Domains of Concern in Software Architectures and Architecture Description Languages

Nenad Medvidovic and David S. Rosenblum

Department of Information and Computer Science
University of California, Irvine
Irvine, California 92697-3425
{neno,dsr}@ics.uci.edu

Presentation Overview

- Background and Motivation
- Introduction to Architectural Domains
- Overview of Architectural Domains
- Discussion and Conclusions

Why Architecture?

- Architecture — a key to reducing development costs
- A shift in developer focus
- Component-based development philosophy

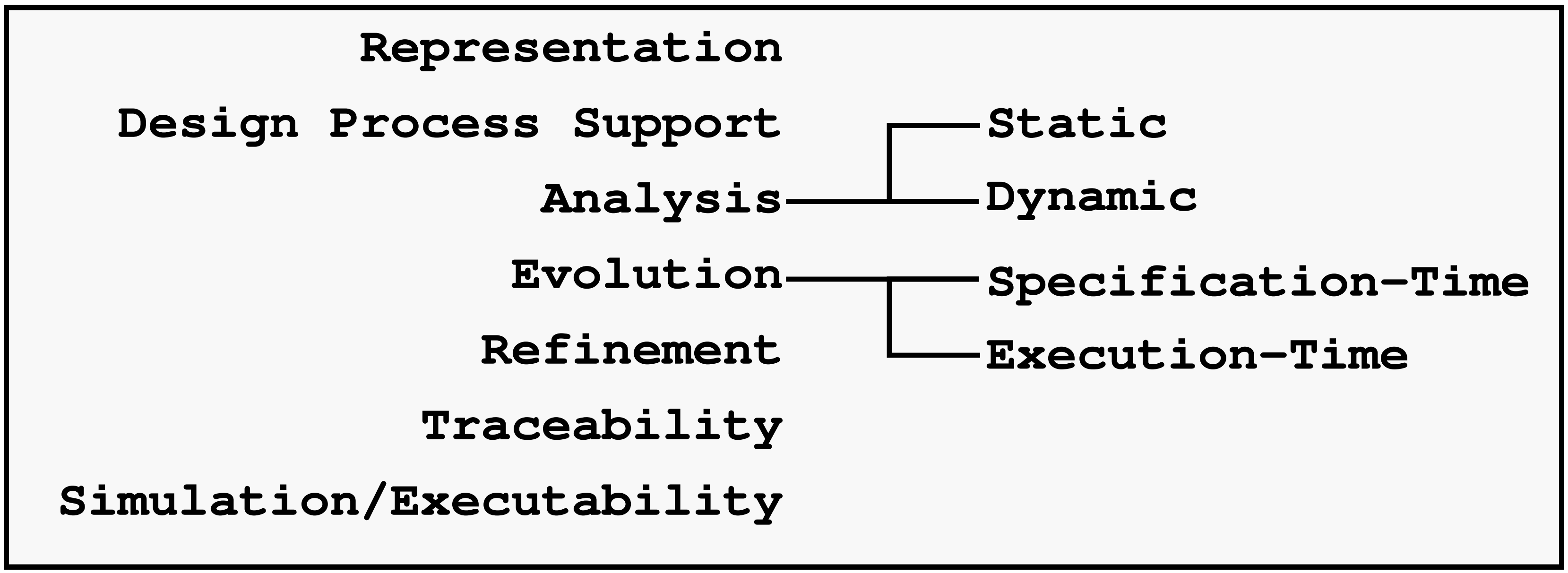
Why ADLs?

- Architectural models — distinct software artifacts
- Informal vs. *formal* models
- General-purpose vs. *special-purpose* languages
- Several prototype ADLs have been developed

What problems are most important to address in the study of software architecture?

Architectural Domains

- Classes of problems or areas of concern in architecture
  - need to be addressed by ADLs
- Relating architectural domains and ADLs
  - ADL conceptual frameworks are key to the relationship
  - focus often on a particular architectural domain



Representation

- **Principal Problems**
  - Aid stakeholder communication and understanding
- **Desired Solutions**
  - Multiple perspectives
- **ADL Support**
  - Explicit configuration
  - Graphical notations
  - Some additional views

Design Process Support

- **Principal Problems**
  - Decompose large, distributed, heterogeneous systems
- **Desired Solutions**
  - Multiple perspectives
  - Design guidance and rationale
- **ADL Support**
  - Active specification
    - proactive vs. reactive
    - non-intrusive vs. intrusive

Analysis (1)

- **Principal Problems**
  - Evaluate system properties upstream to reduce number and cost of errors
- **Desired Solutions**
  - Static analysis
    - internal consistency
    - concurrent and distributed properties
    - design heuristics and style rules
  - Dynamic analysis
    - testing and debugging
    - assertion checking
    - runtime properties

Analysis (2)

- **ADL Support**
  - Static analysis
    - parsers, compilers, model checkers
    - schedulability and resource utilization
    - design critics
    - refinement correctness
  - Dynamic analysis
    - scenarios
    - simulations
    - event visualization and filtering
    - dynamic constraint enforcement

Evolution (1)

- **Principal Problems**
    - Evolution of design elements
    - Evolution of system families
- **Desired Solutions**
    - Specification-time evolution
        - subtyping
        - incrementality
        - system families
    - Execution-time evolution
        - replication, insertion, removal, and reconnection
        - planned or unplanned
        - constraint satisfaction

Evolution (2)

- **ADL Support**
  - Specification-time evolution
    - subtyping
    - explicit and flexible connectors
    - support for partial architectures
    - explicit specification of application family
  - Execution-time evolution
    - “constrained” vs. “pure” dynamism
    - conditional configuration
    - replication
    - constraint enforcement

Refinement

- **Principal Problems**
  - Bridge the gap between informal diagrams and programming languages
- **Desired Solutions**
  - Specify architectures at different abstraction levels
  - Correct and consistent refinement across levels
- **ADL Support**
  - Correctness-preserving mappings
  - Comparative simulations of mapped architectures
  - Constraining the implementation

Traceability

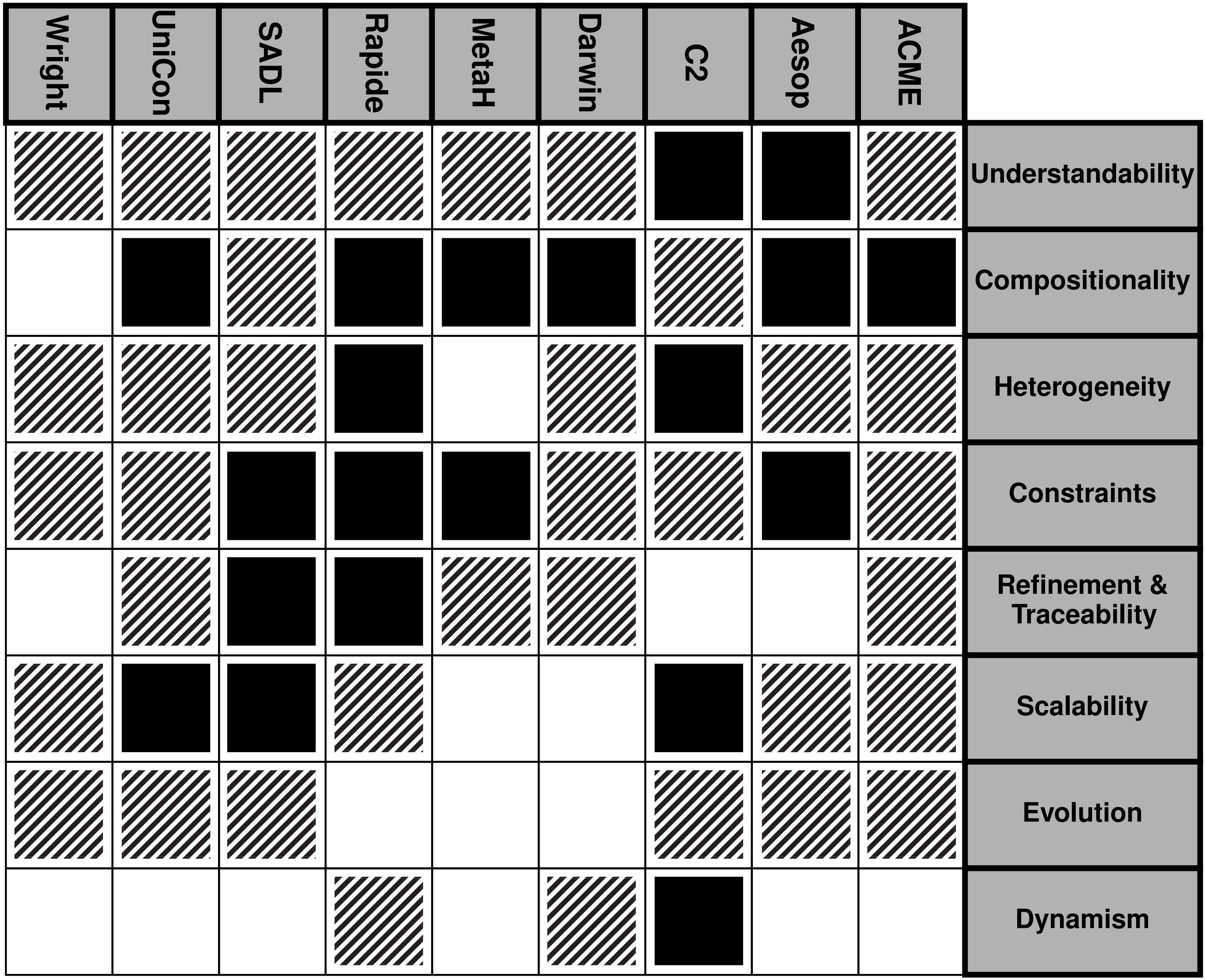
- **Principal Problems**
  - Multiple abstraction levels **+** multiple perspectives
- **Desired Solutions**
  - Traceability across architectural cross-sections
- **ADL Support**
  - Relationships among architectural perspectives is often poorly understood
  - Mapping from requirements to architecture is an open issue
  - Particular problem with *implementation-constraining* ADLs

Simulation/Executability

- **Principal Problems**
  - Some dynamic properties require a running system
  - Early prototypes to demonstrate features to stakeholders
- **Desired Solutions**
  - Construct simulations
  - Principled support for system generation
- **ADL Support**
  - Simulation by generating event sequences and partially-ordered sets
  - Component implementations
  - Abstract class frameworks

Discussion and Conclusions

- Architecture is concerned with several domains
- ADLs reflect properties of one or more domains
- Architectural domains — a unifying view of the discipline
- ADL support varies across domains
  - extensive in some domains, sparse in others
- Extensive support implies
  - important architecture modeling issues
  - simple problems
- Sparse support implies
  - difficult problems
  - areas of future concentration


Wright
UniCon
SADL
Rapide
MetaH
Darwin
C2
Aesop
ACME
Understandability
Compositionality
Heterogeneity
Constraints
Refinement & Traceability
Scalability
Evolution
Dynamism