Intent via Architecture Description

Dewayne E Perry
ENS 623A
Office Hours: T/Th 11:00-12:00
perry @ ece.utexas.edu
www.ece.utexas.edu/~perry/education/382v-s06/

Models of SW Architecture

- Perry & Wolf 89/92 model of SWA
- SWA = (Elements, Form, Rationale)
- Elements : process, data and connecting
- Form is the set of properties of, and relationships among, the elements
- Rationale is the justification for the elements and form

Styles

- An incomplete architectural prescription
- Focuses on certain aspects of the architecture
  - architectural elements
  - formal characteristics
  - constraints on architectural elements
  - constraints on formal characteristics
- Problem: Restrict the architectural structure
  - for example, strict layering of the architecture
- Solution: layered architecture style
  - constrain the interactions
    - any interaction at elements on the same level
    - no interactions at more than one level away
    - level below: initiate interactions only
    - level above: react interactions only

Styles

- Useful rule of thumb: a style for a domain
- Problem: multiple domains in any significant architecture
- Challenge: integrating the styles consistently

Current State

- State of Current Work
  - Pretty much agree about process, data and connecting elements as first class entities
  - Models differ primarily with respect to Form
  - Few models pay attention to rationale
  - Styles tend to focus on element and form restrictions
  - Current
- Approaches to Form
  - Configurations
  - Types
  - Patterns
  - Properties

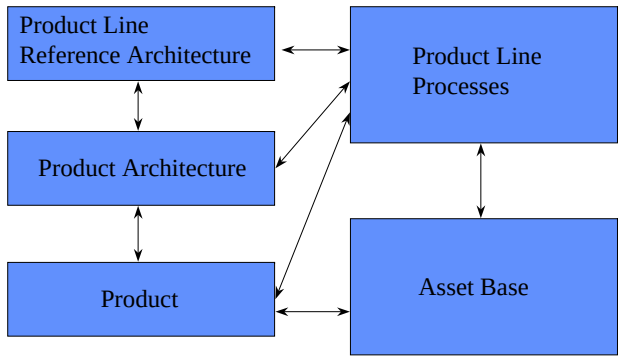
Current State

- Configuration as Form
  - Characterization
    - Basic box and lines approach
    - Components may be processes, subsystems, etc
    - Connections are defined by Provides/Requires clauses
  - Approach to Style
    - Tend not to be interested in styles
    - Except in the context of dynamic arch's
- Types as Form
  - Characterization
    - Typically, an historical approach
    - Look for types and classes of architectural objects
    - Often organized hierarchically
  - Approach to Style
    - Emphasis on the basic classes or types of components and connectors
    - Perhaps, a slight more emphasis on connectors
    - Eg, pipes and filters; blackboard architecture

Current State

- Patterns as Form
  - Characterization
    - Emphasis on patterns of interactions
    - Tendency to focus on connections with components as endpoints
  - Approach to Style
    - Architectural instances are specializations of styles
- Properties as Form
  - Characterization
    - Properties of (or constraints on) data, process and connecting elements
    - Relationships among data, process and connecting elements
  - Approach to Style
    - Selection of some critical elements
    - Selection of some properties and relationships
    - Constraints on properties and relationships

Product Line - Overview



Product Lines

- Basic Aspects
  - Begin with product instances
    - legacy based
    - use architecture recovery processes
  - Focus on appropriate business domain
    - use domain specific architectural processes
    - map from recovered to domain architecture
  - Abstract/Generalize to Product Line Architecture
- Issues
  - Product Line Reference Architecture
  - Product Line Processes
  - Asset Base
  - Supporting Technology
  - Organizational Issues

PLA Description Issues

- What generic features do you need
- Relationships between PLA and PIA
  - Derivation
  - Conformity
  - Analysis
  - Planning
- How is evolution of PLA supported
- Claim:
  - Generic descriptions are necessary for product line architectural descriptions

Generic PLD Approaches

- Style description
- Under-constrained description
- Variance-free description
- Parametric description
- Service/provision oriented description

Styles

- Summary
  - Intuitive appeal
  - Captures essential characteristics
    - basic components
    - minimum interactions
    - basic constraints
  - Ignores variation
- Advantages
  - Minimalist approach
  - Add new products easily
  - As long as they conform to style
  - Some project planning for the PLA applies to the product instance architecture (PIA)

Styles

- Disadvantages
  - Not easy to refine PLA into PIA
    - by extension, addition
  - PLA conformity analysis required
  - When PLA evolves, must revalidate PIA conformance
- Evaluation
  - Possible, but not adequate
  - better uses of styles than for PLA

Under-Constrained

- Summary
  - Difference in completeness
    - style focus: critical features, eliminate non-essential, non-stylistic
  - Capture PL as completely as possible
  - With variations not ruled out by overly constraining the architecture
  - Variance within constraints, not within the aspects not defined
- Advantages
  - Easier to create PIA from PLA than Styles
  - Analysis at PLA level applies to PIA level
  - Planning at PLA level applies to PIA level
  - Evolution via constraint relaxation easy

Under-Constrained

- Disadvantges
  - Extending the PLA is a significantly more constraining task
  - May not be possible to define all new products within constraints
  - PLA evolution may cause conformity problems
- Evaluation
  - Seems appropriate where primary difference is something like performance where the functionality remains the same
  - Too confining for variance often needed for individual products

Variance-Free

- Summary
  - Architecture is not under-constrained
  - Variance is not considered architecturally important
    - product difference a design or implementation issue not an architectural one
    - eg, platform or distribution independence
  - There are implications for the PLA
- Advantages
  - Analysis and planning at the PLA level
  - Product variance depends on implementation and not on architecture
  - PLA is the PIA
  - Evolution of the PLA means evolution of the PIAs

Variance-Free

- Disadvantages
  - Standard specification problem of talking about what is not there
  - May not be able to isolate all variance this way
- Evaluation
  - Useful for range of options for a particular aspect (eg, fault tolerance, distribution …)
  - But may not be able to account for variance in functionality

Parametric

- Summary
  - Standard approach: parametric abstraction
  - Limits depend on the constraints on the arguments
  - Defines a family of possible instantiations
- Advantages
  - Variations well-defined and well-known
  - Instantiation of PIA from PLA is well-understood (possibly automatic)
  - Analysis at PLA level
  - Planning at PLA level
  - Evolution by relaxing constraints or by upward compatible extensions OK

Parametric

- Disadvantages
  - Kinds of parameters allowed may seriously affect how well the PLA covers the necessary variance
  - Incompatible parameter evolution generates conformance problems
- Evaluation
  - Means of abstraction well-understood
  - Instantiation well-understood
  - Good analysis and planning properties
  - May not cover all forms of variance

Service-Oriented

- Summary
  - In large complicated systems often need to provision individual products with different features
  - Not doable with parameters or variation independence
  - Architectural features selectable
- Advantages
  - Instantiation is by selection
  - Possibilities are explicit
  - If done properly, architectural dependencies among services are explicit
  - Analysis at PLA level
  - Planning derived from PLA via selection
  - Evolution via addition OK

Service Oriented -

- Disadvantages
  - Evolution via change/deletion causes conformity problems
  - May not know all the services needed in advance
- Evaluation
  - Simple/effective way of managing product line
  - Likely to be insufficient for complete PLA

Putting It Together

- Comprehensive approach would require all these forms of generic description
- Styles useful for aspects distributed across sets of architectural components
- Under-constrain where flexibility is needed such as changes in technology
- Variations independence for delayed binding
- Parameters where the ranges of solutions are well understood
- Provisioning where the possibilities are enumerable