ISR Institute for Software Research
UNIVERSITY OF CALIFORNIA, IRVINE

Architectural Runtime Configuration Management (WADS ’05)

John Georgas, André van der Hoek, and Richard Taylor

Institute for Software Research

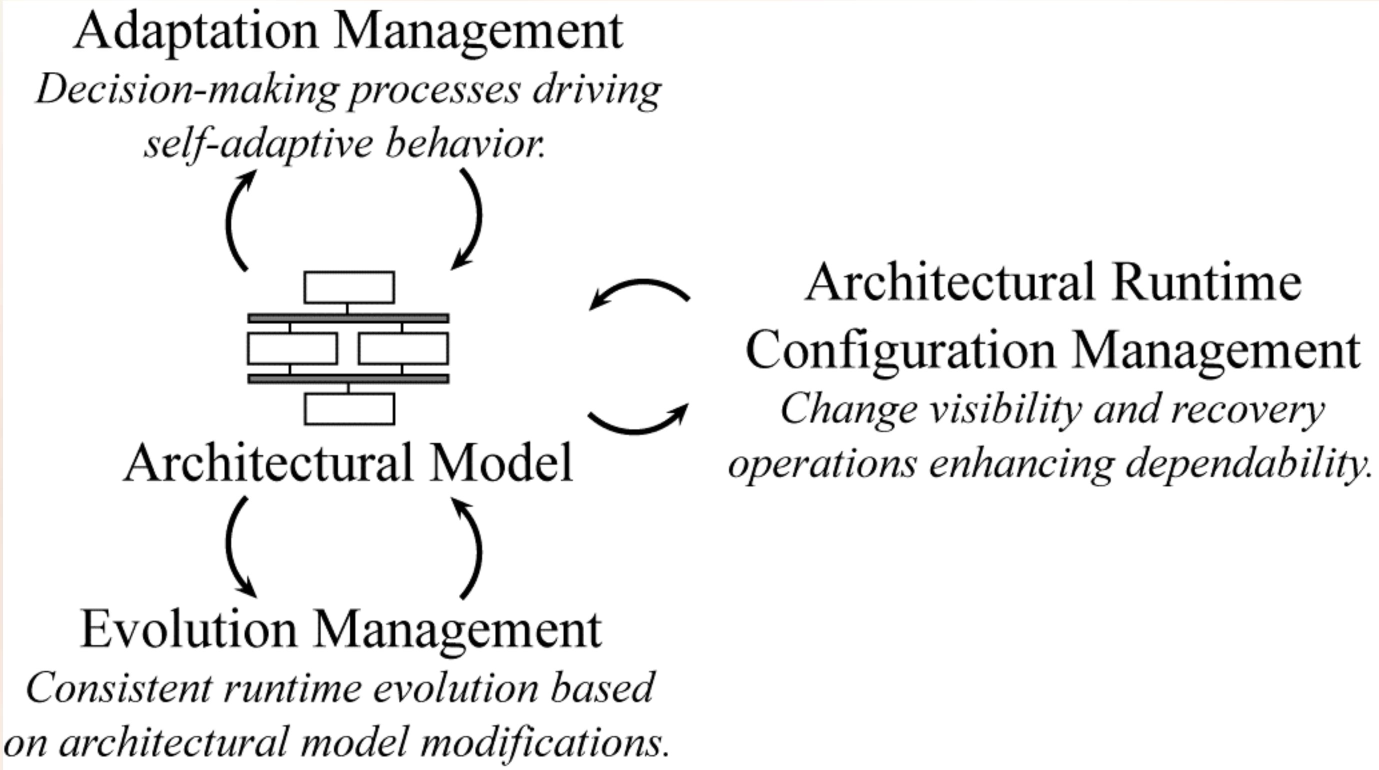
University of California, Irvine

May 17, 2005

http://www.isr.uci.edu/

Background: Self-Adaptive Systems

- Systems which autonomously adapt.



Adaptation Management

Decision-making processes driving self-adaptive behavior.

Architectural Model

Architectural Runtime Configuration Management

Change visibility and recovery operations enhancing dependability.

Evolution Management

Consistent runtime evolution based on architectural model modifications.

Background: Fundamental Assumptions

- Explicit architectural models:
  - Evolution and adaptation through these models.
- Out of scope:
  - Decision-making processes guiding adaptations.
  - State restoration and/or transfer.
  - Quiescence before modifications.
  - Architectural invariants throughout adaptation.

Motivation

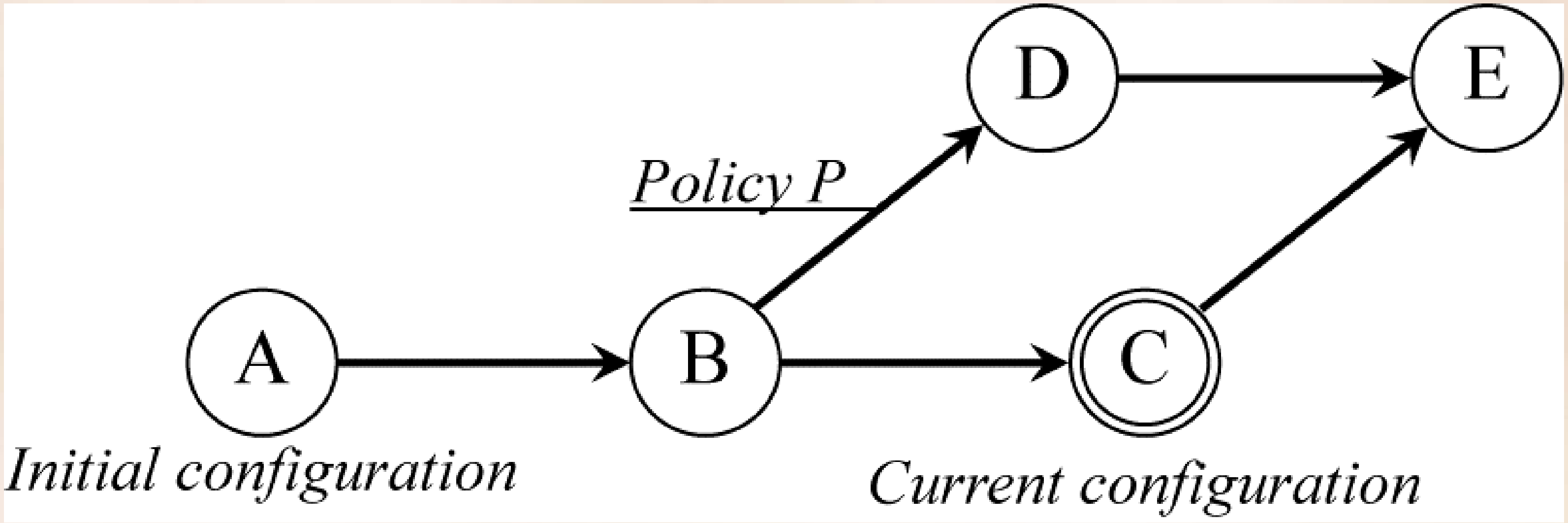
- Low visibility and independent nature of self-adaptive systems *diminish* trust in the adaptation process.
  - Opaque adaptation processes.
  - Behavioral changes only adaptation indicators.
- Dynamic self-adaptive systems can change in unpredictable ways.
  - Dynamic policy-based systems.
- Perceived dependability of the adaptation process.

Approach

- ***Architectural Runtime Configuration Management (ARCM)***
- **Key Features:**
  - Runtime *monitoring* of architecture-based self-adaptive systems.
  - Maintaining a *runtime configuration* version graph.
  - Graphical *visualization* of version information.
  - *Operations* for user-driven fault *recovery*.

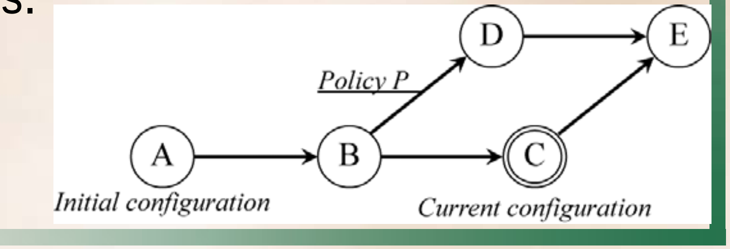
Research Vision: Increasing Visibility

- Configuration version graph indicating adaptations.
  - Cycles, but no loops.
  - Single edge between configurations; anti-parallel.
- Links to policies which cause adaptation.



Research Vision: Increasing Visibility, continued

- Adaptation awareness:
  - Explicit recording of any adaptations in a configuration graph.
  - Generated at runtime, as changes take place.
  - Adaptation history throughout system lifetime.
- Graphical visualization of the configuration graph:
  - Intuitive and easy to understand artifact.
- Enhanced visibility:
  - Reduces the opaque nature of adaptation process.
  - Allows additional questions about systems.
  - Increases trust in the adaptation process.



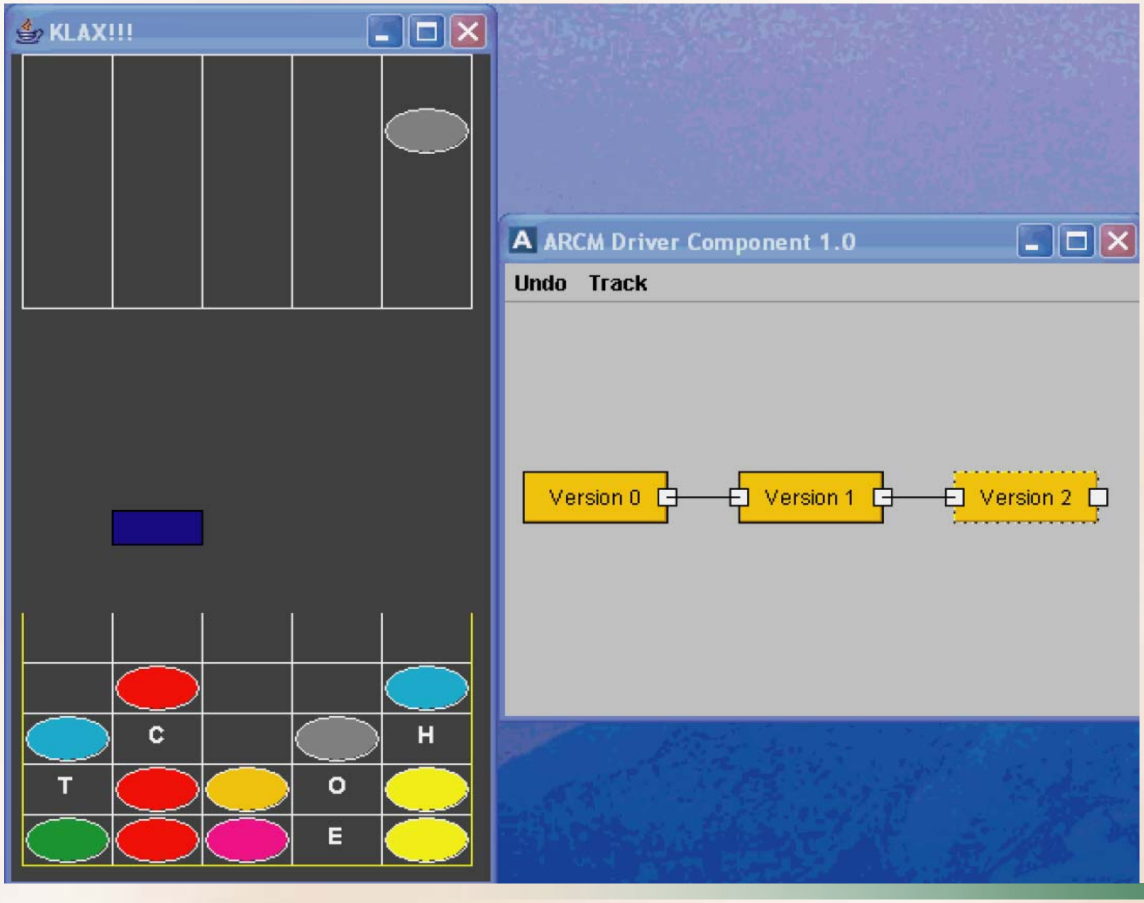
Research Vision: Recovery Operations

- Potentially undesirable adaptations necessitate recovery facilities.
    - Desirability determined by architect.
- Recovery Operations:
    - *Rollforward*
        - Transition in the direction of a graph edge.
    - *Rollback*
        - Transition against the direction of a graph edge.
    - *Out-* and *in-degree* > 1 require user selection.
- These operations provide for user intervention into the self-adaptive process.
    - Leveraging architect expertise.

Prototype Tool Support: *ARCM Driver*

- Integrated into the ArchStudio development environment.
- Observes and monitors systems for runtime adaptations.
- Builds configuration version graph:
  - Records pre- and post-adaptation configuration.
  - Stores bi-directional *diff* files.
- Provides graphical visualization of the version graph.
- Recovery operations:
  - Merges graph's *diff* information for operation enactment.
    - *Diffing* and merging facilities already present.
    - System architecture is evolved by AEM.

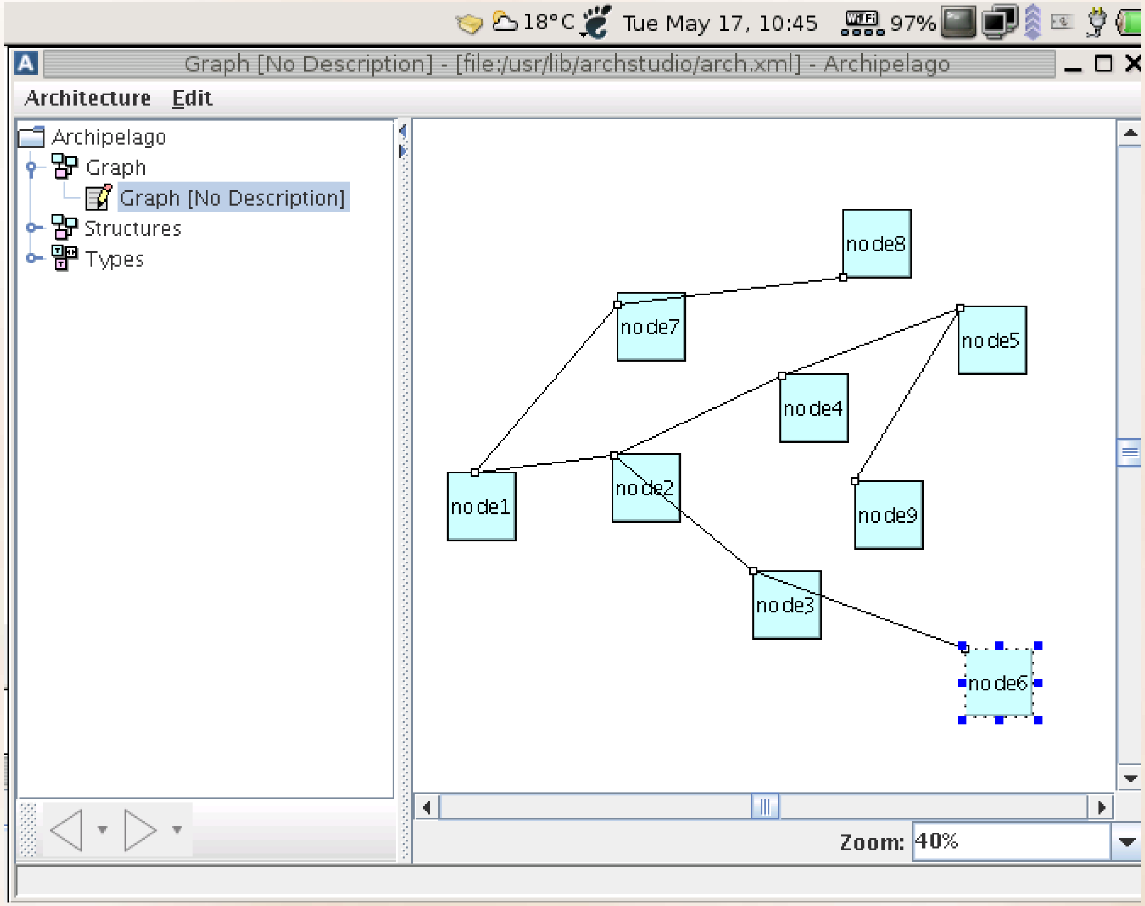
Prototype Tool Support: *ARCM Driver* Screen Capture



Prototype Tool Support: Under Development

- Refined implementation:
  - Transition to xADL schema for graph data (XML-based).
  - Enhanced graphs with support for multiple branching.
  - Identification of duplicate nodes.
    - Architectural configuration hashing.
  - Arbitrary graph transitions.
    - Allows for multi-step recovery operations.
    - *Diff* composition.
- Better visualizations:
  - Integration with Archipelago, the ArchStudio visual editor.
  - Graph layout with DOT.

Prototype Tool Support: Just in...



Future Research Directions

- Further graph annotations:
  - Rejected configurations with counts.
  - Time spent in each configuration.
- Explore automated detection of desirability.
  - Architectural configuration patterns.
- Closer integration with adaptation process:
  - Use recovery operations as an active reflection layer.
  - Include recovery operations into adaptation management decision-making for automated invocation.
  - Leverage graph information in decision-making processes.

Conclusion

- *ARCM:*
  - Maintains a record of adaptation history.
  - Enhances the visibility of adaptations.
  - Provides user-driven fault-recovery facilities.
- Increases in perceived dependability through increased visibility and transparency of the adaptation process.
- Fully decoupled from specific adaptation management and enactment methods.
- Under active development; a new, fully-featured version is expected to be released soon.