

Product-Line Architectures

André van der Hoek
Institute for Software Research
University of California, Irvine
andre@ics.uci.edu

Software Architecture (1)

- “*Architecture* is concerned with the selection of architectural elements, their interactions, and the constraints on those elements and their interactions necessary to provide a framework in which to satisfy the requirements and serve as a basis for the design.”

[Perry & Wolf, 1992]

Software Architecture (2)

- “The *architecture of a software system* defines that system in terms of computational components and interactions among those components. ... In addition to specifying the structure and topology of the system, the architecture shows the correspondence between the requirements and elements of the constructed system, thereby providing some rationale for the design decisions.”

[Shaw & Garlan, 1996]

Components

- A component is a building block that is...
 - ...a unit of computation or data store, with an interface specifying the services that it provides
 - ...a unit of deployment
 - ...a unit of reuse

Connectors

- A connector is a building block that enables interaction among components
 - Shared variables
 - Procedure calls (local or remote)
 - Messages and message buses
 - Events
 - Pipes
 - Client/server middleware
- Connectors may be implicit or explicit

Configurations

- A configuration is...
 - ...the overall structure of a software architecture
 - ...the topological arrangement of components and connectors
 - ...a framework for checking compatibility between interfaces, communication protocols, semantics, ...
- Usually constructed according to an architectural style

Four Uses of Software Architecture

- Individual software system
 - Original intent
- Domain-specific software architecture
 - "Good solution"
- Product-line architecture
 - Organizational
- Standard architecture for the public component market
 - Interoperability

Institute for Software Research

©2001, University of California, Irvine

Product-Line Architecture

- "The common architecture for a set of related products or systems developed by an organization." [Bosch, 2000]
- Organizational asset
 - Architecture
 - Components
 - System

Institute for Software Research

©2001, University of California, Irvine

Product-Line Architecture Objectives

- Use a product-line architecture to...
 - ...cut development cost
 - ...increase software quality
 - ...reduce time-to-market
 - ...reduce maintenance cost
- via...
 - ...careful design of the architecture
 - ...reuse of the components
 - ...reuse of the system

Institute for Software Research

©2001, University of California, Irvine

Product-Line Architecture Design

- Design a "configurable configuration"
 - Stable core
 - Base functionality
 - Options
 - Extra features
 - Extra qualities
 - Variants
 - Alternative features
 - Alternative qualities
- A particular architecture instance is selected from the product-line architecture
- View from the "inside-out"

Institute for Software Research

©2001, University of California, Irvine

Difficulties

- Proper mapping of features onto components
- Proper mapping of qualities onto options and alternatives
- Orthogonality of issues
- Variant explosion
- Coordinate variability

Institute for Software Research

©2001, University of California, Irvine

Product/Feature Selection

Product / Feature	F1	F2	F3	F4	...
S1	X				
S2		X	X		
S3	X	X		X	
S4	X		X		
...					

Institute for Software Research

©2001, University of California, Irvine

A Complication

- Architectures evolve
 - New requirements
 - New customers
 - New ...
- Need to *design for* change
 - Flexibility
 - Generality
 - Incrementality
- Need to *capture* change
 - Versions

Institute for Software Research

©2001, University of California, Irvine

Evolving Product-Line Assets

- New product line
- Introduction of new product
- Adding new features
- Extend standards support
- New version of infrastructure
- Improvement of quality attribute

Institute for Software Research

©2001, University of California, Irvine

Component Reuse

- Different kinds of reuse
 - Over subsequent versions of a software product
 - Over product versions and various products
 - Over product versions, various products, and different organizations
- Of note
 - *Do not* develop a single-purpose component
 - *Do* develop a multi-purpose component
 - *Do not* develop a too-many-purpose component

Institute for Software Research

©2001, University of California, Irvine

Product-Line "Life Cycle"

- | | |
|------------------|-----------------|
| ■ Development | ■ Evolution |
| ■ Definition | ■ Architectural |
| ■ Mapping | ■ Component |
| ■ Implementation | ■ System |
| ■ Deployment | |
| ■ Selection | |
| ■ Delivery | |
| ■ Configuration | |
| ■ Instantiation | |

Institute for Software Research

©2001, University of California, Irvine

Establishing a Product-Line Architecture

- Evolve an existing set of products into a product line
- Replace an existing set of products with a product line
- Evolving a new software product line
- Develop a new software product line

Institute for Software Research

©2001, University of California, Irvine

Questions

- Who is responsible for making things generic
- Which product version "pays"
- What is the reward for a developer to make a generic component
 - Takes more time
 - Results in fewer lines of code/hour
- Requires complete and total management focus and commitment

Institute for Software Research

©2001, University of California, Irvine

Important, Related Issues

- Architecture description languages
- Configuration management
- Architectural mismatch
 - Interoperability
 - How can we test all possible combinations?
- Mappings onto system
 - Flexibility
 - Optimization

Institute for Software Research ©2001 University of California, Irvine

Ménage

Institute for Software Research ©2001 University of California, Irvine

Ménage

Institute for Software Research ©2001 University of California, Irvine

