geensys

Global Embedded Electronics & Networked SYstem Solutions

Modeling Platform Architecture

Dr. Stephan Eberle

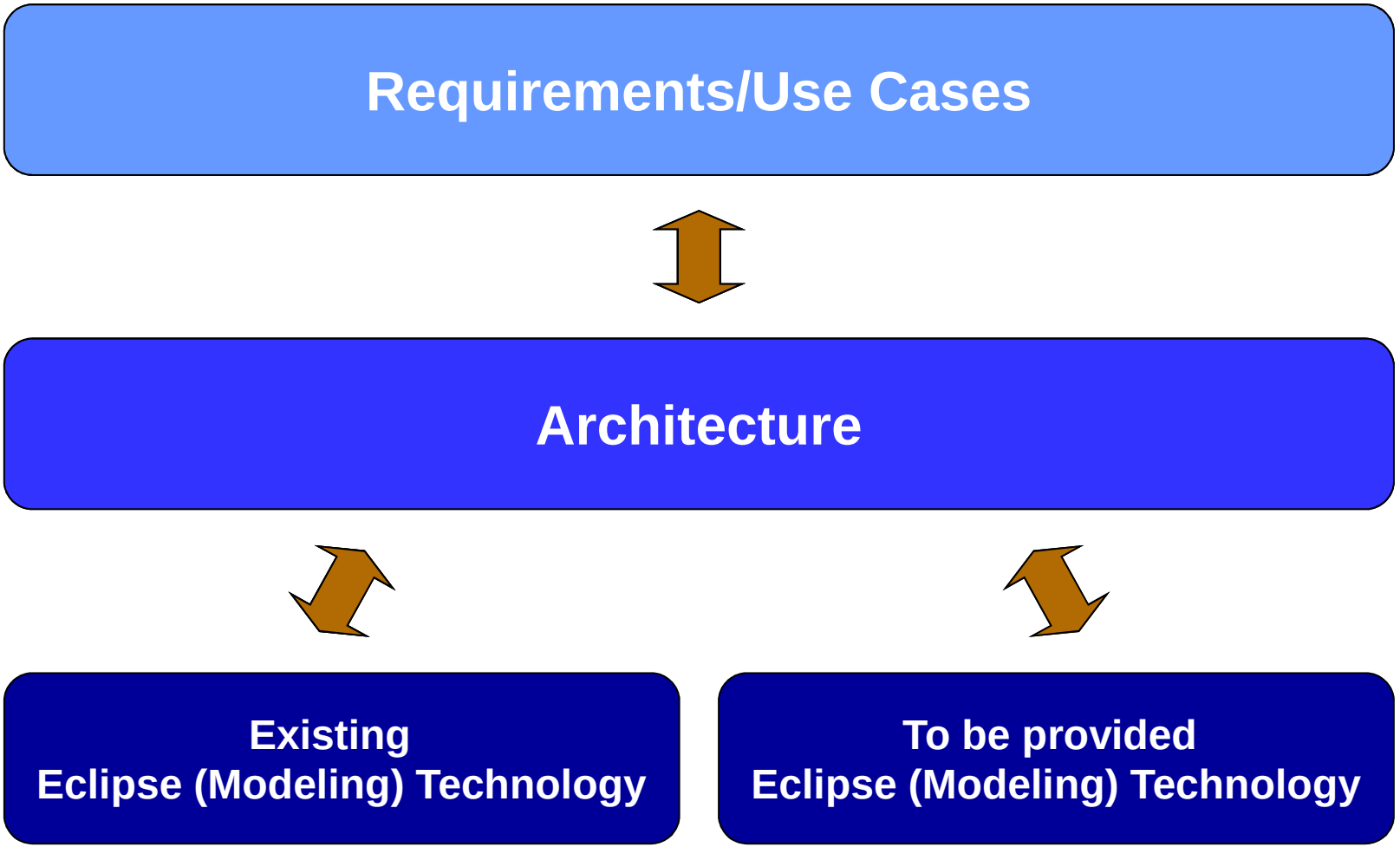
2010-06-14

Overview

- Objectives of Modeling Platform Architecture
- Mapping of Requirements/Use Cases to Architecture
- Demo and Status of Sphinx
- Wrap-up

geensys

Big Picture



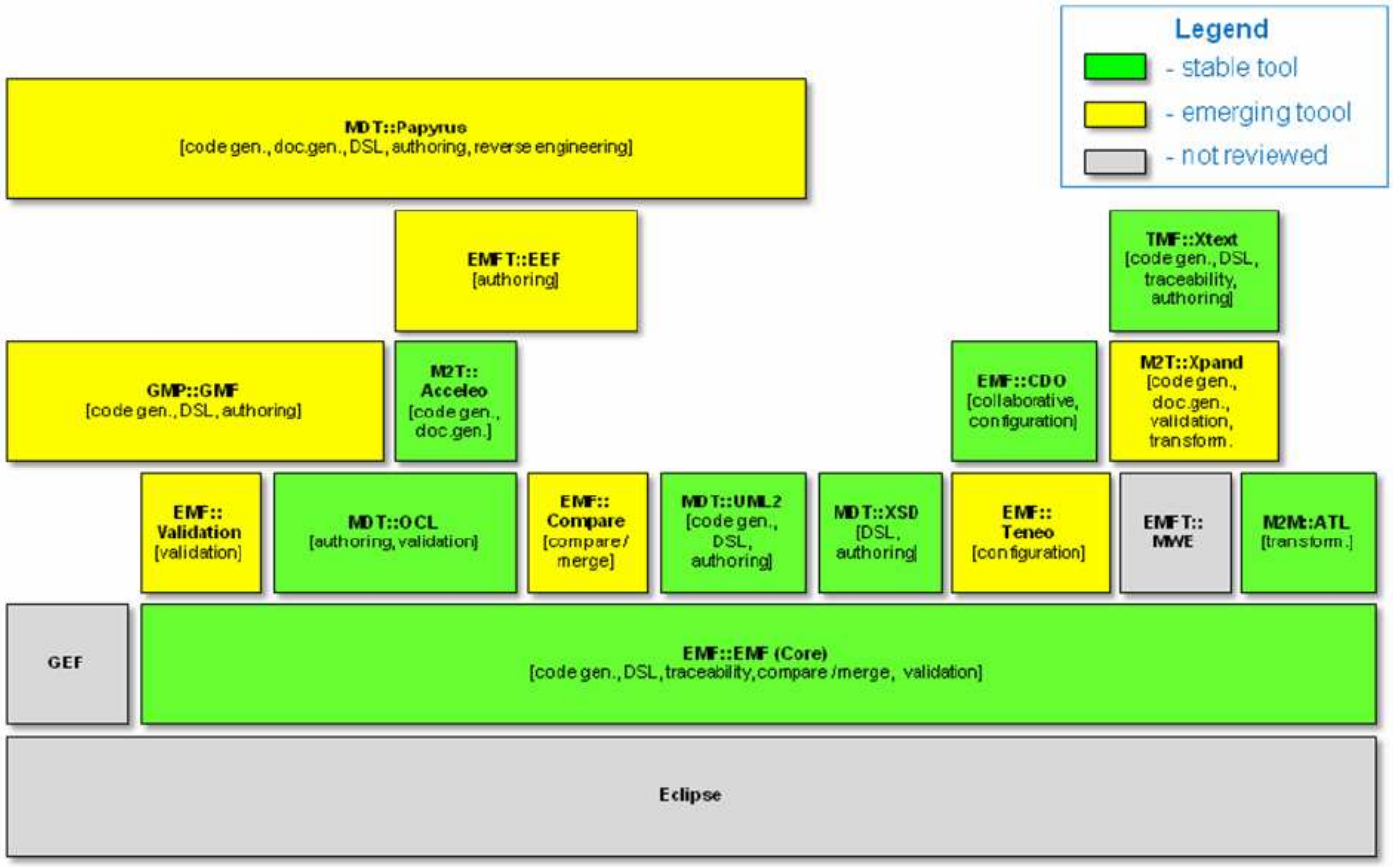
geensys

Architecture Objectives

- To overcome fragmentation and duplication of the Eclipse Modeling projects
  - View that is not driven by existing Eclipse Modeling technologies and their decomposition
- To identify the services the MP should provide (“20 Modeling Things”)
  - Features and scope of each service
  - Dependencies between services
  - Layers regrouping services in meaningful subsets
- To provide a common frame for alternative implementations

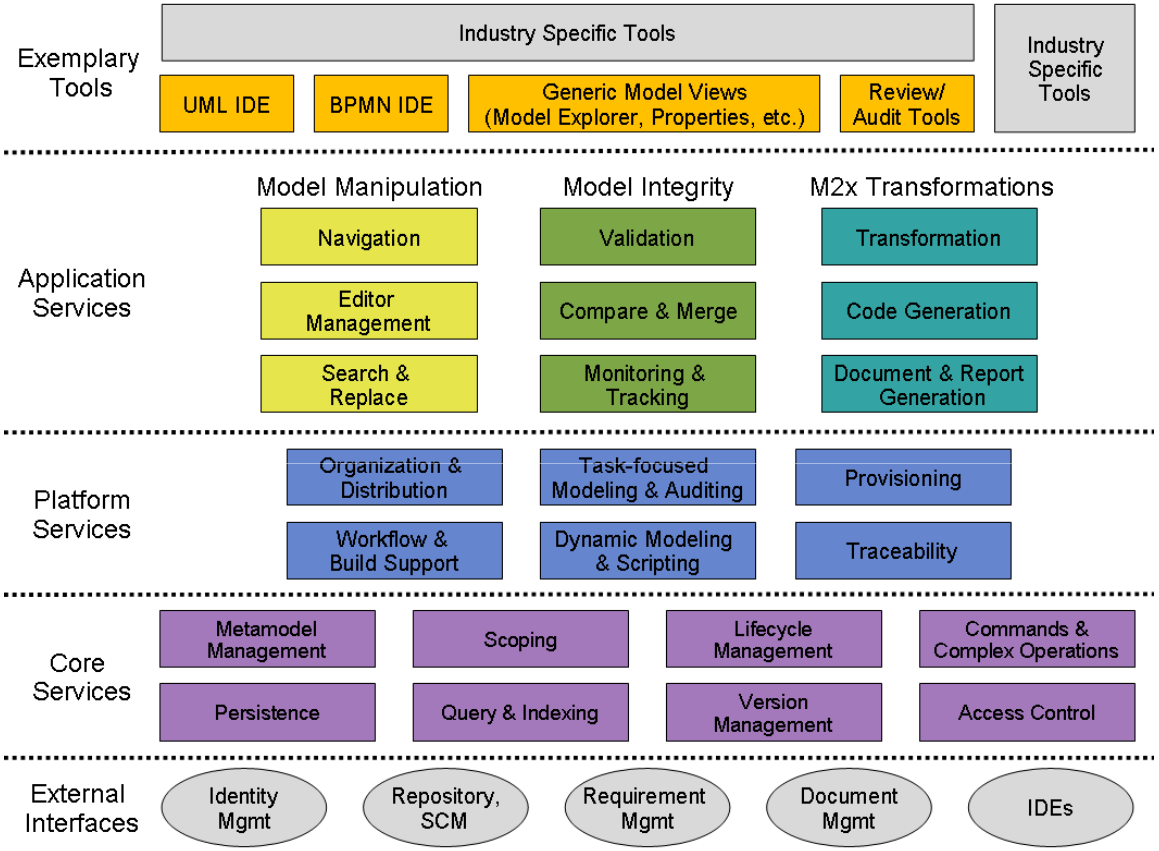
geensys

Architecture Draft from Zeligsoft Report



▶ **Technology-driven**

Architecture Draft of MPIWG



▶ **Service-driven**

Overview

- Objectives of Modeling Platform Architecture
- Mapping of Requirements/Use Cases to Architecture
- Demo and Status of Sphinx
- Wrap-up

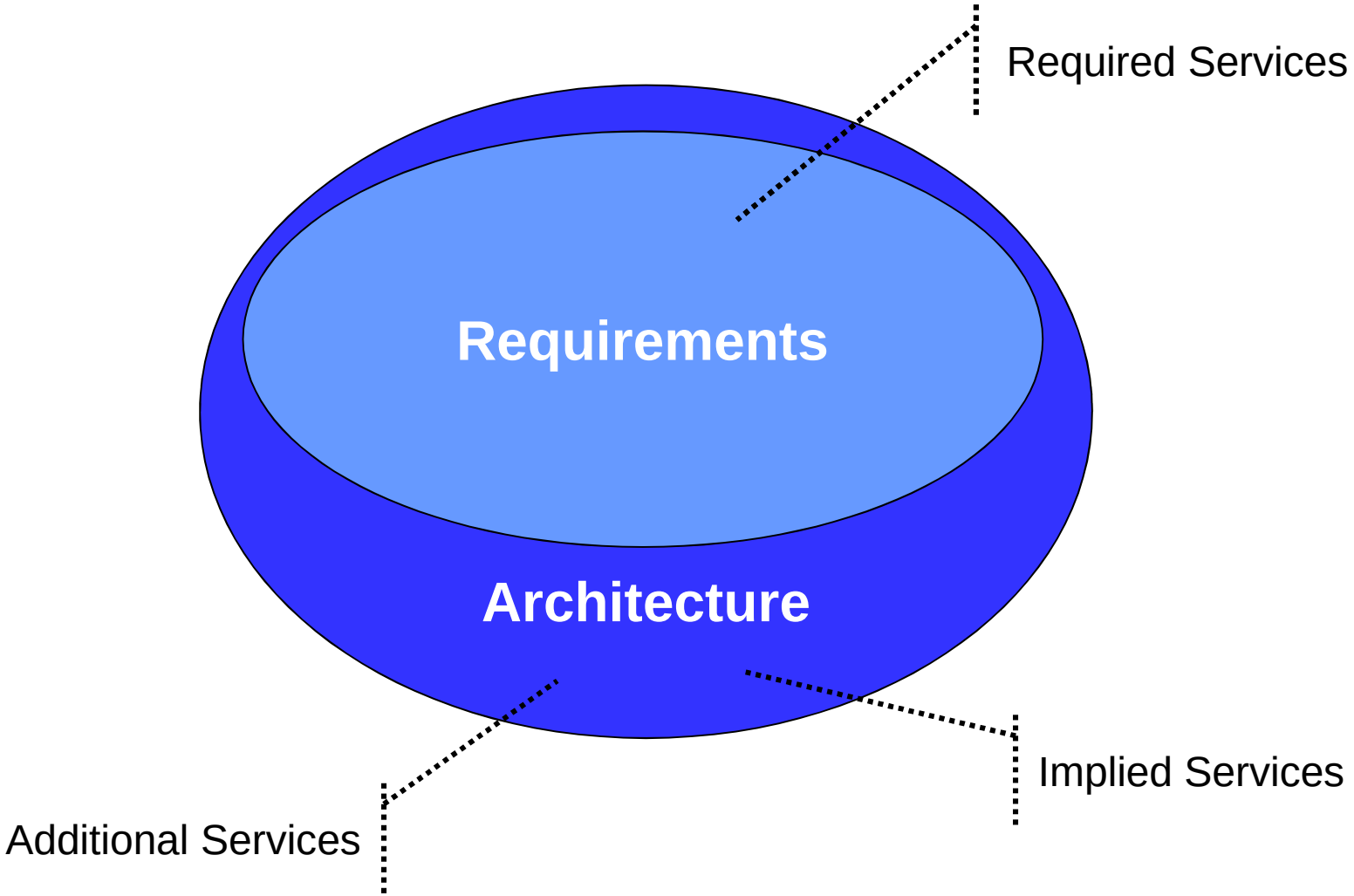
Discussion: Mapping of Requirements/Use Cases to Architecture

▶ **See Google spreadsheet**

http://spreadsheets.google.com/ccc?key=0AuLcEWPBzTcidF9PVkczbjliTFJNZ2UzS3FxS2dORVE&hl=en#gid=0

geensys

Coverage



Implied Services

- Persistence
- Scoping
- Lifecycle (Workspace) Management
- Commands & Complex Operations (partially)
- Query & Indexing
- Navigation
- Editor Management
- Validation

▶ **Mandatory,**
i.e., must be considered in effort estimation/planning

Additional Services

- Organization & Distribution
- Workflow & Build Support
- Dynamic Modeling & Scripting
- Search & Replace

Optional,
i.e., can be omitted in effort estimation/planning

Tentative Architecture Layers

- **Core Services**
  - Essential services (i.e., hardly possible to create any modeling applications without these)
  - Cross-cutting nature (i.e., impact all aspects of modeling applications)
  - Applicable to any kind of modeling application (i.e., modeling tools and non-tool applications)
- **Platform Services**
  - Extended services (i.e., provide support for important additional aspects but possible to create modeling applications without these)
  - Cross-cutting nature (i.e., impact multiple aspects of modeling applications)
  - Primarily used in for modeling tools (but not so much in non-tool applications)

geensys

Tentative Architecture Layers (cont'd)

- **Application Services**
  - Services supporting individual aspects of modeling applications (i.e., not all of them are necessarily required by any modeling application)
  - Applicable to any kind of modeling application (i.e., modeling tools and non-tool applications)

Overview

- Objectives of Modeling Platform Architecture
- Mapping of Requirements/Use Cases to Architecture
- Demo and Status of Sphinx
- Wrap-up

Genesis of Sphinx[1]

- Proposed **Eclipse MDT[2] project** providing an **integrated modeling tool platform**
- Main use case:

Modeling language + Sphinx

⇨ **Integrated modeling tool environment**

- Origins of Sphinx
  - Generic layer (ECL) of **Artop**[3]
  - Backbone of **Papyrus**[4]

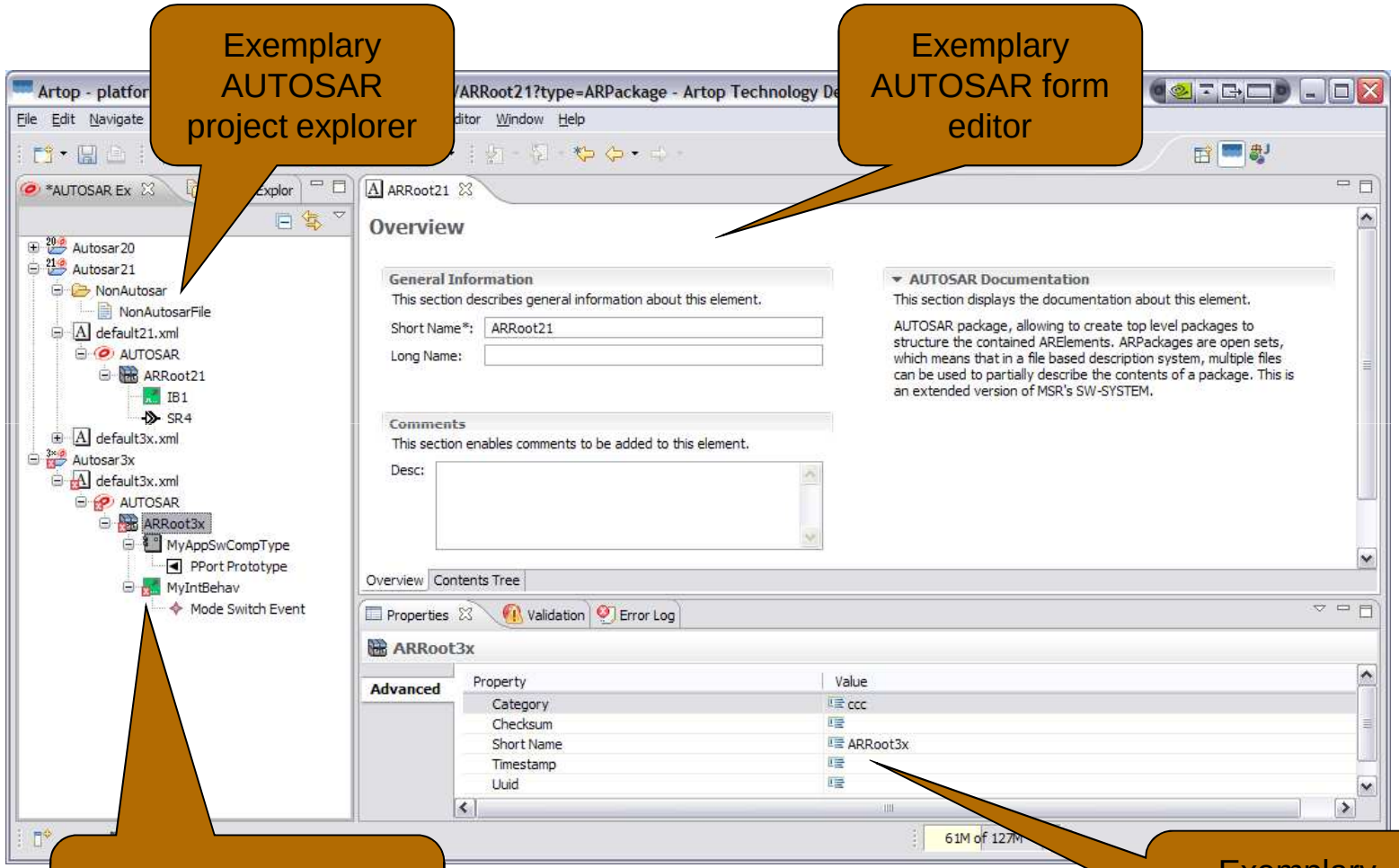
1. http://www.eclipse.org/proposals/sphinx
2. Model Development Tools
3. http://www.artop.org
4. http://www.papyrusuml.org

Objectives

- Open and extensible platform enabling rapid creation of integrated modeling tool environments (IME) for standard or domain-specific modeling languages
- Industrial strength scalability and robustness out-of-the box
- Domain- and vendor-independent interoperability layer (backbone) for off-the-shelf and in-house modeling tool components

geensys

Demo: Artop Technology Demonstrator



Demo: Sphinx

- “Out of the box support of industry standards UML, BPMN ans SysML”
- “Capacity to extend support to any other modeling standard”

Demo: Sphinx (cont'd)

- "All MP services must be applicable to user-defined domain specific modeling languages"
- "Support for definition DSLs, handling of DSL instances, validation of DSL instances against corresponding DSL definition"
- "Support for different versions of a metamodel in the same environment"

Demo: Sphinx (cont'd)

- "Support for models containing 300 000 model objects in 7000 resources"

Demo: Sphinx (cont'd)

- "Support of automatic bulk merge operations"
- "Automatic detection of inconsistencies in merged model and decorations identifying those"
- "Automatic/transparent upgrade of old model instances to newer metamodel version"

Current Status

- Feb 2010: Sphinx project proposed
- March 2010: Project creation/initial contribution delayed, formal approval of code contributions under EPL at BMW Car IT required
- April 2010: ECP project proposed, significant overlap with Sphinx
- May 2010: EPL approval finished BMW Car IT, internal “4 eyes review” of intended code contribution required

geensys

Next steps

- June/July 2010: Creation review
- July/August 2010: Initial code contribution from Artop ECL
- September 2010: Migration of Artop AAL to Sphinx
- September/October 2010: Start of consolidation with Papyrus Backbone

Wrap-up

- **MP Architecture**
  - Meaningful MP services identified
  - Layers and dependencies not final yet
- **Requirements/architecture mapping**
  - All functional requirements have a home in the architecture
  - !! Significant number of implied services that will need to be made available as well !!
- **Sphinx**
  - Striking parallels between multiple MP requirements and things that Sphinx will provide right away
  - Entering home stretch to project creation/initial contribution