

Representing Product Line Architectures

André van der Hoek
Institute for Software Research
University of California, Irvine
andre@ics.uci.edu

Product Line Architecture

- Common architecture for a set of related products or systems developed by an organization
- Vehicle for organizational-wide reuse
- Document architectural alternatives
- Capture architectural evolution

The Problem

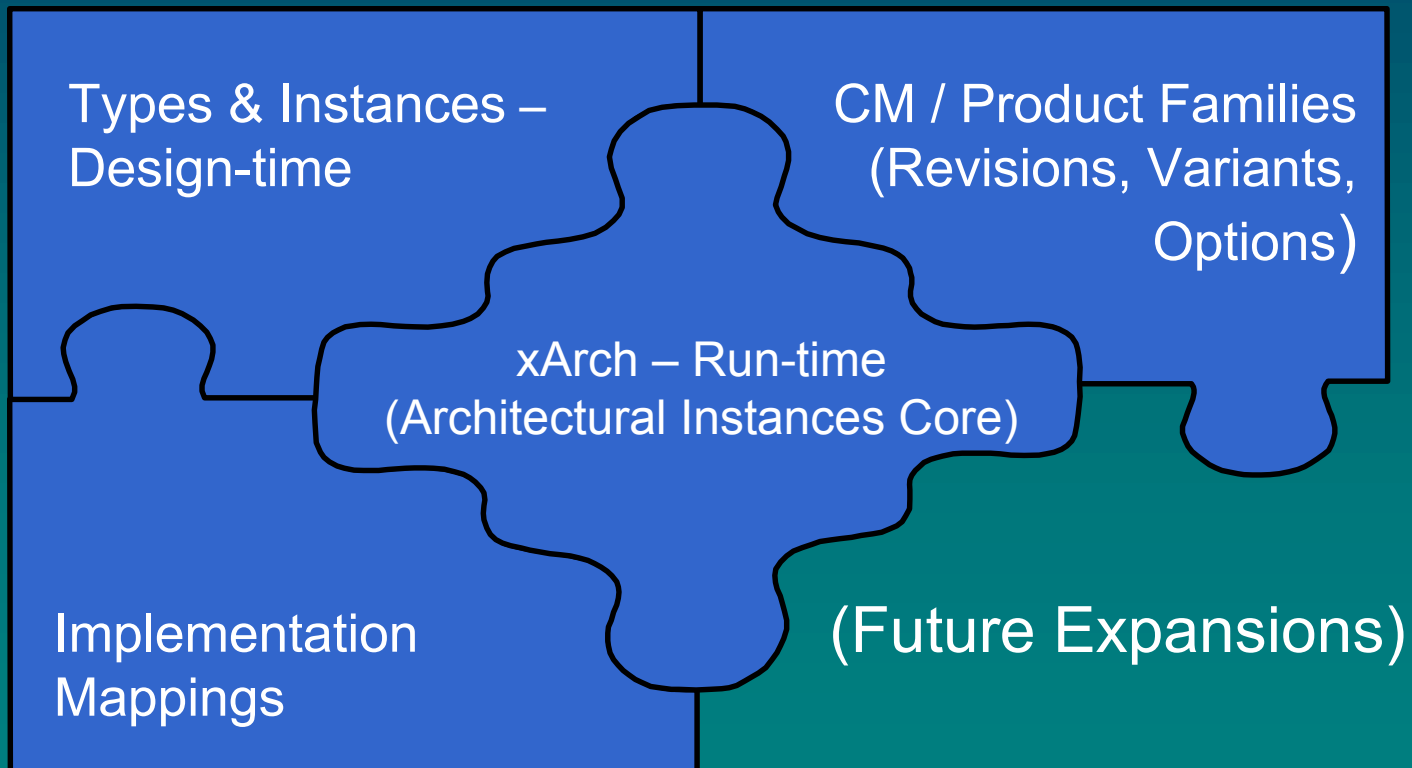
- Many ADLs are available for describing software architectures...
 - Acme
 - Wright
 - Darwin
 - Rapide
- ...but none can accurately and fully describe a product line architecture
 - (Koala)

Modeling Approach: Extensible ADL

- Encapsulate ADL features in individual XML schemas
 - Modularity
 - Extensibility
- Extend and/or modify existing features by adding new XML schemas
 - Allows tailoring of the resulting language to organizational specifics

xADL 2.0

- xADL 2.0 = xArch + modular extensions from UCI
- Each extension is a schema (100-500 lines XML)

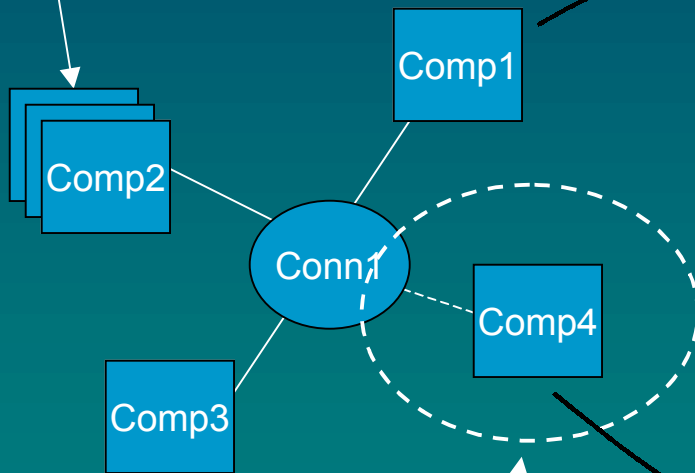


xArch: Architectural Instances Core

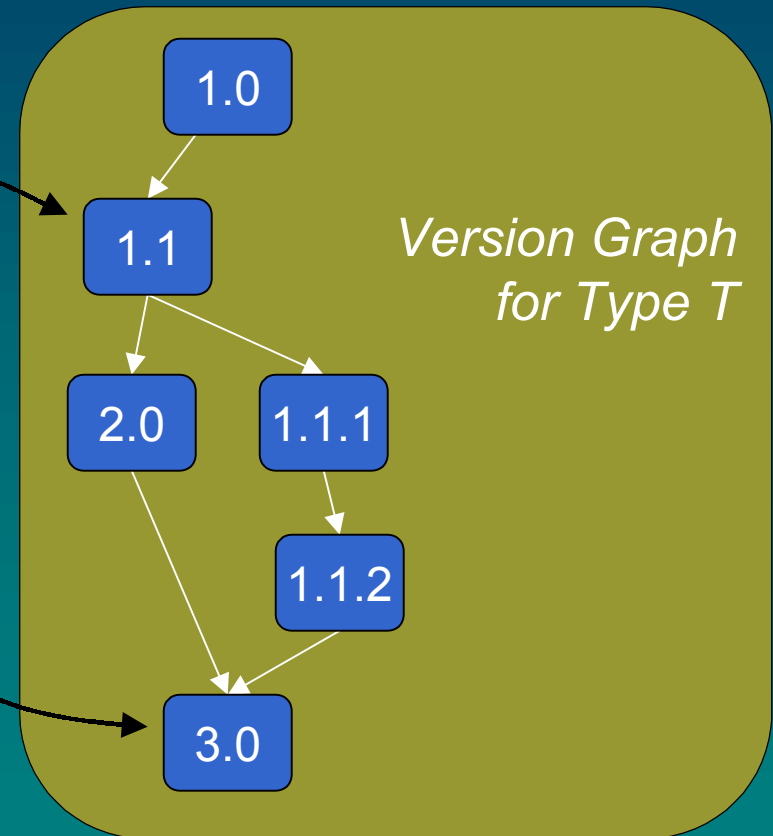
- xArch – the core of our new ADL
 - Written as an XML schema
 - Establishes the common set of architectural constructs
 - ❖ Components, connectors, interfaces, links
 - ❖ Sub-architectures (hierarchical composition)
 - ❖ General groups
 - Inherently distributed
 - Based on a collaboration with architecture researchers at CMU

CM / Product Family Architectures

Component
With Variant Type



Optional
Component & Link



*Version Graph
for Type T*

Design-Time vs. Run-Time

Behavior information
for static analysis

Design-oriented Properties

Author=André
Author=Dick
Last Update:
08/18/2001

Comp2

Comp1

Conn1

Comp3

Constraints

```
Invariant a{
  comp1.interface
  .type = top ->
  comp1.interface
  .link.type =
  bottom
}
```

```
Comp1_Beh{
  If(recv(evt(q))){
    doProcess(q)
    emit(evt(b));
  }
}
```

Run-time State

State = BLOCKED
Waiting on
event "A"

Comp
Inst 2

Comp
Inst 3

Conn
Inst 1

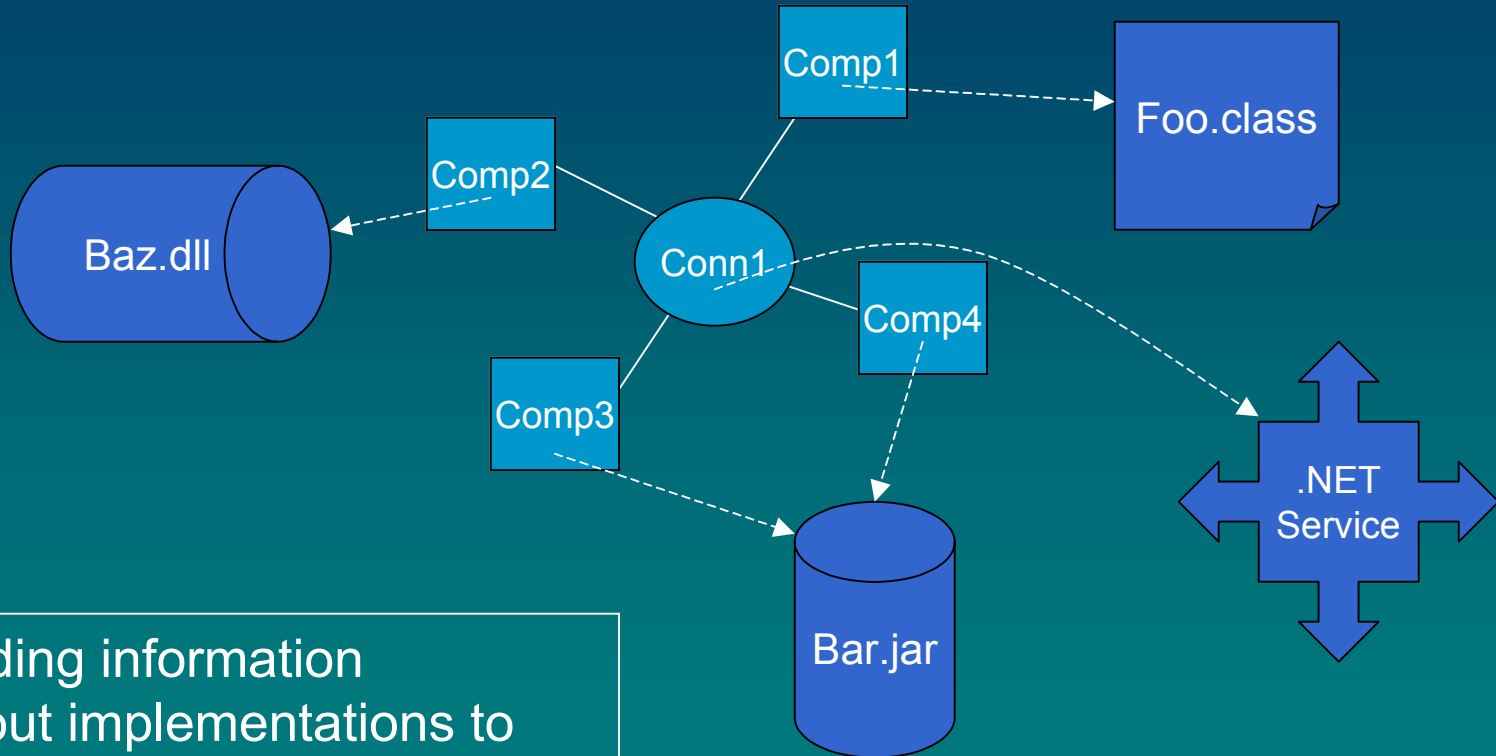
Machine = magister
Pid = 242
CPU = 1
Port = 8080
...

Information about
distributed components

Event queue contents

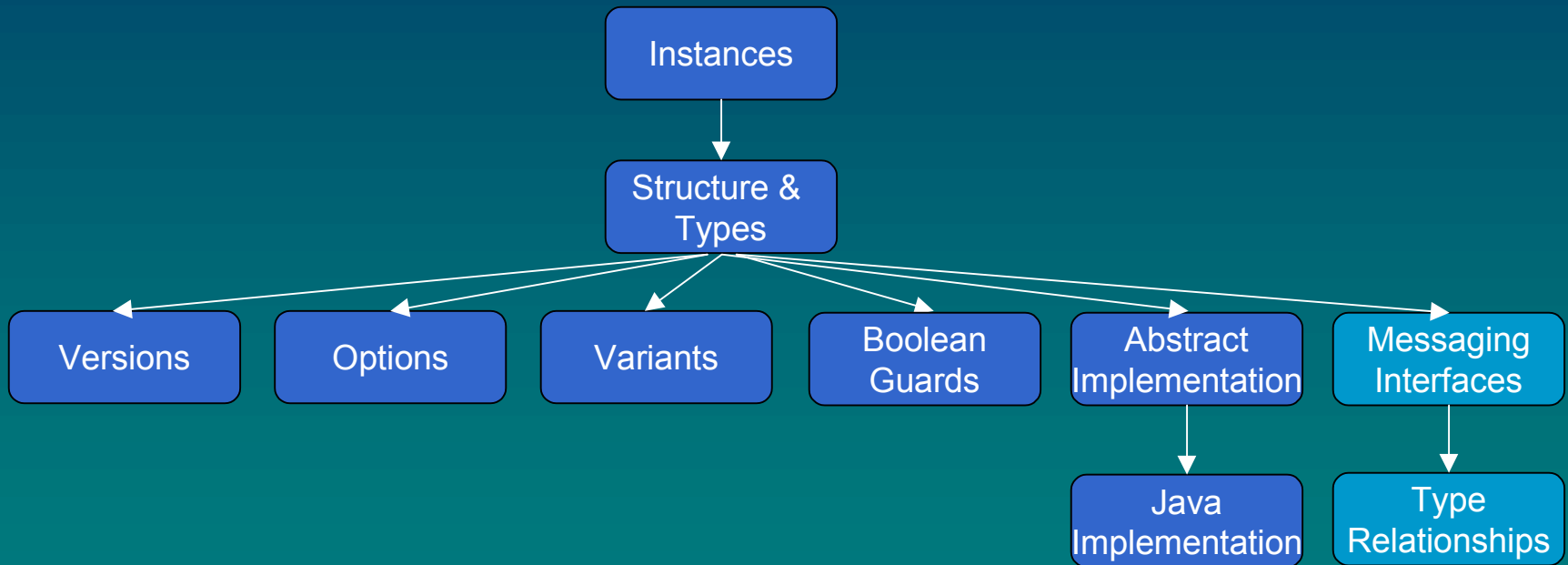
a b a - -

Implementation Mappings



Adding information about implementations to component, connector, and interface types is *essential* if the architecture is to be instantiated.

Current Set of XML Schemas



Tool support

- DOM-based Java Libraries
 - Programmatic, syntax-directed editing of xArch/xADL 2.0 documents, hiding nearly all of XML
- Apigen
 - Generates DOM-based Java Libraries using only the XML schemas
- ArchEdit
 - Syntax-directed graphical tree-based editor for xArch/ xADL 2.0 documents
- “Visio for xADL”
 - Microsoft Visio extensions provide full graphical editing capabilities for xADL 2.0 architectures
- Ménage
 - Design environment with specific support for product line architectures

Experience & Evaluation

- Lockheed Martin Systems Integration
 - AWACS aircraft software systems modeled in 10,000 lines of xADL 2.0
 - Used as the basis for an architecture-derived simulation of the inter-component communication on AWACS
- Jet Propulsion Laboratories (JPL)
 - Extended xADL 2.0 to add domain-specific interface descriptions
 - Experimenting with modeling software architectures in xADL 2.0 for use in future Mars missions

Visit the Web Site

<http://www.ics.uci.edu/projects/xarchuci/>