Eclipse Open Source Platforms for Model-based Engineering with UML, SysML, and more

Philip Langer
planger@eclipsesource.com
Model-based Engineering

● Semi-formal models (UML, SysML, DSMLs, …)
  ○ Describe aspects of interest about a system
  ○ Requirements, architecture, behavior, …

● Pragmatic modeling: don’t model without a reason!
  ○ Answer certain questions
  ○ Automate certain development tasks

● Good reasons
  ○ Analysis of certain system properties
  ○ Validation & verification (consistency)
  ○ Code & test generation (automation)
  ○ Traceability, Exchange, Documentation, Communication, …
Domain-specific Modeling

- Model-based engineering is most successful if it is *domain-specific*
  - Highly customized modeling environments
  - Directly reflecting specific needs of a domain and its users
  - User roles, their backgrounds, methodologies, and tool chains
Domain-specific Modeling
Domain-specific Modeling vs UML, SysML, etc.

- Standardized Modeling Languages (UML, SysML, …)
  - Reuse well-known and -proven language concepts
  - Reuse existing tools and components
  - Interoperability and connectability with other models
  - Conformance to industry standards

- Domain-specific Modeling vs. UML/SysML?
  - Contradiction? NO!
Domain-specific Modeling with UML, SysML, etc.

- Thanks to the great Open Source Eclipse Modeling Ecosystem...

Papyrus  Xtext  EMF Forms  Sirius

GLSP: Diagrams in the cloud
Eclipse Papyrus as a Platform

- Open Source UML modeling platform
  - Not only a UML tool
Eclipse Papyrus as a Platform

- Open Source UML modeling platform
  - Based on the Eclipse Modeling Framework
  - Based on modeling standards: UML, SysML, OCL, fUML, Alf, …
  - Supported by an active open-source community
  - Enables to build domain-specific tools based on UML, SysML, etc.
  - Customizability: graphical syntax, palette, property views, editing behavior, etc.
Really Brief Demo
OpenADx / Systems Modeling Suite

- High complexity of Automated Driving systems
  - Traceability, Consistency, Interoperability are key (ISO26262)
  - Common “virtual description” (model) of relevant concepts
    - Vehicles, devices, sensors, requirements, etc.

- Take advantage of Papyrus and/or other Eclipse technologies
  - Choose suitable basis (SysML, UML, etc.)
  - Choose what’s relevant in which context of OpenADx or development phase
  - Which domain-specific additions are necessary
  - Integration with other tools of the OpenADx tool chain

→ Streamlined OpenADx modeling base tool for the OpenADx Community
OpenADx / Systems Modeling Suite

- Collaboration
  - Domain expertise
  - Eclipse technology expertise

- Successful similar projects
  - Papyrus for Information Modeling (~ 20 days)
  - Papyrus for UML Light (~ 50 days)

- Please get in contact with me
  - Philip Langer <planger@eclipsesource.com>