
Ground Systems Architecture Workshop 2000
February 23-25, 2000

Hoh In (hohin@cs.tamu.edu)
http://www.cs.tamu.edu/faculty/hohin
Computer Science Department
Texas A&M University
Contents

- Introduction
  - What is PnA?
  - What are Challenges?
  - What are Solution Approaches?
- PnA Overview
- Conclusion
  - For Further Information
What is PnA (Plug-n-Analyze)?

- An Integrated Framework for Architecture Tradeoff Analysis Tool Support

- Plug-In individual tools into the framework and Analyze architecture tradeoff points to streamline architecture generation, evaluation, negotiation, and selection

- The underlying assumption is that...
  - Organizations have their own analysis tools, but there is no compatibility with others (inside or outside the organizations)
Challenges

- Difficulties in coordinating multiple stakeholders’ interests and priorities in determining architecture alternatives
- Complicated dependencies of quality attributes in architecture tradeoff analysis
- Isolation of independent analysis tools
- Huge option space of architecture alternative generation
Solution Approaches

- To surface and negotiate architecture alternatives based on supported quality attributes -- WinWin
- To analyze and visualize the complicated dependencies
  - Aids for analyzing and managing the complicated dependencies of quality attributes
  - Aids for visualizing the complicated dependencies to identify tradeoff issues
- To integrate independent analysis tools
  - Aids for integrating independent analysis methods
- To generate potential architecture options
  - Aids for generating architecture options
Contents

- Introduction
- PnA Overview
  - How does PnA work?
  - What aids does PnA have?
  - What does PnA help?
- Conclusion
  - For Further Information
What Aids Does PnA Have?

1. Architecture Generation Aids
2. Architecture Evaluation Aids
3. Architecture Integration Aids
4. Architecture Visualization Aids
5. Architecture Negotiation/Decision Aids
6. Architecture Management Aids
1. Architecture Generation Aids

- Available Approaches/Techniques/Models:
  - Reference architectures/DSSA: Customization/Specialization
  - Architectures styles and Design Patterns: Composition
  - CMU/SEI Product Line Practices

- Our Approach:
  - UML-based, reference architecture generation

- Issues/Challenges
  - Approaches: Fully-automatic, Manually, Semi-automatic
  - Initially proposed architecture vs. revised architecture
  - Validation and verification of generated architecture (based on rqmtrs)
PnA Prototype
3. Architecture Integration Aids

- Available Approaches/Techniques:
  - APIs; Glues; Protocols; XML; Middleware (e.g., CORBA, COM/DCOM)

- Our Approaches: XML-based tool integration
  - IEESIM (Integrated, Exchangeable, Extended, Shared Information Mediator)

- Issues/Challenges
  - Automation or semi-automation of wrapping process
  - Information Fusion of inconsistent analysis results
  - Integrated ontology from individual tools
XML-Based Tool Integration Mediator

- **IEESIM**
  - Integrated, Exchangeable, Extended, Shared Information Mediator
    - *Integrated* views (data schema) from local views
      - For a global view for tool users; transparency of location
    - *Exchangeable* information format
      - Based on XML
    - *Extendable* interface for additional tools
      - Easy to add additional database into the shared database
    - *Shared* information
Centralized IEESIM Architecture

XML converter, Communication Protocol, Transaction Management, Data Integrity Management, SQL Query, Optimization, View Integration Support

Analysis Tool 1

Analysis Tool 2

Analysis Tool 3

XML Applications (based on the Shared database)

Shared Database

XML Server

Internet

XML Transmitter

XML Transmitter (= converter + protocol)

XML Transmitter
Distributed IEESIM Architecture

Diagram Source: Adapted from http://xmls.com/resources/whitepapers/co-existence.pdf
Inside XML Server

XML Server

XML Support Modules

1 Modules = XML converter, Communication Protocol, Transaction Management, Data Integrity Management, SQL Query, Optimization, View Integration Support
4. Architecture Visualization Aids

- Available Approaches/Techniques/Models:
  - Two-Dimension Visualization
    - Conflict, Non-Conflict Zone
  - Multi-Dimension Visualization
    - Acceptable level for each attributes
    - Desired level for each attributes

- Our Approaches:
  - Support Two- and Multi-Dimension Views

- Issues/Challenges
  - What should be visualized?
  - How to visualize? (e.g., two- and multi-dimension)
  - Visualization of impact analysis (e.g., snapshots)
Multi-Dimension View Support

Performance  Modifiability  Reliability  Cost & Schedule

Desired  Acceptable  Poor

Alternative 1

Alternative 2
Two-Dimension View Support

![Graph showing Performance vs. Reliability with regions labeled I to IV and specific values for RE (Reliability Efficiency)]
Overall Issues

- **Granularity of tradeoff points**
- **Scalability of Analysis**
  - Start with small amount of information
  - Be able to add more information and produce more accurate and detailed analysis
- **Flexible, but Predictable framework**
  - What information is necessary for what?
  - Be able to predict the consequences
  - Be general enough to customize their processes
What Does PnA Help?

Better, Cheaper, Faster

- to analyze architecture tradeoffs
  - By plugging analysis tools into the framework
- to understand tradeoff issues
  - By visualizing tradeoff points
- to manage changes
  - By managing architecture/requirements dependency
- to negotiate for a win-win Architecture
  - By capturing the underlying rationale (through WinWin)
- to generate architecture alternatives
  - By providing decision support
Conclusion

- Architecture tradeoff analysis needs not only architecture evaluation, but also architecture generation, integration, visualization, negotiation, and selection.

- Current Status (1 PhD and 1 MS students)
  - Design Architecture Integration Aids with JPL
  - Design Architecture Visualization Aids with JPL
  - Design Architecture Generation Aids
  - Looking for additional case study and/or funding
For Further Information

- **Phone** (409) 458-1547
- **Fax** (409) 847-8578
- **Email** hohin@cs.tamu.edu
- **US Mail** Hoh In
  Computer Science Department -- Mail Stop 3112
  Texas A&M University
  College Station, Texas 77843-03112

- **Web Information**
  - Class: [http://www.cs.tamu.edu/course-info/cpsc689/hohin](http://www.cs.tamu.edu/course-info/cpsc689/hohin)