Software Architecture

- Components interact via connectors
- Connectors enforce interaction constraints
- Configurations reflect topological constraints
- Often specified in (formal) ADLs
Architecture-Level Modeling/Analysis

- Specify component interfaces and behaviors
  - provided and required services
  - invariants
- Allow partial specification and matching
  - suitable for OTS integration
- Exploit heterogeneous connectors
  - middleware-based connectors
- Generate (partial) implementations

Unified Modeling Language: Motivation

<table>
<thead>
<tr>
<th>Academic Approach to Architectures</th>
<th>Industrial Approach to Architectures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus on analytic evaluation of architectural models</td>
<td>Focus on wide range of development issues</td>
</tr>
<tr>
<td>Individual models</td>
<td>Families of models</td>
</tr>
<tr>
<td>Rigorous modeling notations</td>
<td>Practicality over rigor</td>
</tr>
<tr>
<td>Powerful analysis techniques</td>
<td>Architecture as the “big picture” in development</td>
</tr>
<tr>
<td>Depth over breadth</td>
<td>Breadth over depth</td>
</tr>
<tr>
<td>Special-purpose solutions</td>
<td>General-purpose solutions</td>
</tr>
</tbody>
</table>
Unified Modeling Language:

Benefits

✓ Provides an economy of scale
  - more and better tools
  - improved tool interoperability
  - more skilled developers
  - lower training costs
✓ It is extensible
✓ Combine benefits of powerful, specialized notations with those of widely adopted, general notations
  - specific solution: “integrate” ADLs with UML

Integration Strategy:
UML Metamodeling Architecture
From Architecture to Implementation

Current Status

✓ Integrated environment for transforming C2-style architectures into UML