Dealing with Unstable Domains in Product-Line Architecture Development

9th International Conference on Information Systems Implementation and Modeling — ISIM 2006
Přerov, Czech Republic

Valentino Vranić and Vladimír Marko

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technology
Slovak University of Technology in Bratislava

April 25, 2006
Product Lines and Stability

Product lines: an organized approach to software reuse

Domain is an area of knowledge
  - Scoped to needs of its stakeholders
  - Includes a set of concepts and terminology of the respective area
  - Includes knowledge how to build software systems in that area

A domain is expected to be stable and well-understood

How to benefit from product lines in an unstable domain?
Overview

1. Approach Overview
2. Specific Product Analysis
3. Domain Generalization
4. Structure View
5. Conclusions
Dealing with Unstable Domains in Product-Line Architecture Development

Approach Overview

Coarse Feature Model

Refined Feature Model

Partial Use Case Model

Generalized Use Case Model

Components

Subsystems

Partial Domain Model

General Domain Model

Archetypal Entities

Structure

Components
Feature Model

- Concepts expressed by their features
- A feature is an important property of a concept
- Common and variable features
- Focus on configurability
An example: domain of knowledge management

Encompasses applications for acquisition, organization, and maintenance of knowledge in the web
Use Case Model

- Captures stakeholders and functional requirements
- Abstracts from realization details
- Variations in functional requirements are captured by the use cases to features mapping
- No need to use specialized use cases or variants
A Domain Exploration Level Use Case Diagram

Offer job

- Find job offer
  - Job seeker
- Acquire job offer
  - Job tracker
- Check validity

Employer

Active employer

Publish job offer in own information space

Submit job offer
Domain Generalization

- Based on the partial domain model
- Objectives:
  - Identify archetypal entities of the domain and interactions between them
  - Refine feature model
  - Generalize use cases
Archetypal Entities

- A major transition in a model
- Based on the abstract feature model and concrete use cases
- Knowledge acquisition as an example
  - Knowledge acquisition identified as crucial in the domain
  - The use cases are about job offer acquisition
  - Therefore, the domain has been narrowed to offer acquisition
  - The archetypal entities identified: an abstract offer, its producer, and its consumer
Separation of the information content independent acquisition from the dependent one based on the archetypal entities
Generalizing Use Cases (1)

- Based on the archetypal entities identified
- Evolved according to the refined feature model
- The objective is to achieve a use case model that can be mapped to the structural view
  - A use case as a collaboration of several actors
  - Some actors represent subsystems
Generalizing Use Cases (2)
Dealing with Variations in Requirements (1)

- Representing variations of requirements as separate use cases should be avoided.
- Mapping to appropriate features bears this information.
- It is necessary to have separate use cases for variations only if they involve different actors.
Dealing with Variations in Requirements (2)
Let behavior form the structure: the system structure is derived from the use case model
Avoid structuring the system according to the developing organization structure (Conway’s law)
Two levels of structural decomposition
- Subsystem view: logical cohesion
- Component view: functional cohesion
Dealing with Unstable Domains in Product-Line Architecture Development

Structure View

Subsystem View

Application layer

- «subsystem» Offer entry portal

Domain-specific services layer

- «subsystem» Offer database
- «subsystem» Offer acquisition
- «subsystem» Offer maintenance

Domain-independent services layer

- «subsystem» Data acquisition
- «subsystem» Data maintenance

Domain dependence
Conclusions

- An approach that enables exploiting the benefits of product lines in unstable domains
- Presented on examples from a project on whose development part is performed concurrently with the research activities
- Improved understanding of a specific part of a domain can be translated to the whole domain
- Identification of archetypal entities and their interactions highly dependent on the insight of developers
- However, the partial domain model improves the communication with domain stakeholders—inevitable for the generalization