geensys

Global Embedded Electronics & Networked SYstem Solutions

Sphinx

An Industrial Strength Tool Platform Fostering Model-driven Development of Embedded Systems
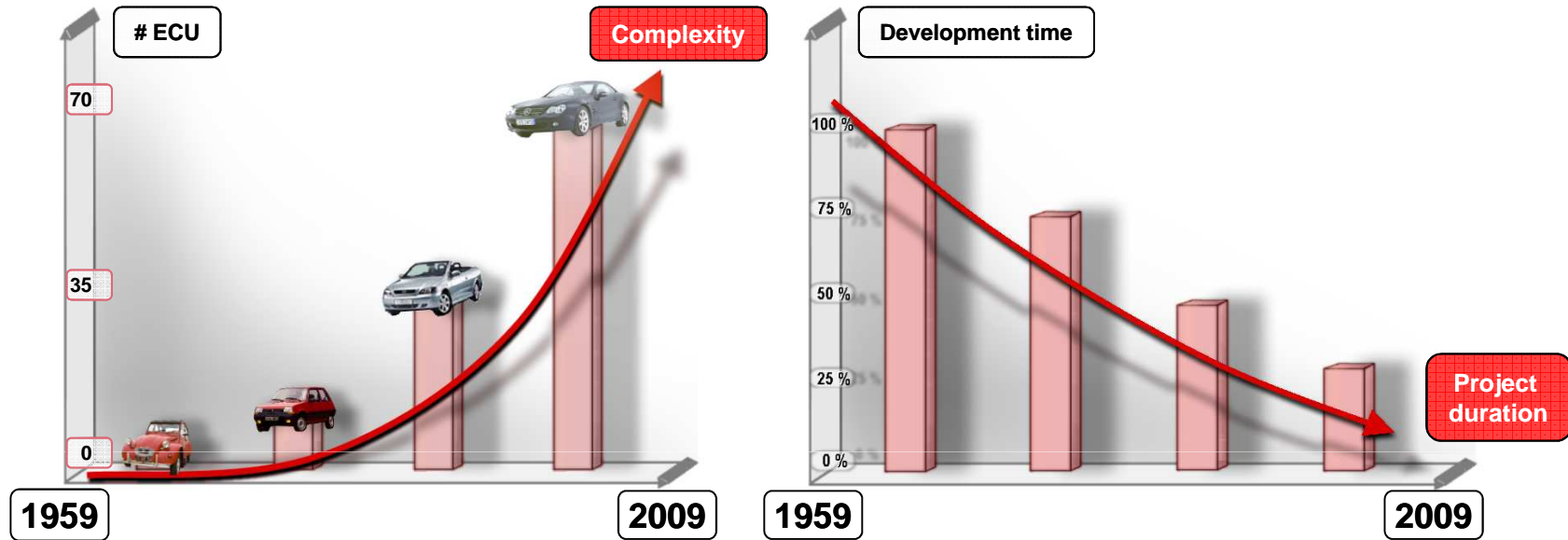
Dr. Stephan Eberle

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

geensys

Outline

▶ **About Embedded System Design Tooling**

■ **Can Eclipse help?**

■ **Upcoming: Sphinx**

■ **Wrap-up**

geensys

Increasing Complexity & Expectations

Compliance with quality and safty standards:

CMMI

DO 178B

IEC 61508

ISO 26262

DO 254

…

Key role:
Integrated Full Lifecycle Tool Support

Requirements definition
Functional design
Variant management
Software architecture
Coding
Software build
Rapid prototyping
Unit testing
Integration
Subsystem verification
Simulation
Diagnosis
System validation

Development process

Design tool chain
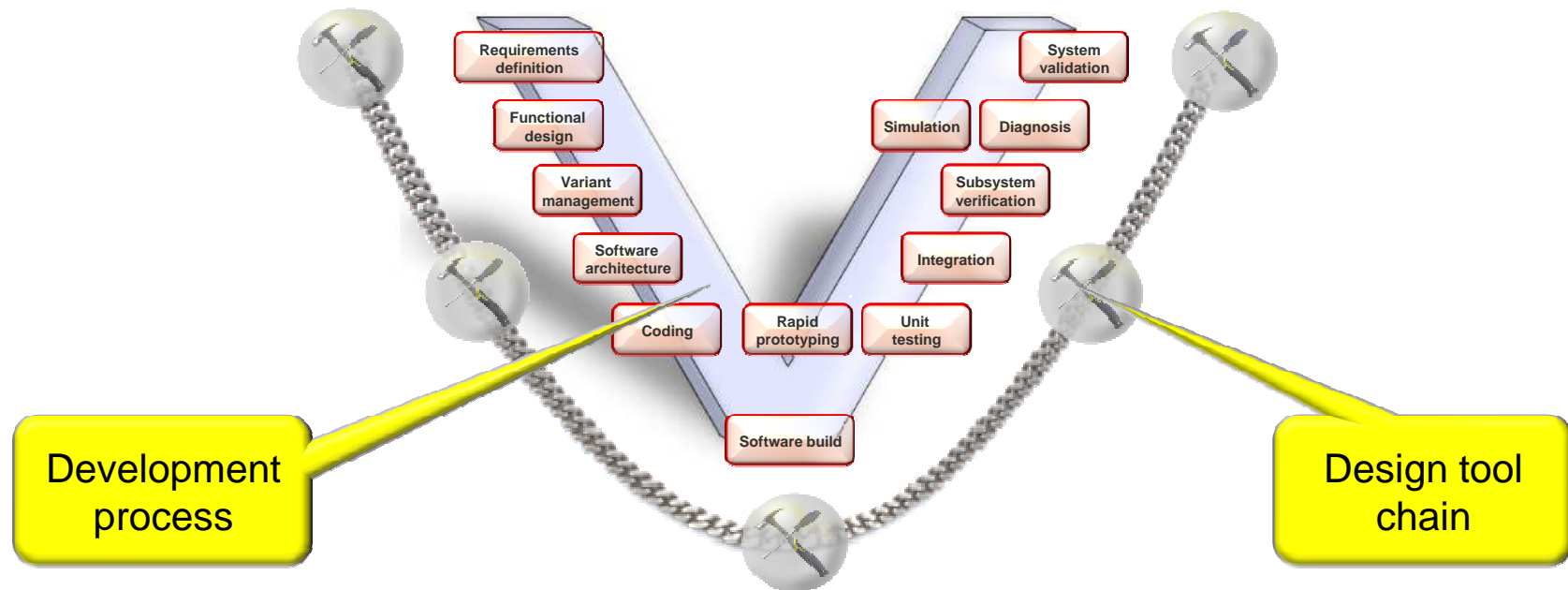
Objectives:

▶ Continuous design flow

▶ Automatic/assisted synchronization of design changes

Challenge 1:
Domain-Specific Design Standards

Design Standard	Covered Aspect	Target Domain
AUTOSAR AUTomotive Open System Architecture	Software architecture	Automotive
ODX Open Diagnostic data eXchange format	Diagnosis	Automotive
RIF Requirements Interchange Format	Requirements definition	Automotive
AADL Architecture Analysis & Design Language	Software architecture	Aerospace
Transmodel	Software architecture	Transportation

geensys

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.)

geensys

Key Requirements

■ **Need for tightly integrated tool environments which**

❑ Support multiple domain-specific standards/methodologies in parallel

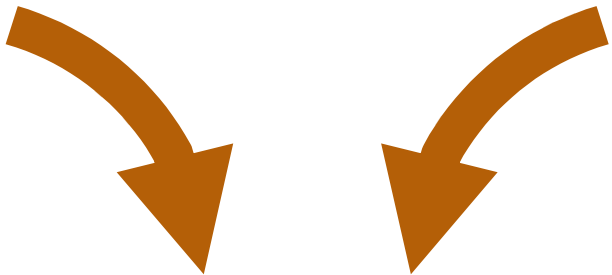❑ Are tailored to user-defined development processes/practicies

Outline

- **About Embedded System Design Tooling**
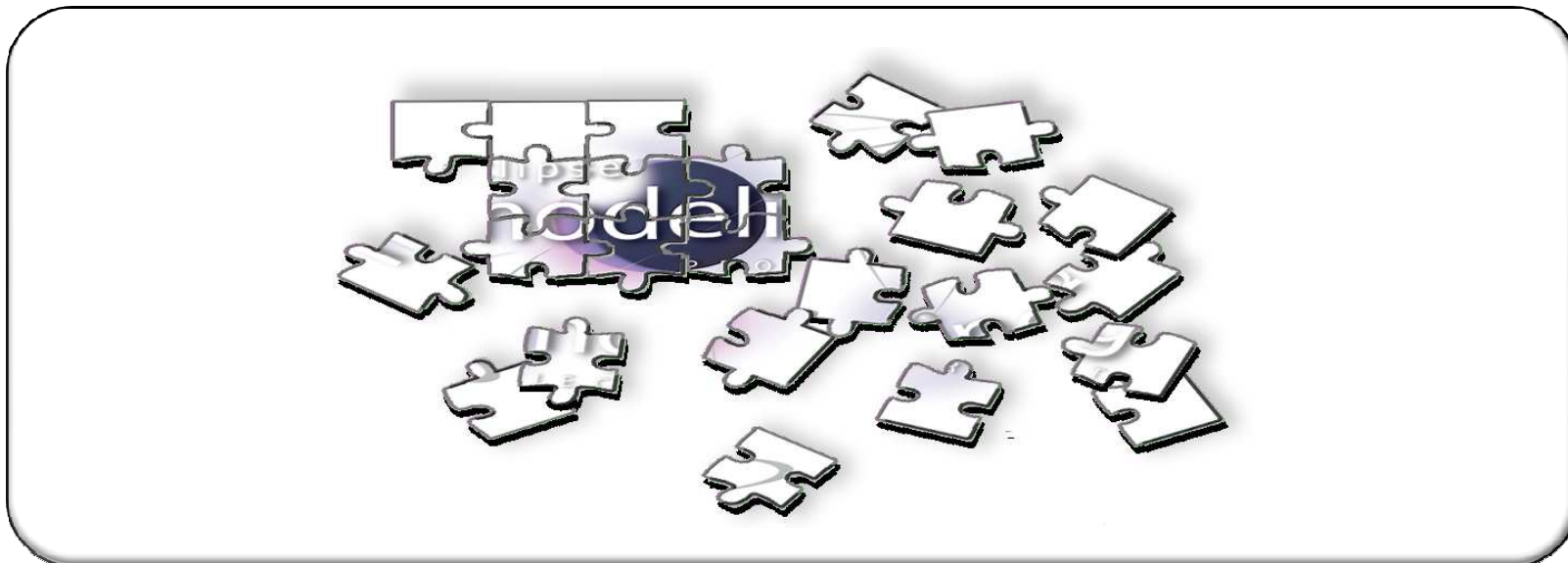- ▶ **Can Eclipse help?**
- **Upcoming: Sphinx**
- **Wrap-up**

Yes, it could…
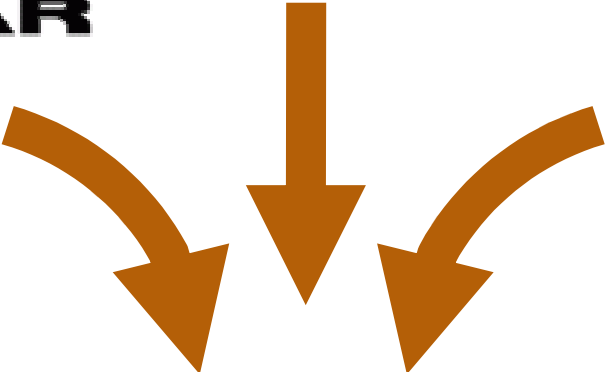
Example: AUTOSAR Design Tools at Geensys

But…

Example:
AUTOSAR Design Tools at Geensys

8..10 man-years

Acceptable AUTOSAR design tools

"Acceptable" means…

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

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

- **Robustness**
  (e.g., no deadlocks)
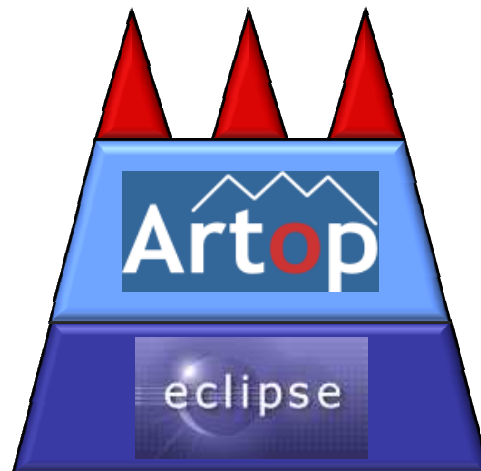
End-user perception at this point:

▶ Meets expectations but no "JDT effect"

▶ No added value in Eclipse

geensys

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**?

geensys

Outline

- **About Embedded System Design Tooling**
- **Can Eclipse help?**
- ▶ **Upcoming: Sphinx**
- **Wrap-up**

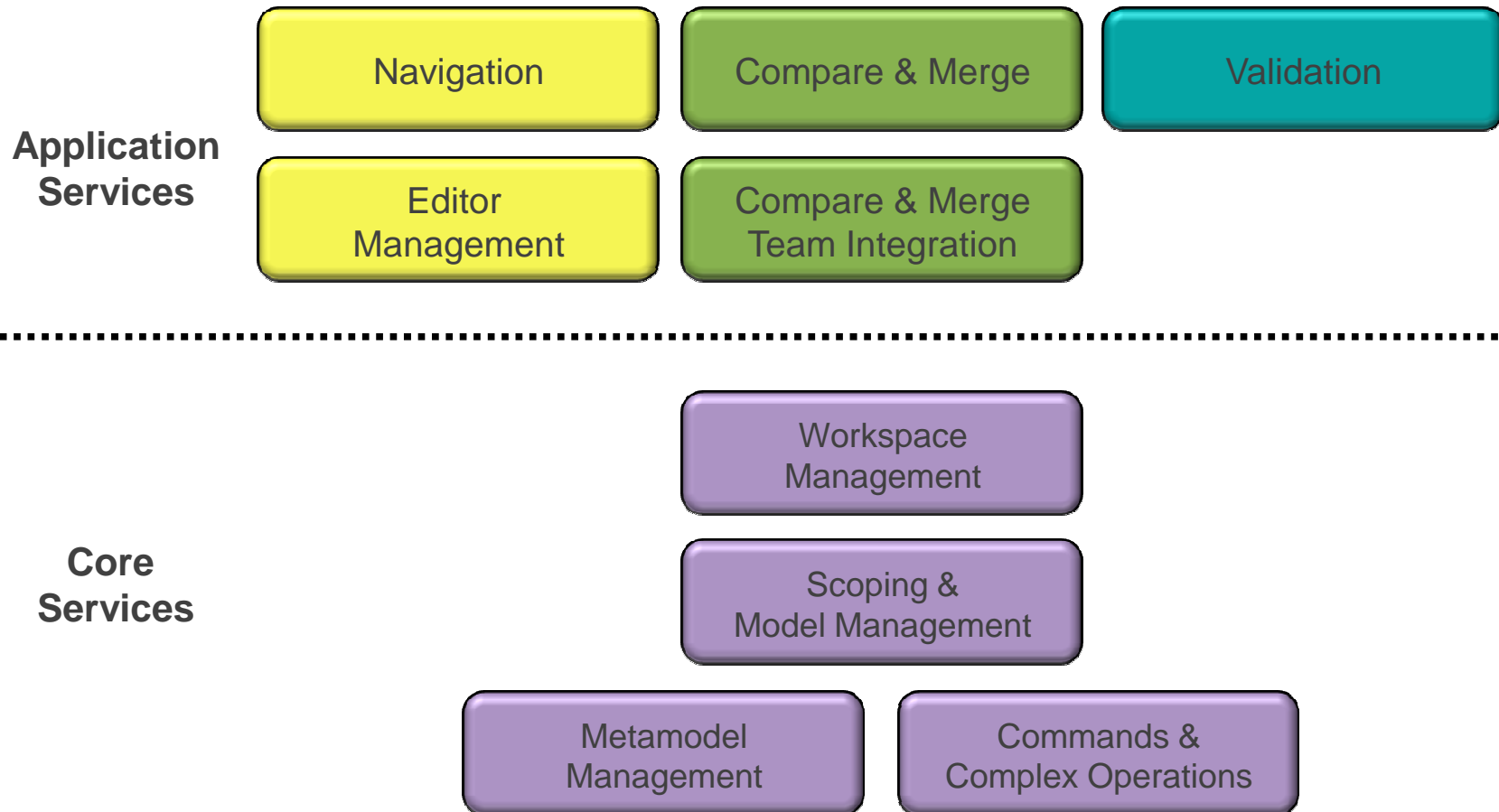geensys

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**

geensys

"Service-Oriented" Architecture

Application Services

Navigation

Compare & Merge

Validation

Editor Management

Compare & Merge Team Integration

Core Services

Workspace Management

Scoping & Model Management

Metamodel Management

Commands & Complex Operations

geensys

19

Industrial Strength

■ Scalability

❑ Fast model loading and proxy resolution

❑ Shared model instances

❑ Memory-optimized model unloading

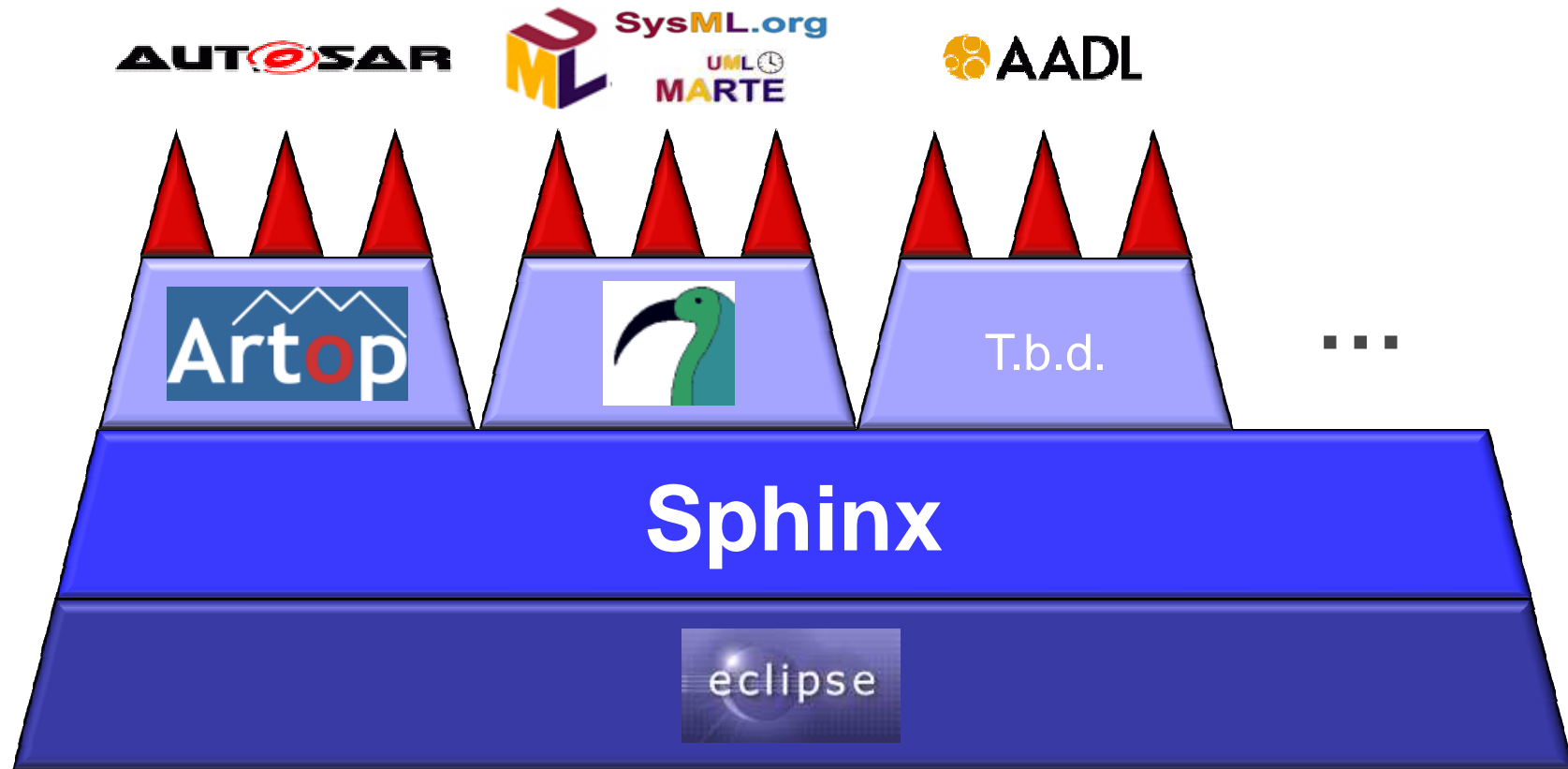❑ Indexing (planned)

❑ Model repository integration (planned)

■ Robustness

❑ Thread-safe model manipulation APIs

❑ Deadlock avoidance (automated integration tests)

❑ Damaged file loading

Wrap-up:
Mission of Sphinx

- ► 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

geensys

Thank you!