Hope you don’t get sleep during the talk
Product Line Engineering for Web Systems

Dr. Rafael Capilla
Universidad Rey Juan Carlos
http://www.urjc.es

Madrid, SPAIN

Contact me at:
(rafael.capilla@urjc.es)
Content

1. Approaches to Product Line Architectures (PLA)

2. Variability Modeling & Management

3. PLA for Web Systems

4. Conclusions
PLA Approaches

3 main axes in Product Families

- **Main assets**
  - Architecture, Components, Systems

- **Views of the organization**
  - Business, Organization, Process, Technology

- **Life-cycle of assets**
  - Development, Deployment, Evolution

R. Capilla, Benevento, Italy, March 2006
PLA Approaches

PL start-up alternatives (motivation)

- Evolution of a set of products to a PF
- Replacement of products by a PF
- Evolution to a new PF
- Development of a new PF
PLA Approaches

PL heavy approaches. STARS

Product Line Engineering for Web Systems

R. Capilla, Benevento, Italy, March 2006
PLA Approaches

PL heavy approaches

Family-Oriented Abstraction, Specification and Translation (FAST)
David M. Weiss & Chi Tau Robert Lai, 1999

- Domain Engineer
- Application Engineer
- Family Definition
- Family Production Facility
- Family Members

R. Capilla, Benevento, Italy, March 2006
PLA Approaches

PL heavy approaches. FAST

Product Line Engineering for Web Systems

R. Capilla, Benevento, Italy, March 2006
PLA Approaches

PL heavy approaches. PuLSE Overview (Fraunhofer institute)

R. Capilla, Benevento, Italy, March 2006
PLA Approaches

PL heavy approaches. SEI PL Practice initiative

Management
Activities of technical and organizational management, without which the PL initiative will collapse

R. Capilla, Benevento, Italy, March 2006
PLA Approaches

The PRAISE Method

Domain Engineering
- Code
- Domain Experience

Domain Analysis
- Domain Design
  - Domain terminology
  - Reference requirements
- Domain Implementation
  - Reference Architecture
  - Reusable Assets
- Feedback/adaptation/reverse architecture

Depósito de Activos
- Requirements
  - Trace
- Components
  - Trace

Application engineering
- New Requirements
- Application Requirements
- System Design
- Coding

R. Capilla, Benevento, Italy, March 2006
PLA Approaches

PL heavy approaches. ITEA EUREKA Projects

ESAPS-CAFÉ-FAMILIES

http://www.extra.research.philips.com/euprojects/cafe/
http://www.esi.es/en/Projects/Families/
http://www.itea-office.org/

R. Capilla, Benevento, Italy, March 2006
PLA Approaches
PLA Approaches

PL Heavy vs Lightweight approaches. GEARS (Krueger)
http://www.bigleveer.com

GEARS

**Feature Declaration**
- Modeling the scope of the variation in the PL

**Product Definitions**
- Modeling of product instances that can be created from the values of the features

**Variation Points**
- Encapsulation of the variants at code and design levels

R. Capilla, Benevento, Italy, March 2006
PLA Approaches

PL Lightweignt approach. GEARS (Krueger)

FIGURE 5. Extractive Model of Software Mass Customization
PLA Approaches

PL Lightweigt approach. GEARS (Krueger)

3.2 Reactive

R. Capilla, Benevento, Italy, March 2006
PLA Approaches

PL Heavy approach. GEARS (Krueger)

3.3 Proactive

FIGURE 7. Proactive Model of Software Mass Customization
Variability Modeling & Management

Nokia Product Line

8855  6510  8910  5210  8270  8390

8265  3350  7650  6310  6360

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Philips Medical Systems

Architectures

Product Family Architectures

X-Ray Acquisition Platform

EasyVision Workstation Platform

Medical Imaging Platform

Architecture & Software Components

URF Surgery CV MR ATL RAD CT EVM RIS PACS

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Automotive Systems

- Vehicle Navigation Systems (# variants >> 10)
- Driver Assistance Systems (# variants >> 100)
- Engine Control Systems (# variants >> 1000)

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Other Experiences

- Celsius Tech (Naval Electronic Systems)
  - 30 groups of functions with 20 functions/group
  - Range of 3000 to 5000 parameters / component
  - More than 100 sistems y 25 different conf.

- Market Maker (Financial Systems)

- Cummins (Diesel Engines)
  - 9 engine types
  - From 4 to 18 cylinders
  - 5 processors
  - 10 fuel engine types
  - 12 electronic control types

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Concept of Variability

Common aspects
- Common characteristics part of a Product Family
- A template for building software systems
- Standard parts belonging to the same system family

Variability
- Differences between software systems
- Customizable parts of systems
- Diferentes types of variations

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Definitions

Variation Point: An area affected by variability

Variant: A customizable option of a variation point

Variability mechanism: Method or technique by which we represent the variability

Variability analysis: The process by which the variable parts are identified

Binding Time: The moment in which the variability is realized

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Vehicle PL Feature Model

Car Product Line

Car Model A
- Common Features
- Features Model A

Car Model B
- Common Features
- Features Model B

Versions for Car A and B

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

FODA (Featured-Oriented Domain Analysis) Kyo C. Kang, 1990

Car

Mandatory Features

Transmission

Automatic

Manual

Horsepower

Alternative Features

Composition Rule
Air conditioning requires Horsepower > 100

Rationale
Manual more fuel efficient

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Extensions to FODA. Quantitative values (Capilla, Dueñas PFE-2001)

Values: Max / Min, Range, NTimes

Action = <specific action>
if <feature>

{GT | LT | EQ | GE | LE | OutRng | InRng | Ntimes} value):

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Extentions to FODA. Feature dependencies (Lee & Kang, 2004)

**Operational dependency**: Implicit or explicit created relationships between features during the operation of the system => Impact in the development of the assets of the PL

- **Usage dependency**: A feature may depend on other features for its correct functioning or implementation.
- **Modification dependency**: The behavior of a feature may be modified by other feature, while it is in activation.
- **Exclusive-Activation dependency**: Some features must not be active at the same time.
- **Subordinate-Activation dependency**: There may be a feature that can be active only while other feature is active.
- **Concurrent-Activation dependency**

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

UML Modeling
OCL constraints

Tagged Values

Class foo
Att1 : Integer = 2
<<vp2>> Att2 : String

Stereotypes

Meta-models

Proprietary (Several authors)
- FODA, FORM
- Feature-RSEB
- Bosch
- Generative Programming

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Binding Time (C. Fritsch et al., 2002)

- **Programming**
  - Core Asset Development
  - Product Development

- **Integration**
  - Compilation
  - Source Selection
  - Image Build

- **Assembly**
  - Configured at the End Of Line

- **Run Time**
  - First o Every Startup
  - Dynamic

R. Capilla, Benevento, Italy, March 2006
## Variability Modeling & Management

### Variability Implementation Lifecycle

<table>
<thead>
<tr>
<th>LifeCycle Products</th>
<th>Decision Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Func. Req XX: The bit rate for Video Codec Y must be greater than 10</td>
<td>Bit Rate Video Codec Y &gt; 10</td>
</tr>
<tr>
<td>&lt;&lt;variation point 1&gt;&gt; Bit Rate</td>
<td>Variation Point UML class</td>
</tr>
<tr>
<td>&lt;&lt;variant&gt;&gt; BTVideoCodec : Integer = 10</td>
<td>Implementation Level</td>
</tr>
</tbody>
</table>

```c
#ifdef BTVIDEOCODE GT 10
#endif
```

---

**R. Capilla, Benevento, Italy, March 2006**
Variability Modeling & Management

Variability implementation phases

Identification  WHEN and WHERE variability is needed

Constraining  Definition of constraints for each VP (e.g.: variants, dependencies, binding time)

Representation  WHICH Modeling technique

Implementation  Implementation technique

Management

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Variability of a proxy architecture for real-time video in mobile devices

R. Capilla, Benevento, Italy, March 2006
Variation Point 1: related to the interface VideoCodec.
Depending on the encoding level: picture size, image quality, bit rate and so on, a specific video codec will be chosen. In this case the algebraic expression is:

$$VP1 = (\text{MPEG-4-Level-0}) \ XOR \ (\text{MPEG-4-Level-1}) \ XOR \ (\text{MPEG-4-Level-2}) \ XOR \ (\text{MPEG-4-Level-3}) \ XOR \ (\text{H.263})$$

Variation Point 2: related to the interface AudioCodec, with similar functionality to the VideoCodec. Depending on the coding algorithm or bit rate used, the audio codec will be chosen. The algebraic expression for this variation point is:

$$VP2 = (\text{GSM 1}) \ XOR \ (\text{GSM 2}) \ XOR \ (\text{AMR})$$
Variability Modeling & Management

Proxy RT video product derivation

2 VP ->
15 Product Configurations

Variants chosen: MPEG-4 & GSM 1

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

COVAMOF: ConIPF Variability Modeling Framework (Bosch et al.)

CVV: COVAMOF Variability View: Variability view of the PF. Variation points constitute a view of the variability in the PF.

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

COVAMOF: ConIPF Variability Modeling Framework (Bosch et al.)

COVAMOF variability realization

Variation Point \{Variants, Rationale, Realization\}

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Variability Management

- Constitutes a big problem when the number of VP becomes unmanageable, specially in industrial settings
- Each VP is characterized by one or more variants and each variant can hold different values
- Dependencies between VPs introduce a level of complexity
- The lack of tools is a barrier

R. Capilla, Benevento, Italy, March 2006
Product Line Engineering for Web Systems

PLA for Web Systems

Motivation

Market Requirements

Functionality

Tactic Development
Guided by Applications

Strategic Development
Guided by the Domain

Application Development

Family Development

Total Development

Market Pressure

Time

$t_1$ $t_2$ $t_3$

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Challenges

Users demand quick development of Web Products

Different technologies involved (e.g.: HTML, XML, PHP, JSP, WS, Portlets, Servlets)

Frequent changes in products

Short time to market

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Current Needs

- Facilitate maintenance and evolution processes
- Flexible designs for supporting the variability of products
- Quick reaction for evolving products
- Decrease time-to-market of products

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Goals

- Development of flexible and scalable software architectures
- Incorporate variability mechanisms
- Provide Product Family support
- Product derivation & product configuration facilities
- Define flexible production plans with reusable assets

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Strategies we can adopt

- Complete PLA approach
  Domain Engineering

- Lightweight PLA approach
  Code analysis tools
  Reverse Engineering

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Current research issues in the URJC

• **PL process:**
  - Lightweight PL approaches
  - Adaptation to include Web Services

• **Architecting:**
  - Scalable software architectures
  - Adaptation to WS development
  - Variability implementation issues

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Lightweight Product Line Approach

Semi-automatic analysis

Variability & Commonality analysis

Reusable assets

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Asset comparison with VisualDiff Tool
PLA for Web Systems

Factorization process

Identification of potential common & variable code parts

<table>
<thead>
<tr>
<th>Pairs of compared assets</th>
<th>Matched Lines for Asset 1 - Asset 2 / Total lines of Asset 1</th>
<th>Matched Lines for Asset 1 - Asset 2 / Total lines of Asset 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actua-Aula</td>
<td>461 / 509</td>
<td>461 / 536</td>
</tr>
<tr>
<td>Actua-Bibliografia</td>
<td>449 / 509</td>
<td>449 / 833</td>
</tr>
<tr>
<td>Actua-Consejo</td>
<td>65 / 509</td>
<td>65 / 168</td>
</tr>
<tr>
<td>Actua-I+D</td>
<td>445 / 509</td>
<td>445 / 493</td>
</tr>
<tr>
<td>Actua-OtroAire</td>
<td>429 / 509</td>
<td>429 / 494</td>
</tr>
<tr>
<td>Actua-Principal</td>
<td>164 / 509</td>
<td>164 / 351</td>
</tr>
<tr>
<td>Actua-QuienEs</td>
<td>463 / 509</td>
<td>463 / 658</td>
</tr>
<tr>
<td>Actua-Suscribete</td>
<td>446 / 509</td>
<td>446 / 579</td>
</tr>
<tr>
<td>Actua-Transferencia</td>
<td>442 / 509</td>
<td>442 / 501</td>
</tr>
<tr>
<td>Actua-Tribuna</td>
<td>464 / 509</td>
<td>464 / 588</td>
</tr>
</tbody>
</table>

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Variability Analysis Tool (A PHP Parser)
PLA for Web Systems

Variability in Web Assets

- HTML BODY
- <TABLE>
- <FORM>
- Menu
- JavaScript
- Detect Resolution
- Check Browser
- COLOR
- Font
- IE
- Netscape
- COLS
- BORDER
- ALIGN
- ROWS
- COLOR
- SPACING

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Variants in the “DataBaseLayer” component

- **Database component**
- **Database type**
- **Database access**
- **Database operation**
- Ms-Access
- My-SQL

- **SELECT string FROM table**
- odbc_exec($connection, $query)
- mysql_query(($query, $connection))
- odbc_fetch_row ($result)
- mysql_fetch_row($result)
- mysql_result($result, row, field)
- odbc_result($result, field)
- odbc_connect (dsn, user, passwd)
- mysql_connect(host, user, passwd)

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems


Built with the LW-PL

Only a subset was analyzed

http://www.biodiversidad.urjc.es

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Some results from the Environmental Observatory Product

**Hours/man**

- **EO without PL**: 240 (EO)
- **EO with Informal reuse**: 150 (EO), 120 (EO)
- **EO using LW-PL**: 60 (EO), 120 (EO), 56 (EO)

**Assets from E-Learning**

- 120 (EO)

**Component development**

- PL Setup (2 men)

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

Issues not supported in the Web PL

- Support for single & composite Web Services
- Alternatives for binding WS
- Service discovery & selection methods
- Variability information suitable for WS development
PLA for Web Systems

Goals for moving to a service-based PL

- Support for web service composition
- Composition issues defined initially at design and implementation phases
- Include a service-oriented engineering approach inside the PL process
- Adequate variability information support as a natural evolution from the variation points described in the web PL
PLA for Web Systems

LW-PL process adapted for Web Services

- Domain Engineering
  - Product Line Identification
  - Business Analysis

- Application Engineering
  - WS composition from single services
  - Define binding time for WS invocation

- Variation points for WS
  - Aggregated classes for composite WS

- Architecture Construction
- Architecture Customization
- Single / Composite WS
- Product Assembly & Test

WS Composition Process
1. Customize architecture for WS composition
2. Select services to compose
3. Pre eval of composite services

Web application assembly with single and/or composite services

New
“a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task”. (Dey & Abowd, 1999)

Web services can use context information for:

- Discover other services and resources in the network
- Incorporate user’s preferences for modifying the current context
- Plan alternatives depending on the conditions of the context
Context-aware systems

Web services applications can employ context information, user preferences and/or quality attributes for matching similar services from different providers which are distributed across the network.

Figure 1: Interaction with context-aware systems
PLA for Web Systems

Context-aware architectures for WS
A scalable SOA approach

Each CM module manages the input to the system, which in some cases is related to physical sensors or particular devices.

Figure 2: Reference architecture for context-aware systems

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

A basic Context-aware architecture for WS

**Customizer**: Initializes the context and user preferences

**Matcher**: Checks for changes in the current context

**Policies**: For selecting new contexts

Changes in the execution of the system when context information is modified in the client-side, represents a reactive approach. In a pro-active approach we should be able to change the context by discovering new services during run-time (server side).

R. Capilla, Benevento, Italy, March 2006
Varibility description for service composition

- **Orchestration model**: Defines alternatives for the sequence of activities that happen during the execution of the service.

- **Service selection**: Different binding alternatives can be defined with specific variation points for selecting between different services.

- **Exception handling**: Different exceptions can be treated during the execution of a service. Several exceptions can be managed with specific variants.

- **Quality factors**: A list of predefined quality factors can be used to select the best available service during the execution of the system. A variation point can be defined to support this feature.
Designing composite services

Customer
- Invoker : String = Customer
- Location : String = http://www.seekloans.com
- Provider : String = FinancialProvider
- WebMethod : String = FinancialOffers()
- Parameters : String = cheap, years, amount
- MessageNotification : String

FinancialProvider
- DataTypes : String
- TypeOfException : String
- ExceptionMessage : String
- OrchestrationType
- TypeOfSelection : String = Dynamic by lookup
- PortType : String
+ ExceptionHandling() : String
+ OrchestrationModel()
+ ServiceSelection()

FinancialCompany
- PartnerName : String = FinancialCompanyZZZ
- PartnerRole : String = FinancialProvider
- PartnerLink : String = Provider
- Location : String = local
+ FinancialOffers()

IndependentProvider
- PartnerName : String = IndependentFinancialXXX
- PartnerRole : String = IndependentFinancialProvider
- PartnerLink : String = IndependentProvider
- Location : String = http://www.independentfinancial.com
+ TestCustomerProfile()
+ SpecialOffers()

Bank
- PartnerName : String = BankX
- PartnerRole : String = Bank
- PartnerLink : String = Bank
- Location : String = http://www.bankX.com
+ OfferApproval()
+ DefinitiveOffers()
PLA for Web Systems

Dynamic reconfiguration of services

The automatic reconfiguration of the context adding or removing components (services), constitutes a big challenge for context-aware systems and in particular for service-oriented systems.

A variable can be stored and checked periodically to know if the context has changed or if a new service has been selected.

A \texttt{restart()} function in the CM-SOS initiates a new service with the new URL.

In the case of composite services we need to identify which sub-service will be replaced by the new one.
PLA for Web Systems

Dynamic reconfiguration of services

Can BPEL4WS/1.1 manage the service reconfiguration?

- The BPEL <variables> section to know if conditions have changed
- Check conditions and <invoke> the new service
- Endpoint references to assign partner services dynamically

Moving towards WSBPEL 2.0

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

BPEL & WSDL

BPEL 1.1

WSDL 1.1

BPEL 2.0

WSDL 1.1

WSDL 1.2

Abstract Service

Orchestration Language

R. Capilla, Benevento, Italy, March 2006
Variability Modeling & Management

Evolution and Extensibility of Variantion Points

• Adding variants and values during execution
• Adding variation points during execution

R. Capilla, Benevento, Italy, March 2006
PLA for Web Systems

LW-PLA Lessons Learned

Quick set-up in LW-PLA
Decrease time-to-market of products
Early results
Less complex tasks
Service-oriented apps. introduce complexity

R. Capilla, Benevento, Italy, March 2006
Key aspects to consider

- Document core assets
- Detailed production plans
- Automate customization and derivation processes of products (tool support)
- Dynamic reconfiguration of services

R. Capilla, Benevento, Italy, March 2006
That's all... Thanks

R. Capilla, Benevento, Italy, March 2006