

Fakultäten für Informatik und  
Maschinenwesen  
der Technischen Universität München

Interdisziplinäres Projekt

Model Transformations for  
Product-Service Systems

Bernhard Radke, Konstantin Govedarski



Fakultäten für Informatik und Maschinenwesen  
der Technischen Universität München

Interdisziplinäres Projekt

Model Transformations for  
Product-Service Systems

Verfasser:	Bernhard Radke Konstantin Govedarski
Aufgabensteller:	Prof. Dr.-Ing. Udo Lindemann
Betreuer:	Christopher Münzberg Danierl Kammerl Konstantin Kernschmidt Thomas Wolfenstetter
Submission Date:	15.04.2014



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Approach</b>	<b>5</b>
<b>3</b>	<b>Implementation</b>	<b>7</b>
<b>4</b>	<b>Results</b>	<b>9</b>
<b>5</b>	<b>Conclusion</b>	<b>11</b>
<b>A</b>	<b>Distribution of Tasks</b>	<b>13</b>



# Chapter 1

## Introduction

In present-day economic circumstances, businesses are presented with numerous challenges. Some of those challenges originate from the businesses themselves, like for example the strategic aim of growing and expanding into new markets, or the forging of customer and market awareness. Other challenges originate from the business and regulatory environment of the business in question. These environments force a business to be ever more competitive and to optimize for sustainability. To achieve this, industries no longer put the product itself at the core. Rather, they incorporate the product into a number of services, which reach from the lowest-level technical detail to the highest-level customer interaction and provide for optimal product utilization. In this sense, what a company tries to sell nowadays is not just the product itself, but a service through which the customer consumes the product, while being isolated from technical detail and unnecessary responsibility. Such complete solutions are denoted as Product-Service Systems (PSS).

While the incorporation of Product-Service Systems in an industry requires an initial effort and an evolution of the companies' cultures, it can lead to a more concise and efficient utilization of resources to the benefit of both the companies themselves and their customers.

### Importance of Integration

After their implementation and market introduction, Product-Service Systems can bring numerous benefits to a company. Nevertheless, their implementation brings a number of challenges of communication, integration and complexity. To explain this, let us consider the parties involved in the development and production cycles of a Product-Service System. Both development and production involve a number of disciplines, each acquainted with its own modelling languages and tools. Each of these domain-specific languages furthermore is concerned only with those aspects of the overall systems, which are relevant for the discipline at hand. As a result, each involved party has only a limited excerpt of the entire Product-Service System at its disposal. In the worst case, this leads to contradictions in the design and implementation of the system. In the moderate case,

it only incurs significant synchronization and management costs to the company. It is thus of crucial importance for the success of a PSS venture to establish a mechanism through which the models developed by different disciplines can be transformed to each other, or even to incorporate them all in a single model, representing the Product-Service System in its entirety.

## PSS Integration Framework

An approach to the integration of the disciplines involved in the development of a Product-Service System is researched jointly in the faculties of Computer Science and Mechanical Engineering of the Technical University of Munich by C. Münzberg, D. Kammerl, K. Kernschmidt and T. Wolfenstetter. The approach is named Product-Service Systems Integration Framework (PSS-IF) and provides a methodology and semantics for bringing the business, computer science and mechanical engineering aspects of a Product-Service System together. In its core, the framework describes a common structure which is sufficiently expressive to incorporate the PSS-wide relevant features of each domain-specific aspect of the PSS.

**TODO PSS-IF Metamodel pics and a bit more text.**

## Scope of the Interdisciplinary Project

In the context of the PSS Integration Framework (PSS-IF), the goal of this interdisciplinary project is to design and implement a Proof of Concept software utility which:

- follows the PSS-IF methodology and semantics
- can transform a model described in one relevant domain-specific language into another relevant domain-specific language
- considers the following domain-specific languages relevant:
  - Event-driven Process Chain (EPC) Diagrams
  - Business Process Modelling Notation (BPMN) Diagrams
  - Revenue-oriented Functional Modelling (UFP) Diagrams
  - Bringing Object-oriented Graph Grammars into Engineering (BOGGGIE) Diagrams
  - Systems Modelling Language for Mechatronics (SysML4Mechatronics) Diagrams
- should, at the time of its delivery, support at least two of the relevant domain-specific languages



## Structure of this Documentation

This documentation is structured as follows:

- **Chapter 2** discusses possible approaches to the conceptual structure of the Proof of Concept developed in this interdisciplinary project.
- **Chapter 3** describes the implementation of the Proof of Concept.
- **Chapter 4** provides the results obtained from the Proof of Concept.
- **Chapter 5** concludes this documentation and provides a few ideas about possible future developments on the basis of the provided Proof of Concept.
- **Appendix A** provides the distribution of tasks between the authors of the Proof of Concept.



## Chapter 2

# Approach

TODO

- what are possible ways of providing
- advantages and disadvantages for them
- argue for metamodeling
- define pssif metamodel (abstract)
- define transformations (abstract)



## Chapter 3

# Implementation

TODO

- technology: java. maven... explain why
- guiding principles of development
- project structure
- core
- transform
- vsdx
- viz?



## Chapter 4

# Results

TODO

- general evaluation of the resulting framework
- for each supported language, what worked, what did not





## Chapter 5

# Conclusion

TODO

- brief summary of what is achieved
- future work: speculate



# Appendix A

## Distribution of Tasks

This appendix describes the distribution of tasks between the developers of this Proof of Concept (PoC) implementation of the PSS Integration Framework.

### Research

The research necessary for the development of the PSS-IF PoC was done by both Bernhard Radke and Konstantin Govedarski.

### Conceptual Development

The conceptual development of the PSS-IF PoC was done by both Bernhard Radke and Konstantin Govedarski. Both authors are in comparable knowledge of the concepts and technology behind each component of the PoC system.

### Implementation

The implementation of the PSS-IF PoC was made by both Bernhard Radke and Konstantin Govedarski. The approximate distribution of specific tasks is provided below.

- **Utilities:** *Bernhard Radke and Konstantin Govedarski*
- **Core:**
  - **Metamodel:**
    - \* **Node and Edge Types:** *Bernhard Radke and Konstantin Govedarski*
    - \* **Connection Mappings:** *Bernhard Radke*
    - \* **Attributes:** *Konstantin Govedarski*
    - \* **Inheritance:** *Bernhard Radke*
  - **Model:** *Bernhard Radke and Konstantin Govedarski*

- **Operational API:** *Bernhard Radke*
- **Canonic Metamodel:** *Konstantin Govedarski*
- **Canonic Persistence:** *Bernhard Radke*
- **Transformations:**
  - **Views:** *Bernhard Radke*
  - **Operators:** *Bernhard Radke*
  - **API:** *Konstantin Govedarski*
  - **Generic Graph:** *Konstantin Govedarski*
- **Serialization and Deserialization:**
  - **GraphML:** *Bernhard Radke*
  - **Visio:** *Konstantin Govedarski*
  - **SysML4Mechatronics:** *Konstantin Govedarski*

## Documentation

The documentation of the PSS-IF PoC was created by both Bernhard Radke and Konstantin Govedarski. The documentation includes:

- This document.
- The documentation of the source code (JavaDocs).