Specification and Utilization of Core Assets: Feature-Oriented vs. UML-based Methods

FP-UML'
2011

Iris Reinhartz-Berger & Arava Tsoury

Department of Information Systems
University of Haifa, Israel

Agenda

- Software Product Line Engineering
  - Core assets and product artifacts
  - Specification vs. Utilization
- Core Assets Specification Methods
  - Feature-Oriented Methods
  - UML-based Methods
- Comparison of Methods
  - CBFM vs. ADOM
  - Empirical Study: Comprehension, Specification, and Utilization
- Conclusions
- Future Work

Software Product Line Engineering

- ***Software product line engineering*** deals with specifying and developing artifacts that can be utilized and adapted into specific software products in a family (line).
  - ***Core assets*** are reusable parts that are built to be used by more than one product in the family.
  - ***Product artifacts*** are specific parts of the software products.
    - Core assets are used for creating product artifacts.
    - Product artifacts can be enhanced by adding specific elements that are not part of the core assets.

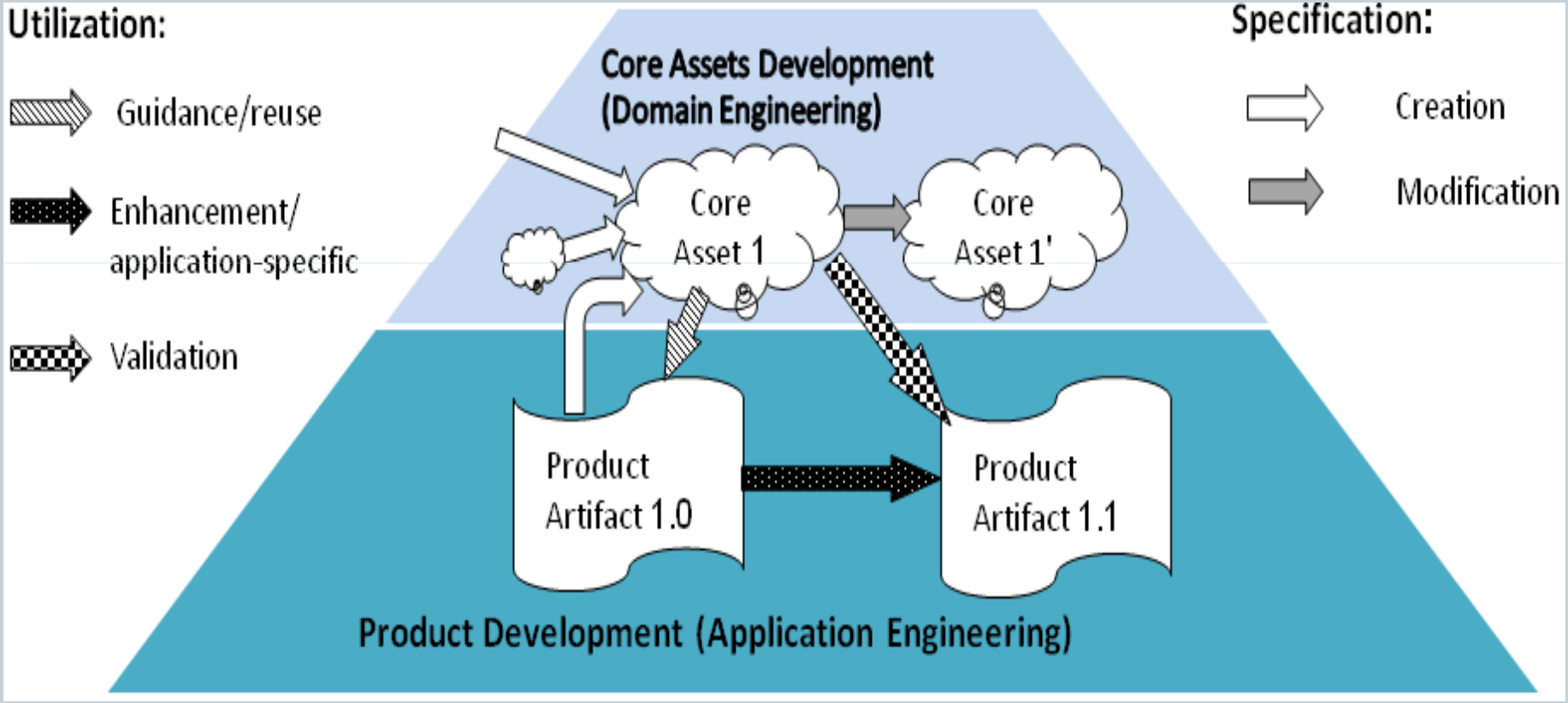
Software Product Line Engineering

- Two main activities related to core assets in SPLE:
  - Specification
  - Utilization
- Specification of core assets includes:
  - **Creation** of core assets can be done by extracting knowledge from particular product artifacts, constructing analogy from other core assets (in parallel domains), or reviewing and studying the domain of interest.
  - **Modification** takes core assets in a certain domain and introduces changes in order to enlarge the domain scope, specify newly discovered constraints, fix inaccuracies or errors, and so on.

Software Product Line Engineering

- Utilization of core assets includes:
  - Core assets are **reused** for guiding the creation of a particular product artifact. This step includes using the common kernel of the domain as captured in the core asset and choosing particular available variants.
  - The product artifact is then **enhanced**, adding application-specific elements in order to satisfy the specific requirements of the product to be developed.
  - The specification of the enhanced product is finally checked and **validated** with respect to the core asset, in order to avoid violation of the domain constraints specified in the core asset or the accompanying production plan.

The Suggested Framework



Core Assets Specification Possibilities

Nature of solution	No. of studies
Feature model	33
Using UML and its extensibility	25
Express variability as part of a technique that models the architecture of the system	8
Using natural language	6
Expressed variability as part of a technique that models the components of the system	5
Formal techniques based on mathematics	4
X-frames organized into a layered hierarchy	4
Domain-specific language	3
Ontology based techniques	3
Solution from the perspective of aspect-orientation	2
Orthogonal variability management	2
Configuration management based modelling	1
Using information visualization techniques	1

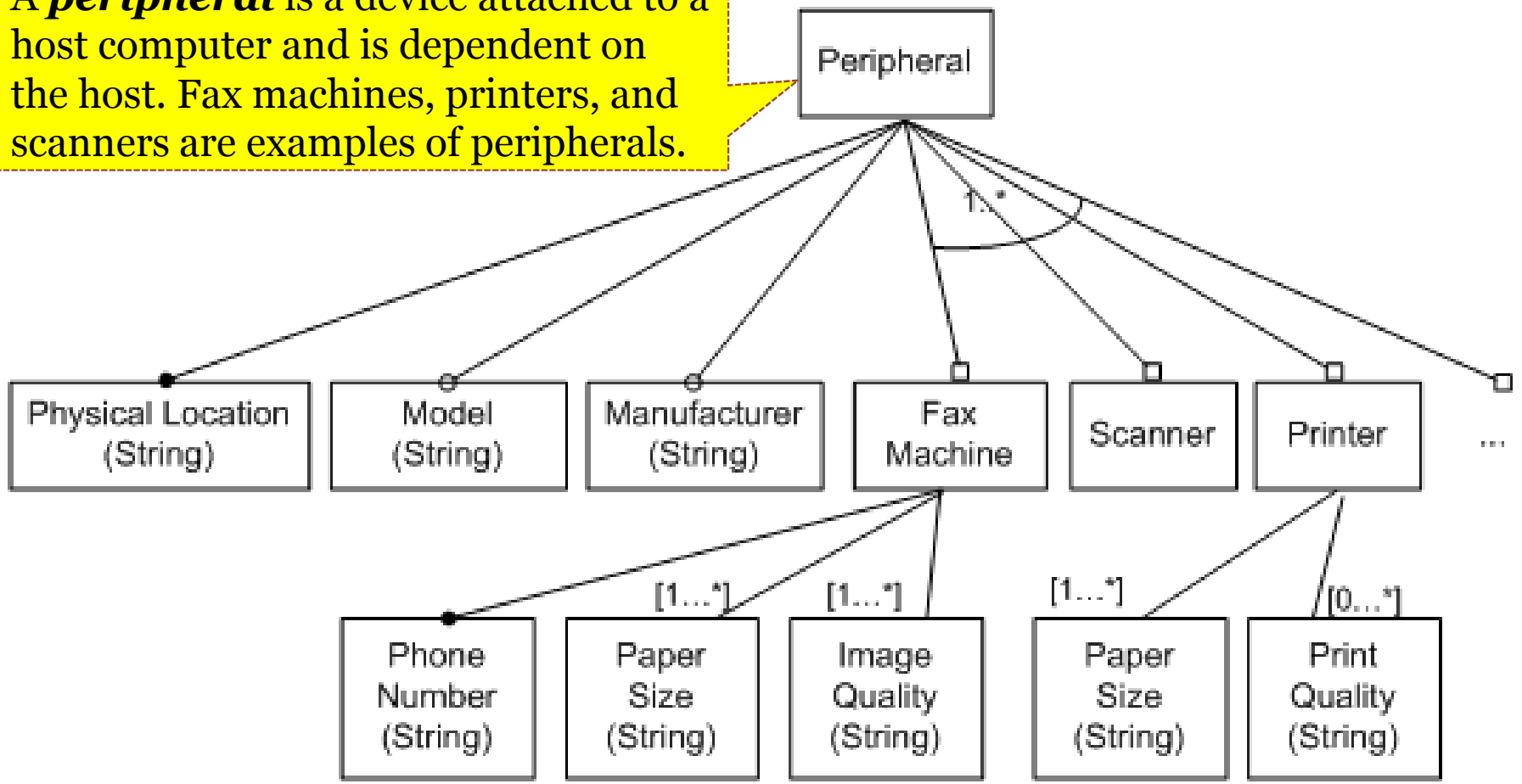
Chen, L. and Babar, M. A. A systematic review of evaluation of variability management approaches in software product lines. Information and Software Technology 53, pp. 344-362, 2011.

Feature-Oriented Methods

- ***Feature-orientation*** supports specifying core assets as sets of characteristics relevant to some stakeholders and the relationships and dependencies among them.
- A product artifact uses the reusable (core) asset and instantiates a sub-set of features that satisfies certain conditions.
- Variability is specified in terms of mandatory vs. optional features, alternatives, OR features, 'require' and 'exclude' dependencies among features, and composition rules.
- The root of most feature-oriented methods is Feature-Oriented Domain Analysis (FODA, 1990)

Feature-Oriented Methods

A ***peripheral*** is a device attached to a host computer and is dependent on the host. Fax machines, printers, and scanners are examples of peripherals.

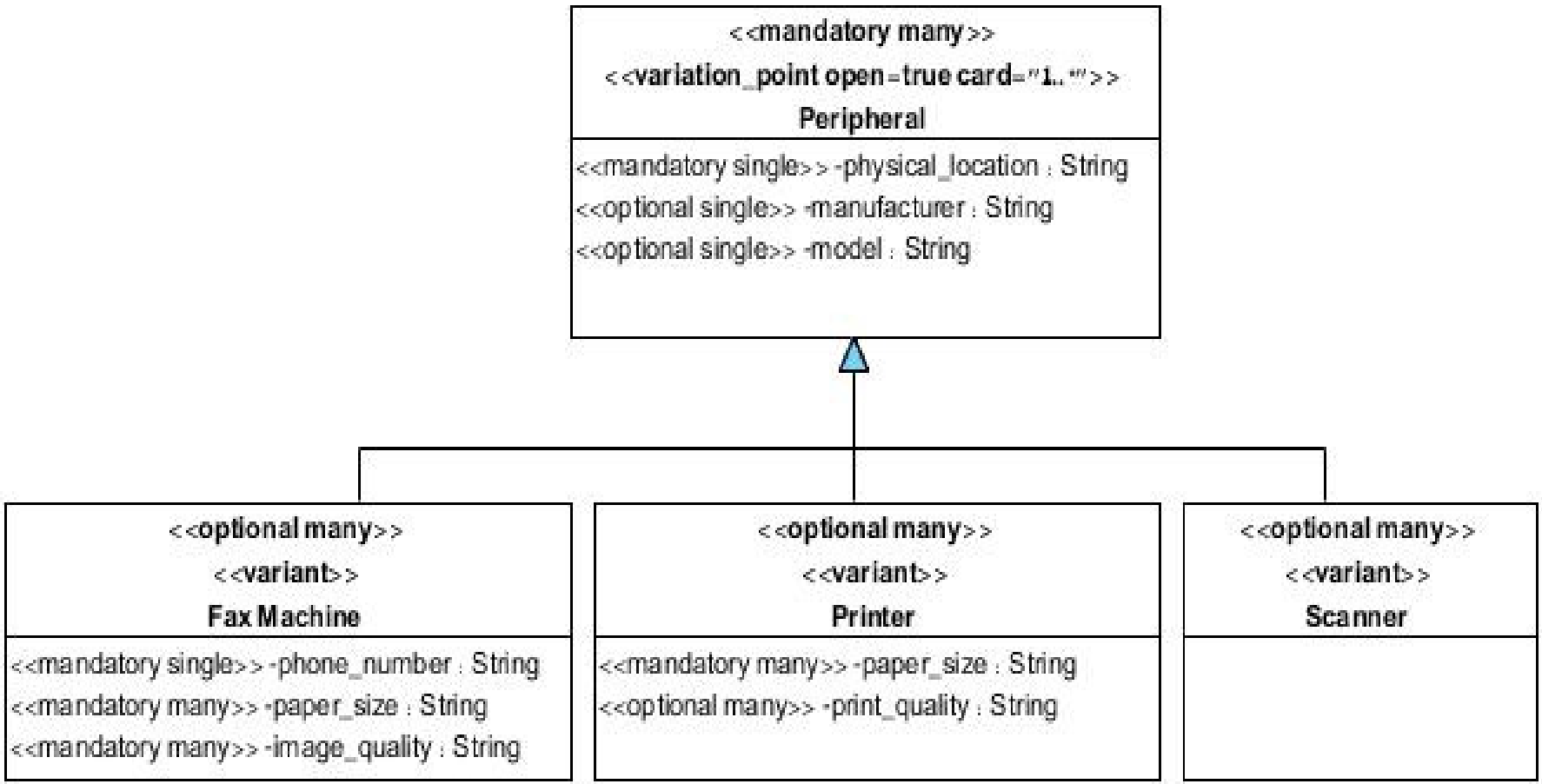


Back

UML-Based Methods

- ***UML-based methods*** extend UML metamodel or more commonly suggest profiles for handling core asset specification and variability modeling.
- They typically represent variation points, variants, and dependencies among them.
  - A ***variation point*** identifies a location in a core asset at which a variation may occur.
    - They further provide guidance and validation aids for adapting core assets to particular contexts (product artifacts).
  - A ***variant*** realizes a possible way to create particular product artifacts.

UML-Based Methods



Back

Comparison of Methods

- While core assets specification methods get a lot of attention in SPLE, their comparison in general and in terms of comprehension, specification, and utilization is somewhat neglected.
- We conducted a pilot evaluation study to fill this lack.
  - The study's results regarding comprehension have been published in EMMSAD'2011.
  - Here we focus on comparison of specification and utilization capabilities of two core assets specification methods (CBFM and ADOM).

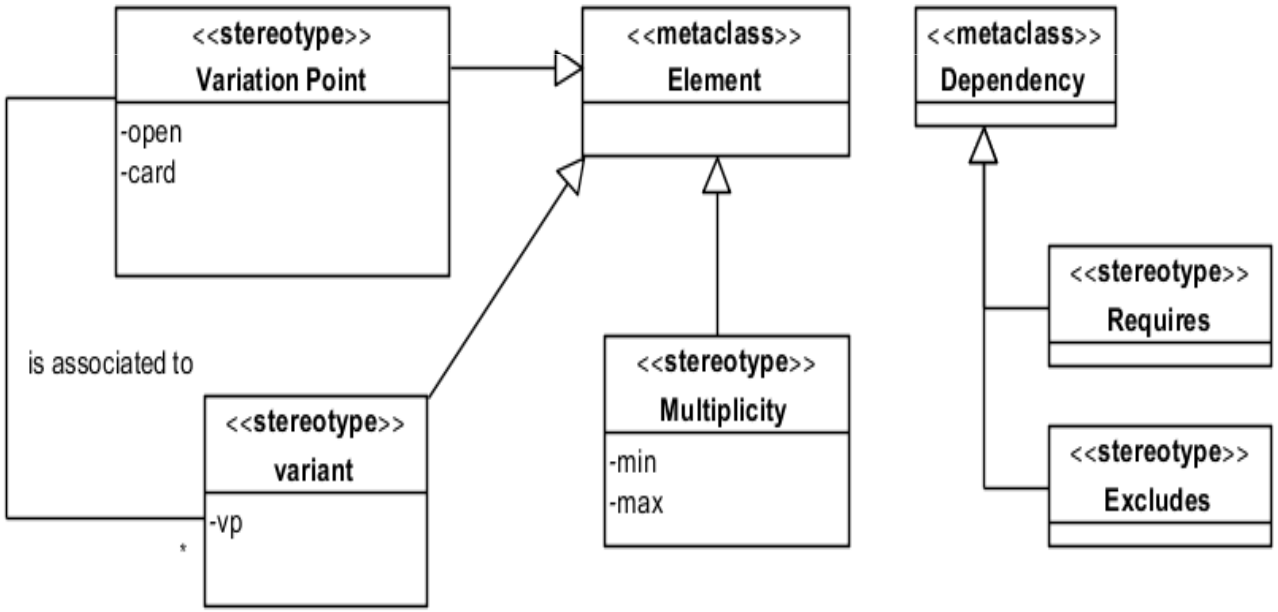
CBFM in Nutshell

- ***Cardinality-Based Feature Modeling*** (Czarnecki & Kim, 2005) adds to the basic feature diagrams (from FODA) the following specification capabilities:
  - Feature cardinality, denoting the range of the feature clones that can be included at certain points
  - Feature groups, enabling organizing features and defining how many group members can be selected at certain points
  - Attribute types, indicating that an attribute value can be specified during configuration
  - Feature model references, enabling splitting large feature models
  - OCL constraints, enabling the description of additional constraints and dependencies

To the example …

ADOM in Nutshell

- ***Application-based Domain Modeling*** (Reinhartz-Berger & Sturm, 2009) is based on a UML profile for handling commonality and variability aspects in core assets specification.
- ADOM's profile:



To the example …

The Selection of the Methods

- The expressiveness of CBFM exceeds that of other feature-oriented methods:
  - It enables using OCL for specifying different inter-dependencies
  - It allows defining various cardinalities for specifying selection rules of variants
- The expressiveness of ADOM exceeds that of other UML-based methods:
  - It explicitly refers to the selection and addition of variants in certain variation points
  - It enables explicit specification of both variation points and variants
  - It allows specifying ranges of multiplicity via the «multiplicity» stereotype

Empirical Study: Research Questions

1. The specifications of which method, CBFM or ADOM, are more **comprehensible**?
2. The specifications of which method are more **modifiable** (i.e., easy to be modified)?
3. The specifications of which method are more **guidable**?
4. The specifications of which method **help enhance** particular product artifacts?
5. The specifications of which method help create **valid** product artifacts?

Empirical Study: Settings

- **Independent Variables:** The modeling methods
- **Controlled Variables:**
  - ***Subjects:*** 18 advanced undergraduate and graduate information systems students in a domain engineering seminar course
  - **Tasks:** Given a Virtual Office (VOF) domain specified in CBFM or ADOM:
    - Comprehension (true/false) questions
    - Modification tasks of the core assets
    - Creation tasks of product artifacts
- **Dependent Variables:** Level of comprehension/ Quality of Specification/Utilization

Empirical Study: Settings

- **Notes:**
  - All the students had previous knowledge in systems modeling and specification, as well as initial experience in industrial projects.
  - The students were pre-assigned into two similarly capable groups.
    - All students had to answer the comprehension questions.
    - Half of the students had to carry out the modification tasks, while the other half had to perform the creation tasks
  - The students were motivated by up to 10 points of the final grade.
  - Post-interviews were conducted.

Comprehension Results
(elaborated in the EMMSAD'2011 paper)

	Criterion	CBFM	ADOM	p-value
Commonality Aspects	Multiplicity	76	71	0.693
	(Inter-) Dependencies	76	74	0.884
	Overall Commonality	76	71	0.692
Variability Aspects	Variability Specification	71	87	0.029
	Selection rules	70	75	0.755
	Addition rules	74	87	0.421
	Overall Variability	72	83	0.299

Specification (modification) Tasks

- The modification tasks mainly required extensions to the core asset and not inventing new parts from scratch.
  - *For checking devices two strategies are available: distance-based and attribute-based. In the distance-based strategy the application locates the nearest available devices for the task in hand … In the attribute-based strategy, the employee needs to supply values to different relevant parameters of the device, such as the paper size and the print quality for printers and the paper size and the image quality for fax machines. Each application in VOF domain must support the distance-based strategy, but may support both strategies.*

Specification (modification) Results

Very Similar Results!

Aspect	CBFM	ADOM
Commonality	0.59	0.57
Variability	0.57	0.56

- However some sources of difficulties to perform this task were identified in the students' interviews:
  - The success in performing modifications was influenced from the model structure
    - The tree-like structure of CBFM models helped perform modification of elements that reside at the same sub-tree
    - cross cutting aspects were easier to be done in ADOM
  - In ADOM some aspects were partially specified using only the expressiveness of UML
    - Neglecting the extra specification of meta-information suggested by the profile
  - The resultant ADOM models were sometimes inconsistent
    - Due to the need to update different types of diagrams for the same task

Utilization Tasks

- The utilization tasks required creating valid portions of a brokers' application in the domain, according to a predefined list of requirements, and listing the parts of the requirements that cannot be satisfied with the given core assets.
  - *While operating a peripheral in the brokers' application, notifications are sent to both employees (via emails) and log files. Furthermore, when performing a task on a device, three main checks are performed: accessibility, feasibility, and profitability.*

Utilization Results

Method	Guidance		Product Enhancement	Product Validation	
	Comm.	Var.		Comm.	Var.
CBFM	0.45	0.64	0.68	0.12	0.5
ADOM	0.69	0.62	0.47	0.25	0

- Analyzing the sources of difficulties, we found:
  - **Product Guidance:**
    - The existence of explicit inheritance relations in ADOM helped understand the hierarchical structure of the domain
    - In CBFM, the subjects experienced difficulties in creating several instances of the same core asset element (due to misinterpretation of features group cardinalities)
  - **Product Enhancement:**
    - Subjects tended to point on requirements as violating the domain constraints rather than application-specific additions in ADOM's models

Utilization Results

Method	Guidance		Product Enhancement	Product Validation	
	Comm.	Var.		Comm.	Var.
CBFM	0.45	0.64	0.68	0.12	0.5
ADOM	0.69	0.62	0.47	0.25	0

- **Product Validation:**
  - In both methods, the main source of difficulties in commonality-related questions was comprehension of dependencies between elements
  - CBFM outperformed ADOM in variability-related issues
    - However, only one task referred to this aspect.
    - The violation in this task referred to a dependency between two variants.
    - We believe that the difference in this case is due to relatively crowd specifications in ADOM with respect to CBFM.

Threats to Validity

- Type of subjects: students
- The small number of the subjects (18 who where divided into groups in the specification and utilization tasks)
- The relatively simple tasks and models
- The specific selected methods
- Only further studies may confirm or disconfirm whether our results can be generalized to more experienced subjects, more complicated models and tasks, and other modeling methods.

Conclusions

- Utilization and specification of core assets are important when dealing with SPLE.
  - In this paper, we analyzed the main difficulties in performing these tasks in two methods from different leading modeling paradigms:
    - CBFM – a feature-oriented method
    - ADOM – based on a UML profile
- We found similar results in specifying, or more accurately modifying, core assets in the two methods, while some difficulties and problems in core assets utilization were encountered in the two modeling paradigms.
- Despite the low number of subjects, we tried to provide some explanations to these problems, analyzing the subjects' outcomes and interviews.

Future Work

- We have already conducted experiments on larger classes of students. The results are quite similar.
- We intend to perform similar studies with professionals in industrial settings.
- We plan to theoretically check the expressiveness of different methods and to empirically compare more methods.

Future Work

Category	Specification Aids		Utilization Aids	
	Commonality	Variability	Guidance	Validation
Feature-oriented	Mandatory and optional elements, dependencies	Feature groups, alternatives, and OR-related features	Cardinality, rationale, constraints	Instantiation and configuration conformance
UML-based	Mandatory and optional elements, dependencies	Variation points, variants	Cardinality, openness, reuse mechanisms, binding time	Specialization and configuration conformance

Legend: Eval 1, Eval 2, Eval 3

Thank you for your attention. Any Questions?

FP-UML'
2011

Iris Reinhartz-Berger iris@is.haifa.ac.il
Arava Tsoury aravabt@gmail.com