A Comparative Analysis of Architecture Description Languages

Nenad Medvidovic

Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
neno@usc.edu
http://sunset.usc.edu/~neno/

Presentation Overview

- Background and Motivation
- ADL Classification and Comparison Framework
- ADL Classification and Comparison
- Discussion and Conclusions
The Nature of ADLs

- Architectural models as distinct software artifacts
  - communication
  - analysis
  - simulation / system generation
  - evolution
- Informal vs. formal models
- General-purpose vs. special-purpose languages
- Several prototype ADLs have been developed
  - ACME
  - Darwin
  - Aesop
  - LILEANNA
  - UniCon
  - ArTek
  - MetaH
  - Weaves
  - C2
  - Rapide
  - Wright
- Many open questions remain

ADL Definition and Comparison Framework

- An ADL is a language that provides features for modeling a software system's conceptual architecture.
  - Essential features: explicit specification of
    - components
    - component interfaces
    - connectors
    - configurations
  - Desirable features
    - facets of components, connectors, and configurations
    - tool support
Classifying Existing Notations

- Approaches to modeling configurations
  - implicit
  - in-line
  - explicit

- Associating architecture with implementation
  - implementation constraining
  - implementation independent

- Related Notations
  - high-level design notations
  - module interconnection languages (MIL)
  - object-oriented notations
  - programming languages
  - formal specification languages

ADL Components

- A component is a unit of computation or a data store. Components are loci of computation and state.
- ADLs may use different terminology
- Component Classification Categories
  - Interfaces
  - Types
  - Semantics
  - Constraints
  - Evolution
A Comparative Analysis of Architecture Description Languages

ADL Connectors

- A **connector** models interactions among components and rules that govern those interactions.
- ADLs may use different terminology
- Some ADLs do not treat connectors as first-class entities

Connector Classification Categories

- Interfaces
- Types
- Semantics
- Constraints
- Evolution
A Comparative Analysis of Architecture Description Languages

An architectural configuration is a connected graph of components and connectors that describes architectural structure.

- Configurations help ensure architectural properties
  - proper connectivity
  - concurrent and distributed properties
  - adherence to design heuristics and style rules

- Configuration Classification Categories
  - Understandability
  - Compositionality
  - Heterogeneity
  - Constraints
  - Refinement and Traceability
  - Scalability
  - Evolution
  - Dynamism

Annual Research Review, USC, February 8-9, 1999

Neno Medvidovic
A Comparative Analysis of Architecture Description Languages

ADL Tool Support

- Formality of ADLs enables their manipulation by tools
  o Toolset is not part of an ADL
  o Usefulness of an ADL depends on its tool support
- Every ADL provides some tool support
- Focus typically on a particular area and/or technique
- Limited overall support motivated ACME
- Tool Support Classification Categories
  o Active Specification
  o Multiple Views
  o Analysis
  o Refinement
  o Code Generation
  o Dynamism

Annual Research Review, USC, February 8-9, 1999
Neno Medvidovic
Conclusions and Future Work

- Value of the framework
  - distinguishes different kinds of ADLs
  - definition is a simple litmus test
  - aids development of new ADLs
  - aids architecture interchange

- Existing ADLs
  - several ADLs straddle the boundary
  - extensive support in certain areas, lacking in others

- The framework is modifiable and extensible
  - completeness of categories
  - additional issues to address
  - ADLs are a moving target