Runtime Variability in Domain Engineering for Post-Deployment of User-Centric Software Functional Completion

SEDES
April 2004

Alexandre Manuel Tavares Bragança
alexandre.braganca@i2s.pt
Departamento de I&D
I2S Informática – Sistemas e Serviços, S.A.
Trv. Ribeiro de Sousa, 40, 4250-409 Porto

alex@dei.isep.ipp.pt
Departamento de Engenharia Informática
Instituto Superior de Engenharia do Porto / Instituto Politécnico do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Thesis supervisor:
Ricardo Jorge Silvério de Magalhães Machado
rmac@dsi.uminho.pt
Departamento de Sistemas de Informação / Escola de Engenharia / Universidade do Minho
Campus de Azurem, 4800-058 Guimarães
Contents

• Introduction

• Context and Problem

• Related Work

• Proposed Approach

• Future Work and Conclusions
Introduction

• Background research field is domain engineering

• Researcher is integrated in enterprise organization.
  • Action research [Baskerville 2001].
  • I2S Informática – Sistemas e Serviços, S.A.
There is almost no reuse. Each time we need to start from fresh...
Commonalities enable reuse.

Variability enable differentiation of applications in a domain.
It is then necessary to identify, represent and implement variability.

Variability in software is achieved fundamentally by pre-deployment techniques. [Svahnberg and Bosch 2000]
Context: Variability Implementation

- Conditional execution of code (e.g. if ... then ...);
- Inheritance;
- Parameterization (e.g. C++ templates);
- Runtime loading of variation implementation (e.g. ODBC driver);
- Component connection based on interfaces (e.g. COM and XPCOM);
- Reflection in programming languages (e.g. Java).

[ Gurp 2003]
Some **vertical domains** are very dynamic, e.g., the **insurance domain**.

For instance in insurance companies:
- the need to market new products is high and time-to-market is normally very low;
- insurance products are very complex; each new product has new software requirements.

10 Companies * 20 Products = 200 Variations!
Related Work: Variability Representation

**Feature diagram** is a major tool used to represent variability [Kang et al. 1998; Griss et al. 1998; Deursen et al. 2002; Gurp 2003].

“A feature is a logical unit of behavior that is specified by a set of **functional and quality requirements**” [Bosch 2000].
Related Work: Variability Representation

Example of feature diagram [Kang et al. 1990]

---

Composition Rule:
Air Conditioning requires Horsepower > 100

Rationale:
Manual more fuel efficient
Related Work: Variability Implementation

One of the most interesting variability technique is **layer component composition**. One example of such technique is GenVoca [Batory and O’Malley 1992].

A component can be easily expressed, e.g.,

```
bag[concurrent[sizeof[unbounded[managed[heap]]]]]
```

“A feature maps well to a component layer”
Related Work: Domain-Specific Languages

A domain-specific language regards a programming language with syntax and semantics specialized for a particular application domain or type of problem [Hudak 1998].

- DJ (horizontal) [Lopes 1997]
- RISLA (vertical) [Deursen et al. 1996]
Proposed Approach

Thesis statement

The adoption of domain-specific languages (with runtime framework) in domain software engineering can significantly improve product line variability and flexibility by
(a) enabling an effective technique for implementing runtime variability;
(b) enabling the possibility of a post-deployment kind of software engineering; and
(c) enabling variability in a technically heterogeneous product line.
Proposed Approach

Domain Reference Architectures

Application Architecture

Component Composition

Repository of Component Implementations

(1) Selecting existing components to implement a feature
(2) Specifying/programming component implementations

Application Domain Specialist: Domain expert playing the role of post-deployment application specialist
Proposed Approach

Our approach to the runtime variability aspects:

(1) how to combine features at runtime, i.e., how do Application Domain Specialists select features at runtime (static runtime variability).
   - Our approach will be based on pre-deployment techniques like GenVoca component composition applied at runtime.

(2) how to enable behavior specification holes in the applications for the Application Domain Specialists to fill at runtime (dynamic runtime variability).
   - Our approach will be based on runtime component implementation based on domain-specific languages.
Future Work and Conclusions

Runtime Variability in Domain Engineering for Post-Deployment of User-Centric Software Functional Completion
Expected contributions to the research field:

• **Methodology contribution**;

• **Notation contribution**;

• **Technology contribution**.
Runtime Variability in Domain Engineering for Post-Deployment of User-Centric Software Functional Completion

SEDES
April 2004

Alexandre Manuel Tavares Bragança
alexandre.braganca@i2s.pt
Departamento de I&D
I2S Informática – Sistemas e Serviços, S.A.
Trv. Ribeiro de Sousa, 40, 4250-409 Porto

Further information regarding this work can be found at http://www.dei.isep.ipp.pt/~alex
References


