Towards Evolving Configuration Models

Thorsten Krebs, Lothar Hotz

Universität Hamburg

Christoph Ranze, Guido Vehring

encoway ebusinessautomation

Outline

- Introduction
  - Application Context
  - Basic Approach
  - General View on Evolution
- Consistency of a Model
- Basic Operations for Modifications
- Degrees of Modifications
- Support for Evolution
- Summary

Application Context

- ➔ Configuration of Industrial Product Families (**ConIPF**):
  - Combining *Product Lines* and *Structure-oriented Configuration*.
- ➔ Software-based products are rarely without relatives.
- ➔ Products evolve to accommodate specialized demands by:
  - Becoming **variants and versions** of the same product, or
  - Becoming **separate but similar products**.
- ➔ This set of products represents a **product family**.

General Approach

➔ Defining a methodology for deriving products starting with features / customer requirements.

- Use **feature and artifact models**,
- Develop **intermediate knowledge representations**, and
- Define a **configuration notation** with the expressiveness to handle the variability in industrial product lines.

Evolution in the Application domain

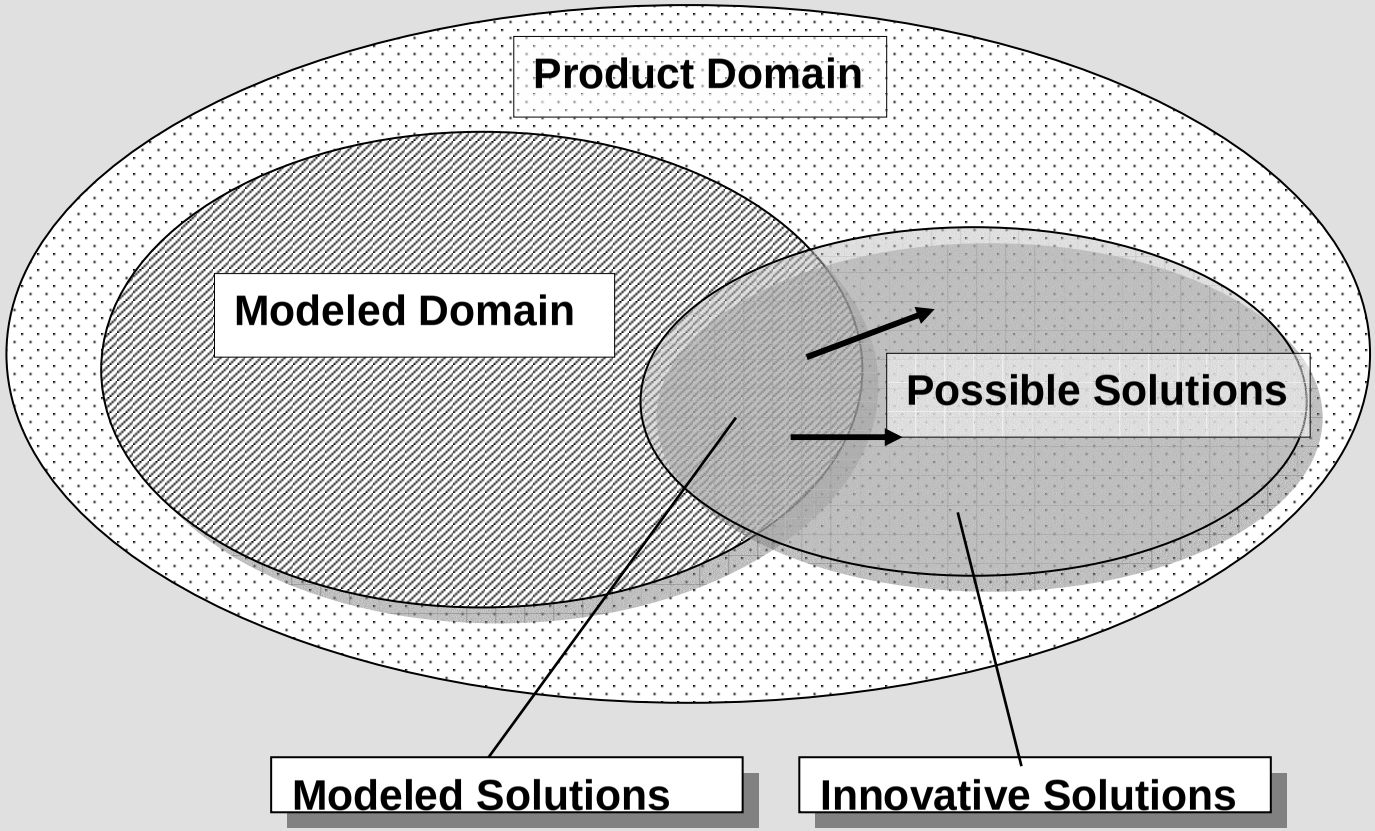
- ➔ Reasons for evolution:
  - New requirements
  - New technical abilities
  - Bug fixes
- ➔ Preventative evolution during domain engineering
- ➔ Adaptive evolution during application engineering
- ➔ Corrective evolution during maintenance

Evolution of the existing artifacts vs. evolution of the models

Structure-based Configuration

- ➔ Configuration Knowledge Modeling Language **(CKML):**
  - Conceptual model:
    - Describing objects by types and properties
    - Relations:
      - Taxonomic (*is-a*),
      - Compositional (*has-parts*), and
      - Constraints
  - Procedural model
- ➔ Goal Specification
  - A priori known facts about the *product-to-be*

General View



Evolution/Innovative Configuration

Aspects of Innovative Configuration

- ➔ Changing the domain model
- ➔ Changing already (partly) configured configurations
- ➔ Changing the real artifacts
- ➔ Evolution and the configuration process:
  - Interchanging approach vs.
  - Separate processes
- ➔ Preventative evolution does not have impacts on the configuration process!
  - „Only“ new real artifacts have to be produced

Change the domain model, don‘t change the configuration goal.

Aspects of Innovative Configuration

- ➔ **Changing the domain model**
- ➔ **Changing already (partly) configured configurations**
- ➔ Changing the real artifacts
- ➔ Evolution and the configuration process:
  - Interchanging approach vs.
  - Separate processes
- ➔ Preventative evolution does not have impacts on the configuration process!
  - „Only“ new real artifacts have to be produced

Change the domain model, don‘t change the configuration goal.

General Questions

- ➔ How can the configuration model be changed?
- ➔ What are pre-requisites?
- ➔ What is allowed to be changed?
- ➔ What has to be true after the change?
- ➔ What is the relation between innovative configuration and knowledge acquisition?

Pre-requisite for reasoning: A consistent model!

Consistency of a Model

- ➔ Specialization conditions:
  - Subsets between each property of a sub- and super-concept
- ➔ Compositional conditions:
  - Common super-concept of the parts
  - Disjunctive parts
  - Correct number restrictions
  - One part can be related to several wholes (non-exclusiveness assumption on parts)
- ➔ Constraint conditions:
  - Constraint must be related to concept properties
  - Constraint may only compute subsets of value ranges

Basic Operations for Modifications

- ➔ Add, delete, modify (= delete plus add)
- ➔ Extending/reducing parameter value ranges
- ➔ Adding/deleting parameters
- ➔ Adding/deleting specializations
- ➔ Adding/deleting decompositions
- ➔ Adding/deleting constraints

- ➔ Changing procedural knowledge has no impact on the result!

Modification Impacts

Operation	Impacts on the Model	Impacts on old Configurations
Extending parameter value ranges	Check the subset relation and corresponding constraints	Existing configurations are still consistent
Deleting parameter values	Check the specialization sub-tree and the corresponding constraints	May be inconsistent
Adding new parameters to a concept	Perform inheritance	May miss the new parameter
Deleting parameters	Check the specialization sub-tree and the corresponding constraints	May be inconsistent
Adding new specializations	Perform inheritance, check concept types	Instances may be of different type
Deleting specializations	Modify/delete the sub-concepts	May be inconsistent

Modification Impacts

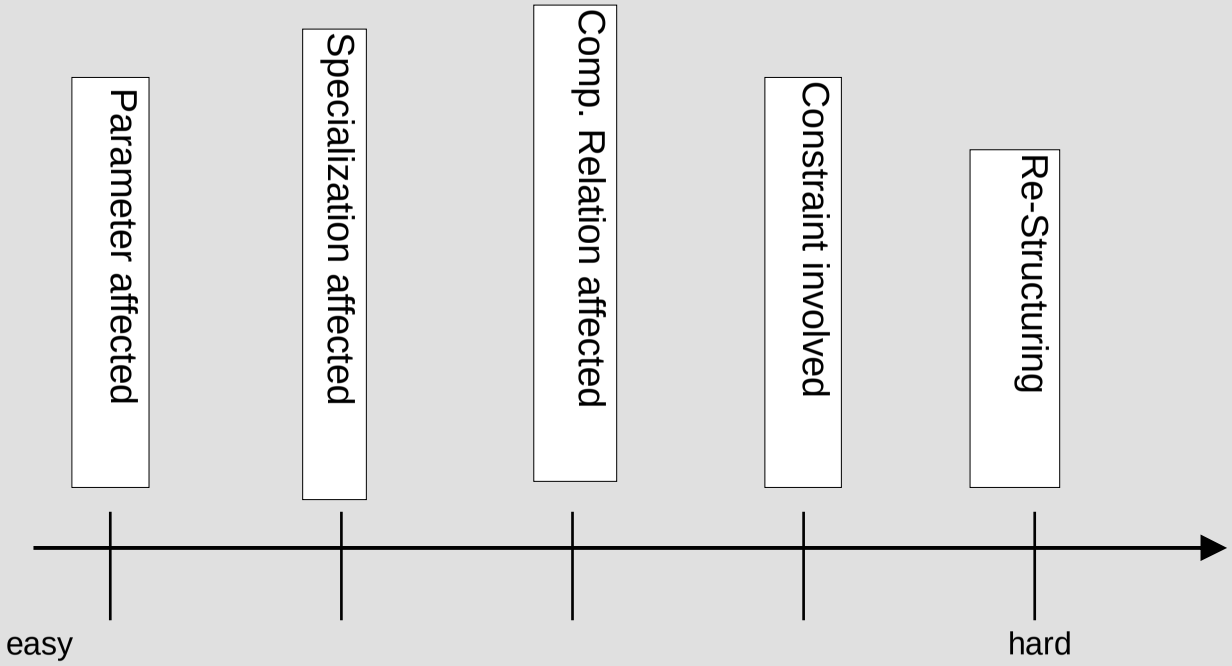
Operation	Model impacts	Configuration impacts
Adding new decompositions	Check the compositional conditions	May contain too less instances
Deleting decompositions	Check the compositional conditions	May contain too many instances
Adding constraints	Check the constraint conditions for the new constraint	May be inconsistent
Deleting constraints	No impacts	May be too narrow

Often only mixed operations make sense.

Degress of Modifications

- ➔ Easy to handle (e.g. adding parameters)
  - Modifications do not cause inconsistencies
  - Modifications do not entail further modifications
- ➔ Unproblematic
  - Impacts on several (sub)concepts (e.g. parameter changes)
- ➔ Problematic
  - Involving Constraints
  - Multiple Concepts
  - Support through dependency-analysis
- ➔ Critical
  - Complete restructuring of taxonomies and partonomies
  - E.g. Movement of subtrees
  - Configuration solutions are substantially changing

Degrees of Modifications



Support through the Model

- ➔ Declarative representations can be used for computing dependencies between knowledge entities
  - Impact analysis
  - Estimate costs for evolution
- ➔ Existing configurations can give hints for relevant modifications
- ➔ Support for the evolution of artifacts
  - Compute focused views on related concepts and relations
  - Automatic classification of new concepts
- ➔ Specific, previously known *evolution points* can be specified in the domain model

Configuration Process with Evolution

- ➔ Evolution initiation:
  - User starts the evolution process
  - A conflict occurs which is not wanted to be solved by changing the goal specification
- ➔ Change the model, possibly by being supported through the model
- ➔ Add the changes in the current configuration process
- ➔ Perform a new configuration with the old goal specification (reuse of user interactions)

Summary

- ➔ Configuration in a changing world
- ➔ Integrate domain modeling and configuration
- ➔ Integrate even, modeling, configuration, and **manufacturing**
- ➔ Types of modifications and their impacts
- ➔ Support of evolution tasks through declarative models

Thank You for Your Attention

Introduction

EU project with 4 partners:

- Industrial Partners:
  - Robert Bosch GmbH (project leader)
  - Thales Nederland B.V.
- University Partners:
  - Rijksuniversiteit Groningen
  - Universität Hamburg