Chris A. Mattmann, Ph.D.
Jet Propulsion Laboratory, California Institute of Technology /
University of Southern California / Apache Software Foundation

Big Data Architecture: Fundamentals

Sponsored by the Jet Propulsion Laboratory and by the Keck Institute for Space Studies, California Institute of Technology

Outline

- Introduction to Software Architecture
- Styles, Patterns, Reference Architectures
- Architectural Modeling, Visualization
- Architectural Drift and Recovery
- Case Study: Grid Computing
- Conclusions

Architectural Modeling

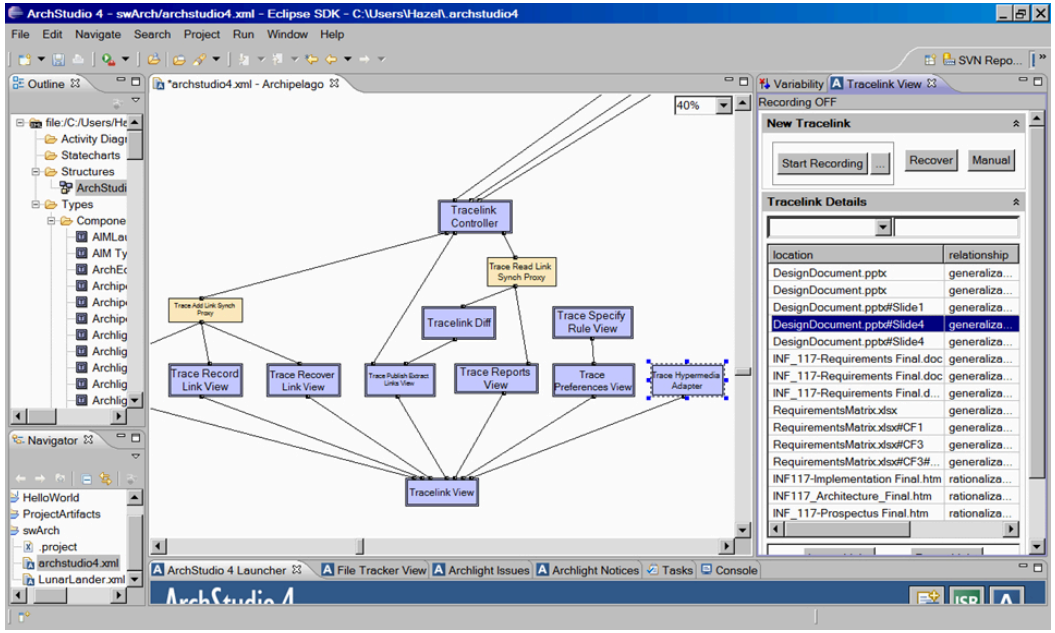
- Need ways of capturing principle design decisions
- Architectural models
  - Definition: captures and records the principle design decisions about a software system
- Several notations developed over the years during the 90s and 2000s (golden age of software architecture)
- Evaluating modeling notations: fidelity; ability to capture architectural structure, behavior, etc.

Modeling Notations

- Modeling Notations aka "Architectural Description Languages" or ADLs
  - Specific to architectural element
    - Components and structure – Darwin
    - Components, connectors and structure and effect – Rapide
    - Wright – components, connectors, structure, behavior
  - Specific to domains
    - Product lines – Koala
    - Avionics and system specifications - AADL

Modeling Notations – cont.

- Modeling Notations aka “Architectural Description Languages” or ADLs
  - Flexible and generic
    - ACME – developed by David Garlan’s group at CMU
    - ADML – XML + ACME
    - xADL – developed at UC Irvine in Dick Taylor’s group



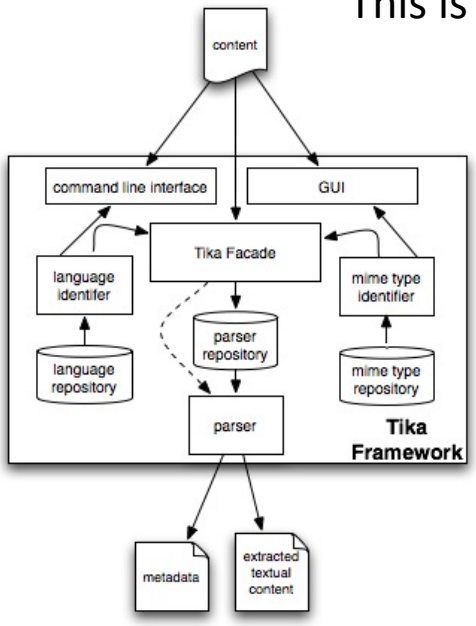
http://isr.uci.edu/projects/archstudio/whatis.html

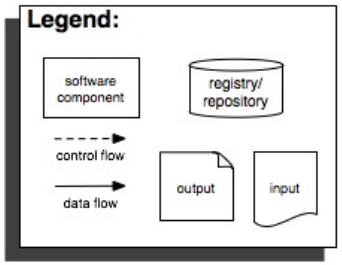
Architectural Visualization

- Two keys of Architectural Visualization
  - Depiction
  - Interaction (separates visualizations from simple drawings or static diagrams)
- Visualization Approaches
  - Textual, Graphical (PowerPoint), Hybrid (Unified Modeling Language), Effect

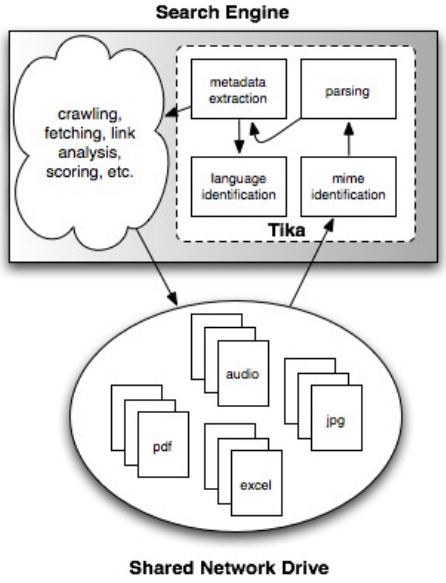
Two views of the same thing

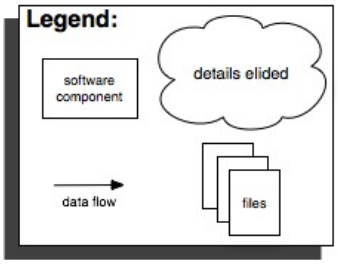
This is Tika





So's this





Architectural Views, and Viewpoints

Architectural Views and Viewpoints

- Architectural Model
  - Captures some or all of the principle design decisions about the software architecture
    - Model of the components (*Component model*); model of the interactions (*Dynamic model*), etc.
- Architectural View
  - Subset/collection of the architectural model design decisions – a *filter* on the model
- Architectural Viewpoint
  - Named collection of architectural model subset of design decisions (the *name* of the view)

Architectural Recovery

- Often, code will become adrift from the architecture
  - *Architectural drift* is when the as-implemented architecture (code) varies from the as-conceived (prescribed) architecture, but does not violate any of its core assumptions
  - *Architectural erosion* – same case as above, but the as-implemented architecture does in fact violate some core assumption of the as-conceived architecture
- Process for dealing with architectural drift and erosion = Architecture recovery

Architecture Recovery

- Recovering the architecture of the software system by automatically or semi-automatically processing the code base
  - Static Analysis
  - Dynamic (state) analysis
  - Component/Connector analysis
  - Mapping to architectural style
- Various approaches over the years, by e.g., Kazman et al 1999, Jakobac/Medvidovic et al/ Focus, etc.