assisted synchronization of design changes
## Challenge 1: Domain-Specific Design Standards

<table>
<thead>
<tr>
<th>Design Standard</th>
<th>Covered Aspect</th>
<th>Target Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTomotive Open System Architecture</td>
<td>Software architecture</td>
<td>Automotive</td>
</tr>
<tr>
<td>ODX</td>
<td>Diagnosis</td>
<td>Automotive</td>
</tr>
<tr>
<td>RIF</td>
<td>Requirements definition</td>
<td>Automotive</td>
</tr>
<tr>
<td>Architecture Analysis &amp; Design Language</td>
<td>Software architecture</td>
<td>Aerospace</td>
</tr>
<tr>
<td>Transportmodel</td>
<td>Software architecture</td>
<td>Transportation</td>
</tr>
</tbody>
</table>
Challenge 2: Changing Tool Provisioning Approach

- Shift from vendor-driven to end-user-driven tooling
  - Increasing reluctance against depending on vendor-controlled product strategies
  - Increasing in-house/off-the-shelf ratio in design tool chains
  - Increasing demand for off-the-shelf tools being based on open source tool platforms (e.g. ISOFT, Denso)
  - Increasing involvement of large end-user enterprises in open source modeling projects (SAP, UBS, Airbus, Bosch, etc.)
Key Requirements

- Need for tightly integrated tool environments which
  - Support multiple domain-specific standards/methodologies in parallel
  - Are tailored to user-defined development processes/practices
About Embedded System Design Tooling

Can Eclipse help?

Upcoming: Sphinx

Wrap-up
Yes, it could…
Example:
AUTOSAR Design Tools at Geensys
But…
Example:
AUTOSAR Design Tools at Geensys

8..10 man-years

Acceptable
AUTOSAR design tools

© 2010 Geensys. All rights reserved.
“Acceptable” means…

- **Obvious basic services**
  (e.g. undo/redo)

- **Scalability**
  (e.g. fast loading of big models)

- **Robustness**
  (e.g., no deadlocks)

**End-user perception at this point:**
- Meets expectations but no “JDT effect”
- No added value in Eclipse
Some good news: Artop & Papyrus

Since Oct 2008: Artop

Since Nov 2008: MDT Papyrus
Open Questions

- Who is going to provide similar platforms for other design standards?
- How about cross-platform interoperability?
About Embedded System Design Tooling
Can Eclipse help?
Upcoming: Sphinx
Wrap-up
Genesis of Sphinx

- Proposed **Eclipse MDT project** providing an **integrated modeling tool platform**
- Main use case:

  Modeling language(s) + Sphinx

  ⇒ **Industrial strength integrated modeling tool environment**

- **Origins**
  - AUTOSAR-independent layer (ECL) of **Artop**
  - Backbone of **Papyrus**
“Service-Oriented” Architecture

Application Services
- Navigation
- Editor Management
- Compare & Merge
- Compare & Merge Team Integration
- Validation

Core Services
- Workspace Management
- Scoping & Model Management
- Metamodel Management
- Commands & Complex Operations

© 2010 Geensys. All rights reserved.
Industrial Strength

■ Scalability
  - Fast model loading and proxy resolution
  - Shared model instances
  - Memory-optimized model unloading
  - Indexing (planned)
  - Model repository integration (planned)

■ Robustness
  - Thread-safe model manipulation APIs
  - Deadlock avoidance (automated integration tests)
  - Damaged file loading
Wrap-up: Mission of Sphinx
Wrap-up: Mission of Sphinx (cont’d)

- Reduced effort in supporting other design standards
- Increased cross-vertical interoperability
Wrap-up: Outlook

- Feb 2010: Sphinx project proposed
- July 2010: Creation review
- August 2010: Initial code contribution from Artop
- August 2010: Start of consolidation with Papyrus Backbone
- November 2010: Rest of Artop migrated to Sphinx
- June 2011: First Sphinx release
Thank you!