Exploring Multi-Level Modeling Relations Using Variability Mechanisms

Iris Reinhartz-Berger, <u>Arnon Sturm</u>, Tony Clark







Software Product Lines, Multi Level Modeling

Outline

1. Introduction

- Software Product Lines
- 2. Current Multi-Level Modeling Approaches
- 3. Aligning Variability Mechanisms and Type-

Instance Relationships

4. Conclusions and Future Work

1. INTRODUCTION

Multi Level Modeling (MLM)

4



Software Product Line Engineering (SPLE)

□ A software product line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of domain assets in a prescribed way. [SEI at Carnegie Mellon web site]

Examples:

Cars, Mobile phones, Control systems ...

□ SPLE can be considered as dealing with multi-levels.

Software Product Line Engineering (SPLE)



Software Product Line Engineering (SPLE)

Domain Engineering

- Domain engineering is the development and evolution of domain specific knowledge and artifacts to support the development and evolution of systems in the domain.
 - The purpose of domain engineering is to identify, model, construct, catalog, and disseminate the commonalities and differences of particular domain applications
 - Domain engineering includes engineering of domain assets, i.e., models, components, methods and tools
 - These assets can later be reused in application engineering activities.

Software Product Line Engineering (SPLE) Application Engineering

- Application engineering deals with developing specific software products or applications utilizing domain assets.
 - It basically takes all common requirements and similarities from the domain/product line and the specific requirements of the software products or applications to be developed.
 - Using the base from the domain engineering phase and the specific requirements of the product/application a complete and new product/application can be built.

SPLE Variability Mechanisms

Configuration
Parameterization
Template Instantiation

Demonstrating SPLE Variability Mechanisms

10

Check-In-Check-Out domain



SPLE Variability Mechanisms as "of" relationships



Demonstrating SPLE Variability Mechanisms

12	
Variability Mechanism	Example
Configuration	Elements of the higher level are selected to be included in the lower level. Partial selections are possible, as in the case of UnivBook which selects only 2 out of the 3 attributes of Book: ISBN and Title.
Parameterization	Supports assigning values to parameters defined in a higher level. The assignment is done in a lower level. In our example, the maximal number of copies of a certain book is assigned to 5 in LMS4Univ.
Template Instantiation	in contrast to parameterization that deals with value assignment, deals with type adaptation, is exemplified by constraining book copy check out with enumeration type (which represents user type, e.g., student vs. staff).

Research Motivation

13

□ Lack of formalism in SPLE

- Configuration Choice between alternative functions and implementations [17]; Modifying selected elements of a core asset based on predefined rules that refer to specific requirements or situations [6, 7].
- Parameterization Variation points for features [17]; Data items serving as arguments for distinguished software behavior [5].
- Template Instantiation Type adaptation or selecting alternative pieces of code [17]; Enables filling in product-specific parts in a generic body [5].
- Lack of simplicity in representation type-instance relationship

Objectives and Contribution

- Explore the potential relations between SPLE and MLM.
- □ The contribution of the paper is two folded:
 - To MLM, SPLE variability mechanisms can be used to simplify the representation of complex relationships in current MLM approaches.
 - To SPLE MLM concepts can be used to more precisely define SPLE variability mechanisms.

2. CURRENT MULTI-LEVEL MODELING APPROACHES

Mechanisms for MLM

Clabject

- Has attributes, operations, links, properties, methods and associations.
- Potency
 - Defines the depth to which a model element can be instantiated.

Mechanisms for MLM

17

Orthogonal Classification Architecture (OCA)

- Elements can be instantiated along the linguistic dimension and across the ontological dimension.
- □ Power Type
 - A power type is an object type whose instances are subtypes of another object type.
- Interoperability
 - specialization by extension, specialization by refinement, standard instantiation, instantiation with extension, subtypes of another type, enumeration relation.

3. ALIGNING VARIABILITY MECHANISMS AND TYPE-INSTANCE RELATIONSHIPS

SPLE Variability Mechanisms as "of" relationships



The Kernel Language





Mapping the Kernel Language and Variability Mechanism

21	
Variability Mechanism	MLM Instantiation Constraints and Examples
Configuration	Checks that the structure of the instance is consistent with the variability specified by the type. The LMS is a family with Book as an instance of ConfigurableClass with PublicationYear as an optional attribute. The class UnivBook is a type-consistent instance, and therefore a configuration of Book.
Parameterization	Assigns a value to a type to create an instance of that type. In the case of the LMS, the LMS4Univ assigns the value 5 to the parameter n, appearing in LMS and specifying the maximal number of BookCopy associated to a single Book.
Template Instantiation	Assigns a value to a type to create another type. The class BookCopy is parametric with respect to the parameter Constraints and the binding of Constraints to Enumeration is shown to produce the instantiated class UnivBookCopy.

4. CONCLUSIONS AND FUTURE WORK

Conclusions

MLM approaches have been proposed in order to relax the traditional strictness requirements on inter- and intra-level typeinstance relationships.

these proposals are formal,

- they address the representation of complex relationships to a limited extent
- In this work we address this tradeoff by adopting SPLE variability mechanisms within a type-instance framework.

We plan to further develop the kernel language and test it in the context of SPLE and to use it as the basis of mixing different MLM approaches with SPLE variability mechanisms.

Questions???

24

Iris Reinhartz-Berger, Information Systems, Haifa University, Israel iris@is.hevra.haifa.ac.il

Arnon Sturm, Information Systems Engineering, Ben-Gurion University of the Negev, Israel

sturm@bgu.ac.il

Tony Clark, Computer Science, Sheffield Hallam University, UK <u>t.clark@shu.ac.uk</u>