The Software Engineering Institute

# Overview of the XML Schema for the SAE AADL

**Peter Feiler**
**Software Engineering Institute**
**phf@sei.cmu.edu**
**412-268-7790**

The Software Engineering Institute

# Outline

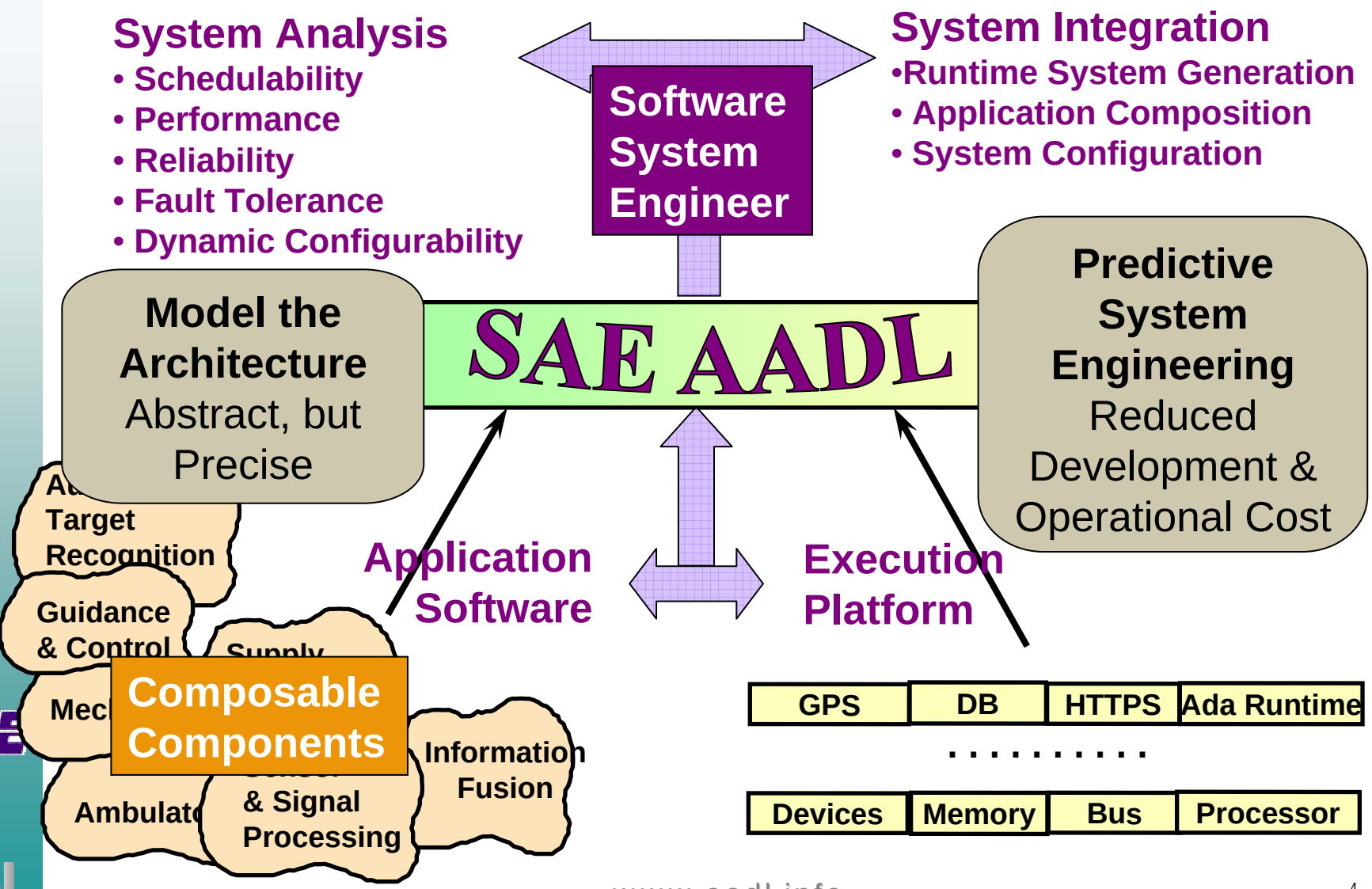
- Background: SAE AADL Standard
- An XML-Based Tool Strategy
- Defining the XML representation
- Plug’n’play tool components
- Summary

The Software Engineering Institute

# SAE Architecture Analysis & Design Language

- Notation for specification of task and communication architectures of Real-time, Embedded, Fault-tolerant, Secure, Safety-critical, Software-intensive systems
- Fields of application: Avionics, Automotive, Aerospace, Autonomous systems, …
- Based on 15 Years of DARPA funded technologies
- Standard approved by SAE in Sept 2004
- www.aadl.info

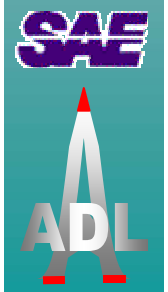

# Model-Based Engineering



The Software Engineering Institute

# The AADL Standard

- Requirements document SAE ARD 5296
  - Input from aerospace industry
  - Balloted and approved in 2000
- SAE AADL document SAE AS 5506
  - Core language approved by SAE Sept 2004
- In review to be balloted late 2004
  - Graphical AADL notation
  - UML profile of AADL for UML1.4 and UML 2.0
  - XMI domain model, XML schema
  - Ada and C Annex
- In development
  - Error Model Annex
  - ARINC 653 Annex



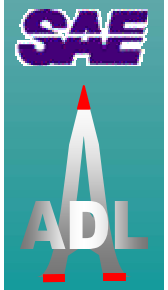
The Software Engineering Institute

# Standard Model Interchange Format

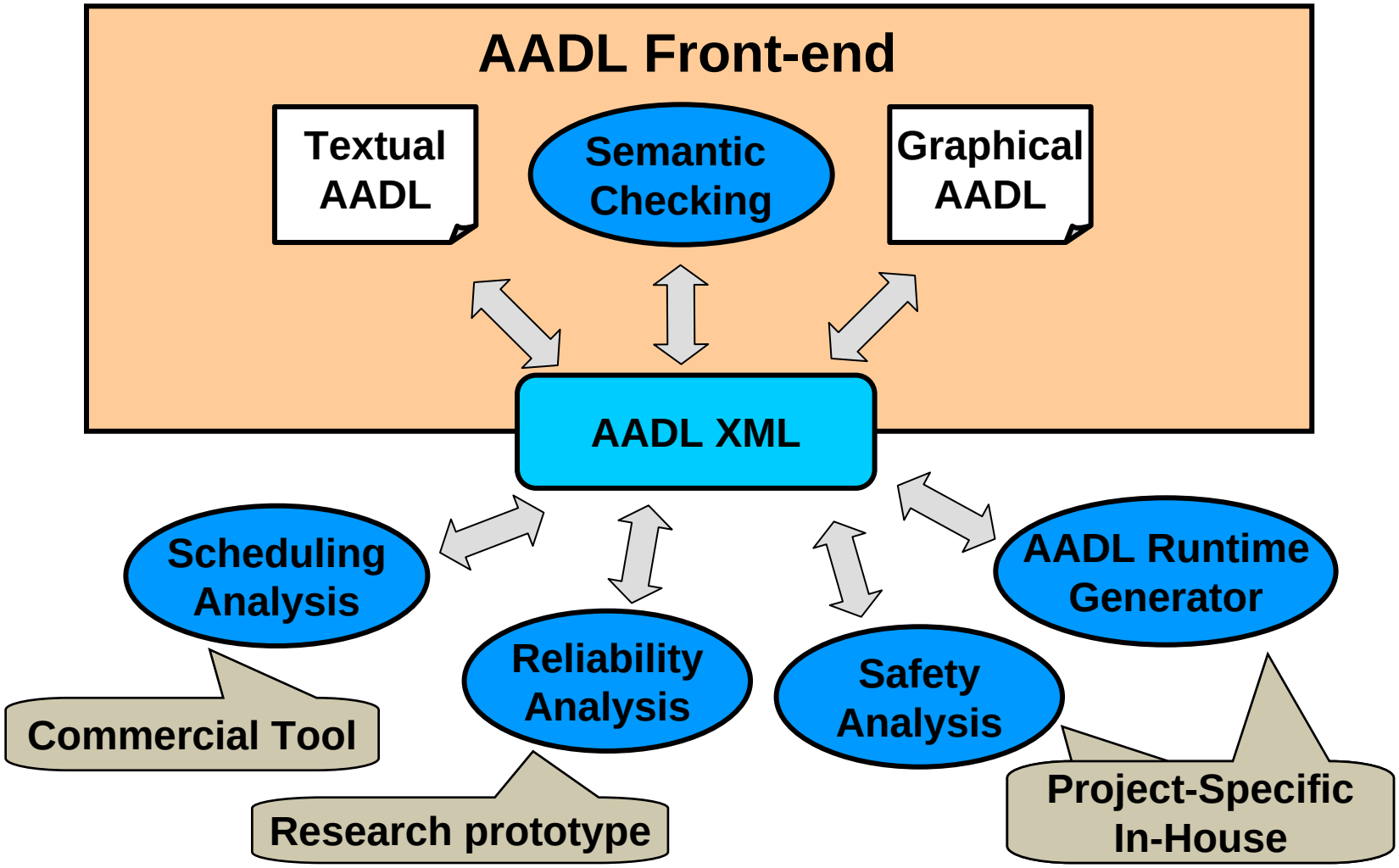
## Benefits

- (Sub-)Contractor architecture model delivery & exchange
- Company internal architecture model library
- Tool interoperability

**Why is textual AADL not sufficient?**



# An XML-Based Tool Strategy



The Software Engineering Institute

# Two-Tier Tool Strategy

- Open Source AADL Tool Environment (OSATE)
  - Developed by SEI
  - Low entry cost solution (no cost CPL)
  - Multi-platform based on Eclipse 3.0
  - Prototyping environment for project-specific analysis
  - Architecture research platform
- Commercial Tool Support
  - UML tool environment extension based on UML profile
  - Extension to existing modeling environment with AADL export/import
  - Analysis tools interfacing via XML or XML to native filter
  - Runtime system generation tools

The Software Engineering Institute

# AADL Text Example

```
package edu::cmu::sei::XMIExample
public
    system GPS
    features
          init: in event port;
          signal: out data port GPS_Signal;
    end GPS;
    system implementation GPS.basic
    end GPS.Basic;
    data GPS_Signal
    end GPS_Signal;
end edu::cmu::sei::XMIExample;
```

The Software Engineering Institute

# AADL XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<core:AadlSpec xmi:version="2.0" ……>
  <packageSpec name="edu::cmu::sei::XMIExample">
    <aadlPublic>
      <systemType name="GPS">
        <features>
          <eventPort name="init"/>
          <dataPort name="signal" direction="out"
        dataClassifier="#//packageSpec[@name=edu::cmu::sei::XMIExample]/aa
        dlPublic/dataType[@name=GPS_Signal]"/>
        </features>
      </systemType>
      <systemImpl name="GPS.basic"
        compType="#//packageSpec[@name=edu::cmu::sei::XMIExample]/aadlP
        ublic/systemType[@name=GPS]"/>
      <dataType name="GPS_Signal"/>
    </aadlPublic>
  </packageSpec>
</core:AadlSpec>
```

The Software Engineering Institute

# Large-Scale Development

- Component type and implementation declarations in *packages*
  - Name scope for component types
  - Public and private package sections
  - Grouping into manageable units
  - Nested package naming
  - Qualified naming to manage name conflicts
- Supports independent development of subsystems
- Supports large-scale system of system development

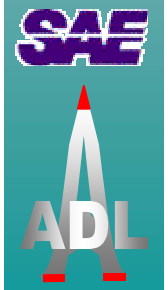
The Software Engineering Institute

# Subcontractor Management

- Subcontractor specific name spaces
- Separation of public and private package sections

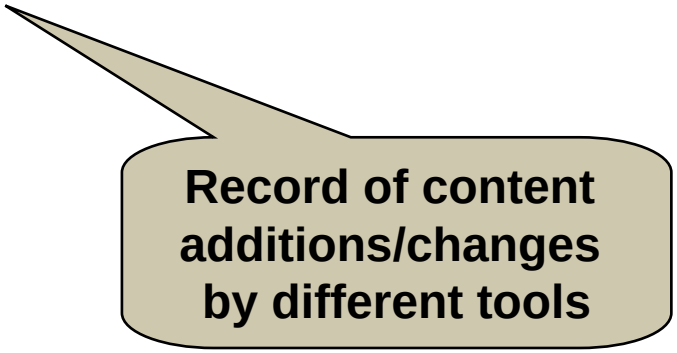
```
package edu::cmu::sei::redundancypatterns
public
-- publically accessible component interfaces
       system PrimaryBackupPattern
       -- features
       end PrimaryBackupPattern;
end edu::cmu::sei::redundancypatterns;

-- separate file
package edu::cmu::sei::redundancypatterns
private
-- source not accessible to outsiders
end edu::cmu::sei::redundancypatterns;
```

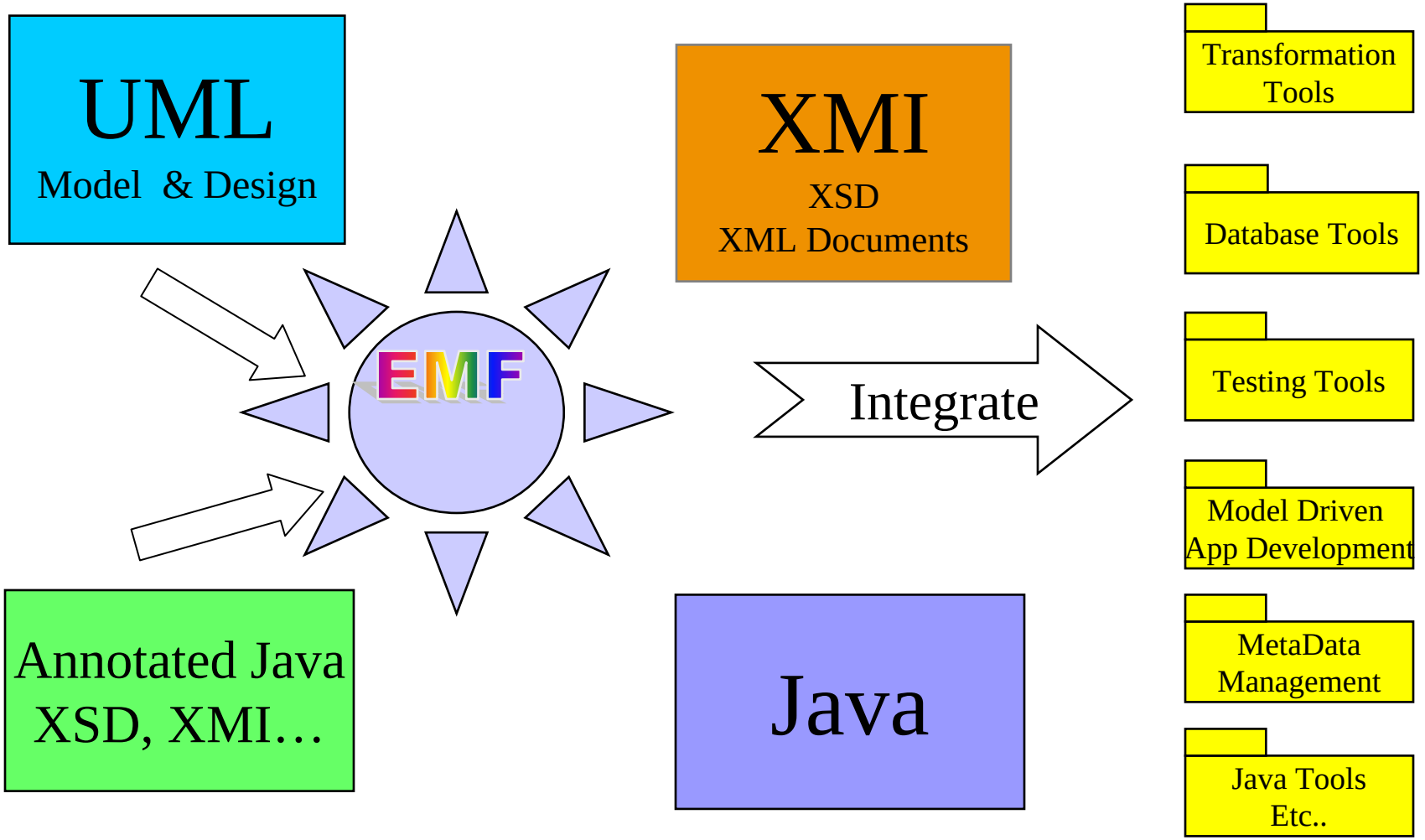


# Defining the XML Representation

- XML Data Type Definition (DTD)
  - Define tags & hierarchical structure
- XML Schema
  - Better handling of references & typing
- XMI
  - Meta model definition
  - XML documents with document change information



# Leverage of Existing Technology

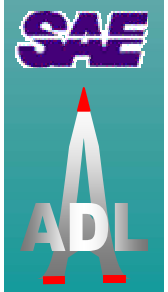



**EMF - The Data Integration Foundation of Eclipse**

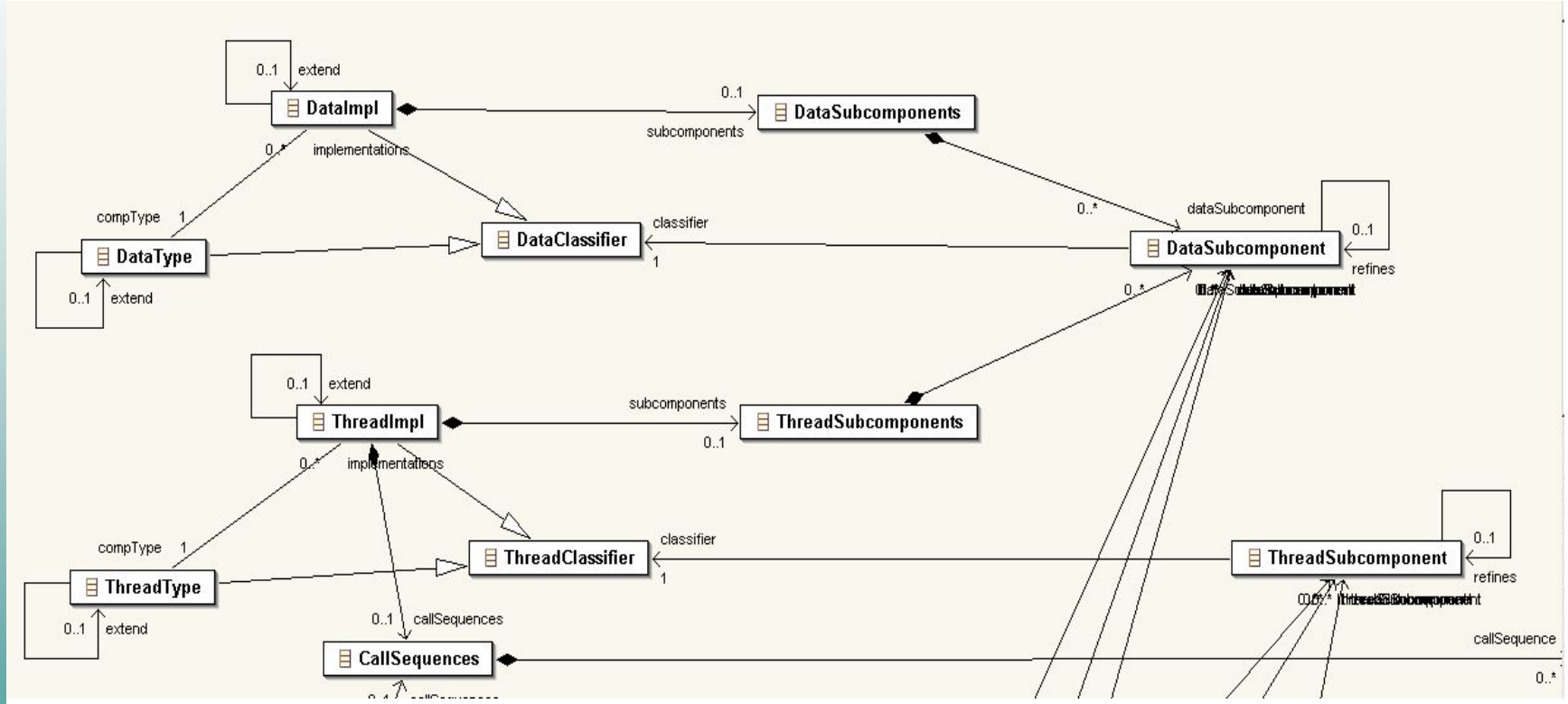
The Software Engineering Institute

# AADL Meta Model

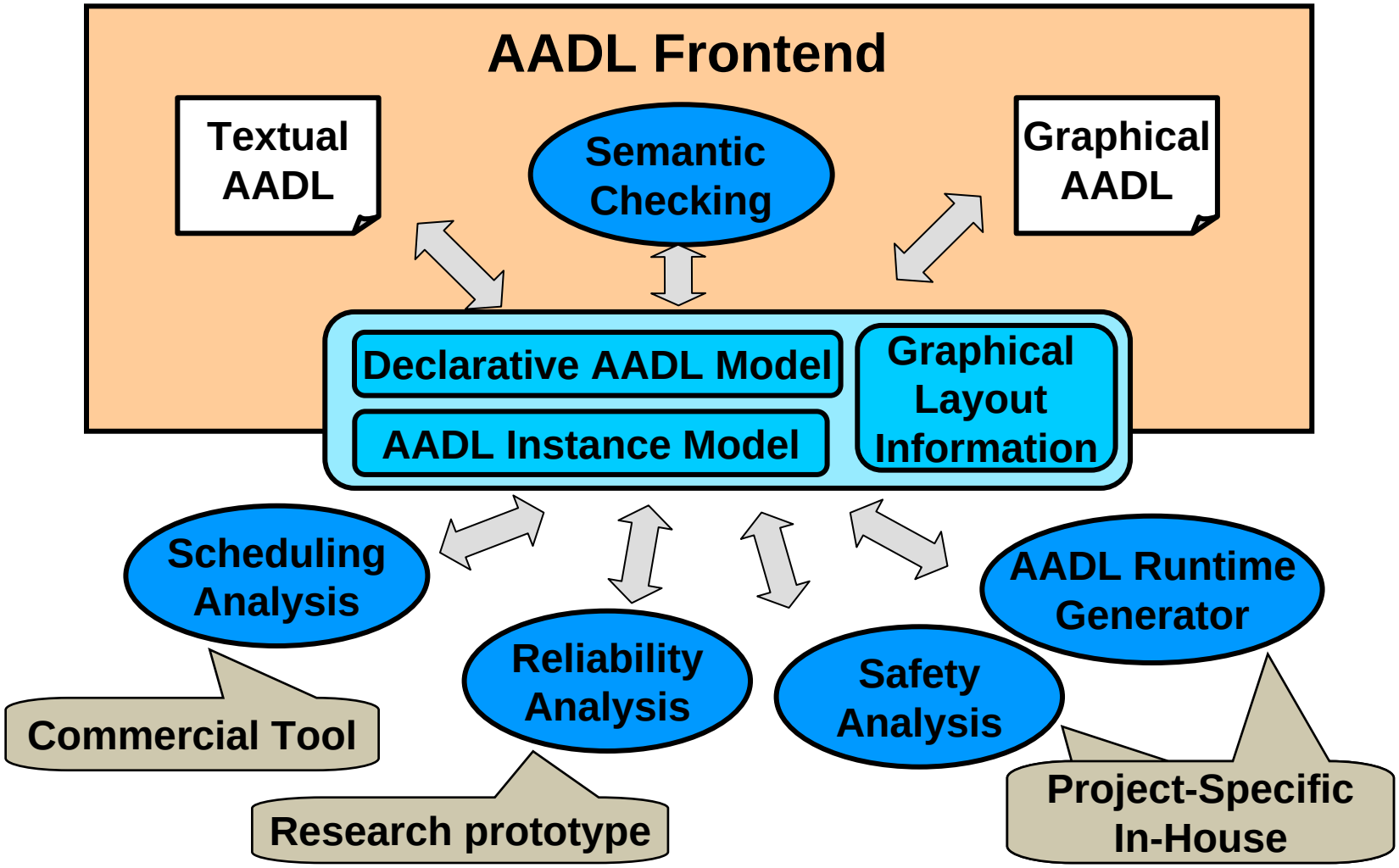
- Defined in Eclipse Modeling Framework (EMF)
  - Collection of meta model packages with graphical views
  - Separate from, but close to UML profile of AADL
- XML as persistent storage
  - XMI specification from Ecore meta model
  - Generated XML schema
- In-core AADL model
  - Generated methods for AADL model manipulation
  - Edit history, deep copy, object editor, graphical editor
  - Methods to support
    - AADL extends hierarchy
    - feature “inheritance”
    - property value “inheritance”



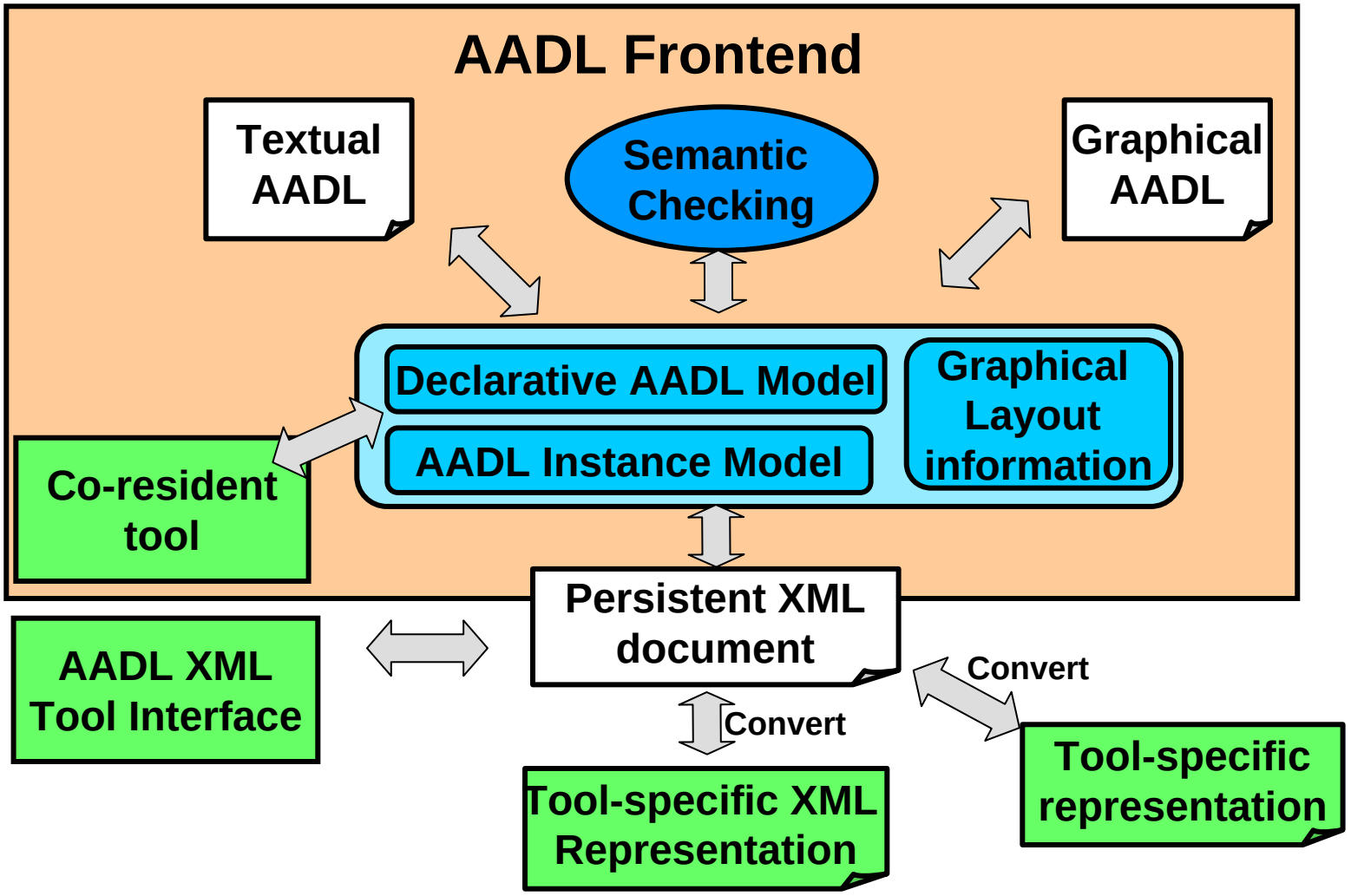
# AADL Meta Model Fragment



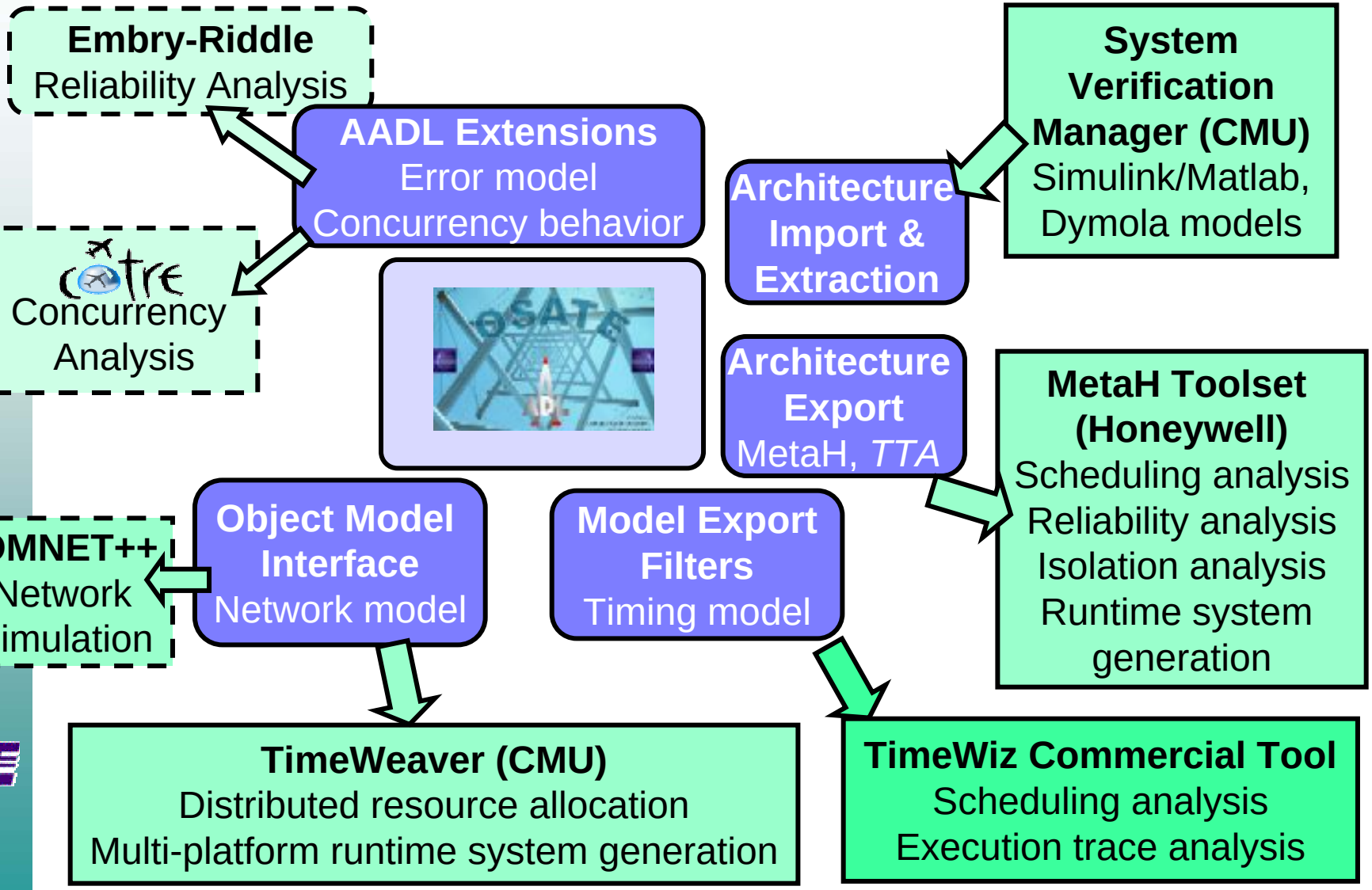
# Three Model Components



# Tool Interoperability



# An Open Source AADL Tool Environment



The Software Engineering Institute

# Final Observations

- Industry-standard architecture modeling notation for embedded systems
  - Abstract but precise modeling
  - Early and repeated predictable analysis
  - Architecture model libraries
- Industry-standard XML interchange format
  - Architecture model interchange between contractors
  - Lower cost plug'n'play AADL tools
  - Interoperability of analysis and generation tools