
by

Tassio Ferreira Vale

M.Sc. Dissertation

RECIFE, MARCH/2012
Tassio Ferreira Vale


A M.Sc. Dissertation presented to the Center for Informatics of Federal University of Pernambuco in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Advisor: Silvio Romero de Lemos Meira
Co-Advisor: Eduardo Santana de Almeida

RECIFE, MARCH/2012
I dedicate this dissertation to my mother and my grandfather.
I would like to thank this dissertation to the following people:

my mother, Geruza Maria Tavares Ferreira,

my fiancée, Lorena de Jesus Costa e Silva,

my grandmother, Ivanildes Magalhães Valle,

my grandfather, Edson Santana Valle,

and my co-advisor, Eduardo Almeida.
Keep it simple, stupid.

—CLARENCE LEONARD “KELLY” JOHNSON (1910 - 1990)
Resumo

A combinação de Linhas de Produto de Software (LPS) e Computação Orientada a Serviços (COS) tem recebido atenção de pesquisadores e praticantes, já que uma área é capaz de solucionar determinados problemas da outra. A junção dessas duas áreas é chamada de Linha de Produto Orientada a Serviços (LPOS), e tem se mostrado uma área de pesquisa emergente nos últimos anos.

No contexto da Computação Orientada a Serviços, identificação de serviços é uma das primeiras atividades para modelagem de uma solução COS, e consiste na determinação de serviços candidatos. Essa não é uma tarefa trivial, e os erros adquiridos durante a identificação de um serviço pode propagar-se para as atividades subsequentes do processo de desenvolvimento orientado a serviços.

Existem diversas propostas para identificação de serviços. Apesar da diversidade, não existe um método de identificação de serviços capaz de adequar-se a diversos contextos e necessidades. Considerando a heterogeneidade desses métodos, esta dissertação tem o objetivo de verificar quais métodos podem ser aplicados em linhas de produto de software, bem como as vantagens, desvantagens e desafios existentes neste campo.

Esta pesquisa utilizou uma abordagem multi-métodos, que combina estudos primários e secundários a fim de incrementar o corpo de conhecimento em determinada área baseada nos achados de tal investigação. Desta forma, este trabalho utilizou um método secundário (revisão sistemática) e dois métodos primários com o intuito de aumentar a confiabilidade dessa pesquisa.

Além disso, foi realizada uma revisão da literatura a fim de coletar as abordagens de identificação de serviços existentes, visando propor uma recomendação dos métodos mais adequados considerando três cenários de LPS: top-down, bottom-up e hybrid. Esta recomendação foi avaliada através de um estudo de caso com uma linha de produto de sistemas médicos, usando dois métodos classificados no cenário o qual os sistemas médicos encontram-se, top-down.

A principal contribuição dessa pesquisa é prover um instrumento que auxilie os interessados em identificar serviços a escolher um método apropriado, levando em consideração o cenário SPL a ser aplicado. Os resultados do estudo de caso mostram que as abordagens de identificação de serviços podem ser aplicadas num contexto de LPS. Além disso, este métodos facilitam a realização da atividade de identificação.

**Palavras-chave:** Linha de Produto de Software, Linha de Produto de Software Orientada a Serviços, Computação Orientada a Serviços, Identificação de Serviços.
The combination of Software Product Line (SPL) and Service-Oriented Computing (SOC) have started to receive attention by researchers and practitioners, since they can address issues of each other. Putting these two areas together is called Service-Oriented Product Lines Engineering (SOPLE), presenting itself as an emerging area in the last years.

In the Service-Oriented Computing, service identification is one of the first activities in the modeling of a SOC solution, that consists of determining candidate services. This is not a trivial task, and the errors made during the identification can propagate mistakes to the next activities of the service-oriented development process.

There are several proposals addressing service identification for several contexts. However, an unified method for identifying services has not yet been reached. Regarding the heterogeneity of these methods, this dissertation aims to verify which methods can be applied in the SPL context, the advantages, disadvantages and the existing challenges of this field.

This research used a multi-method approach that combines primary and secondary studies in order to increase the availability of empirical knowledge based on the findings of the investigation. Thus, this work used one secondary method (systematic review) and two primary methods (survey and case study) to address it.

Furthermore, it was performed a literature review in order to collect all existing approaches of service identification, aiming to propose a set of decision models that recommend the most suitable methods according to three SPL scenarios: bottom-up, top-down and hybrid, onde decision model for each scenario. One of the decision models was evaluated through a case study in a medical applications domain, using two approaches classified in the decision model of the top-down scenario.

The main contribution of this research is to provide an instrument that can help the service identification stakeholders to choose a suitable method, taking consideration of their SPL scenarios. In addition, these systematic methods facilitate the application of the identification activity. The case study results evaluated some service identification approaches and presented evidence that the methods can be applied in SPL.

**Keywords:** Software Product Lines, Service-Oriented Product Lines, Service-Oriented Computing, Service Identification.
# Table of Contents

## List of Figures

## List of Tables

## List of Acronyms

## 1 Introduction

1.1 Motivation ...................................................... 1
1.2 Scope ............................................................ 2
   1.2.1 Context .................................................... 2
1.3 Out of Scope .................................................... 4
1.4 Research Questions ............................................. 5
1.5 Research Design ................................................ 6
   1.5.1 A Multi-Method Research Methodology ..................... 8
1.6 Related Work .................................................. 9
1.7 Main Contributions ............................................ 9
1.8 Dissertation structure ......................................... 10

## 2 An Overview on Software Product Lines and Service-Oriented Computing

2.1 Software Product Lines (SPL) ................................. 11
   2.1.1 Introduction ............................................... 11
   2.1.2 The SPL Development Process ........................... 12
   2.1.3 The Benefits ............................................... 14
2.2 Service-Oriented Computing (SOC) ........................... 15
   2.2.1 Introduction ............................................... 15
   2.2.2 Some Definitions ......................................... 16
   2.2.3 The Principles ............................................. 17
   2.2.4 The Service-Oriented Development Process ............... 19
   2.2.5 The Benefits ............................................... 20
2.3 Service Identification ........................................ 22
2.4 Chapter Summary .............................................. 23

## 3 A Systematic Mapping Study on Service-Oriented Product Line Engineering

3.1 Methodology .................................................... 26
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>Research Questions</td>
<td>27</td>
</tr>
<tr>
<td>3.3</td>
<td>Inclusion and exclusion criteria</td>
<td>28</td>
</tr>
<tr>
<td>3.4</td>
<td>Search strategy</td>
<td>29</td>
</tr>
<tr>
<td>3.5</td>
<td>Screening of Papers</td>
<td>30</td>
</tr>
<tr>
<td>3.6</td>
<td>Classification scheme</td>
<td>32</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Research method</td>
<td>32</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Service-Oriented Computing (SOC) Topics</td>
<td>33</td>
</tr>
<tr>
<td>3.6.3</td>
<td>Software Product Line (SPL) Topics</td>
<td>34</td>
</tr>
<tr>
<td>3.6.4</td>
<td>Proposal type</td>
<td>34</td>
</tr>
<tr>
<td>3.6.5</td>
<td>Application type</td>
<td>35</td>
</tr>
<tr>
<td>3.7</td>
<td>Results</td>
<td>35</td>
</tr>
<tr>
<td>3.8</td>
<td>Analysis</td>
<td>44</td>
</tr>
<tr>
<td>3.9</td>
<td>Threats to Validity</td>
<td>47</td>
</tr>
<tr>
<td>3.10</td>
<td>Chapter Summary</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>Service Identification Methods Comparison and Classification</td>
<td>49</td>
</tr>
<tr>
<td>4.1</td>
<td>Literature Review</td>
<td>49</td>
</tr>
<tr>
<td>4.1.1</td>
<td>SPL Scenarios</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>Criteria Evaluation</td>
<td>55</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Survey</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>Setting the Objectives and Survey Design</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Developing and Evaluating the Survey Instrument</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Expert Opinion Process</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Survey Analysis</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>Methods Comparison</td>
<td>61</td>
</tr>
<tr>
<td>4.4</td>
<td>Methods Classification</td>
<td>64</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Decision Models Building</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>Pairwise Comparison</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Similarity Analysis</td>
<td>65</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Resulting Decision Models</td>
<td>66</td>
</tr>
<tr>
<td>4.5</td>
<td>Threats to Validity</td>
<td>68</td>
</tr>
<tr>
<td>4.6</td>
<td>Chapter Summary</td>
<td>68</td>
</tr>
<tr>
<td>5</td>
<td>The Case Study</td>
<td>76</td>
</tr>
<tr>
<td>5.1</td>
<td>Case Study Protocol</td>
<td>76</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Objective</td>
<td>76</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.1.2</td>
<td>The Case</td>
<td>77</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Units of Analysis</td>
<td>78</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Case Study Research Questions</td>
<td>80</td>
</tr>
<tr>
<td>5.1.5</td>
<td>Data Collection</td>
<td>81</td>
</tr>
<tr>
<td>5.1.6</td>
<td>Data Analysis</td>
<td>82</td>
</tr>
<tr>
<td>5.2</td>
<td>Results and Findings</td>
<td>83</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Kang et al. (2008) method</td>
<td>83</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Lee et al. (2008) method</td>
<td>84</td>
</tr>
<tr>
<td>5.3</td>
<td>Threats to Validity</td>
<td>85</td>
</tr>
<tr>
<td>5.4</td>
<td>Chapter Summary</td>
<td>86</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
<td>87</td>
</tr>
<tr>
<td>6.1</td>
<td>Research Contributions</td>
<td>88</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>89</td>
</tr>
<tr>
<td>6.3</td>
<td>Concluding Remarks</td>
<td>89</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>91</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
<td>106</td>
</tr>
<tr>
<td>B</td>
<td>Survey</td>
<td>110</td>
</tr>
<tr>
<td>B.1</td>
<td>SPL Survey</td>
<td>110</td>
</tr>
<tr>
<td>B.1.1</td>
<td>Invitation Letter</td>
<td>110</td>
</tr>
<tr>
<td>B.1.2</td>
<td>Chronology</td>
<td>110</td>
</tr>
<tr>
<td>B.1.3</td>
<td>Questionaire</td>
<td>111</td>
</tr>
<tr>
<td>B.2</td>
<td>SS Survey</td>
<td>114</td>
</tr>
<tr>
<td>B.2.1</td>
<td>Invitation letter</td>
<td>114</td>
</tr>
<tr>
<td>B.2.2</td>
<td>Chronology</td>
<td>115</td>
</tr>
<tr>
<td>B.2.3</td>
<td>Questionaire</td>
<td>116</td>
</tr>
<tr>
<td>C</td>
<td>Pairwise Comparison Results</td>
<td>118</td>
</tr>
<tr>
<td>D</td>
<td>Case Study Interview</td>
<td>120</td>
</tr>
</tbody>
</table>
## List of Figures

1.1 RiSE Labs Influences ................................................. 3
1.2 RiSE Labs Projects .................................................. 4
1.3 Research Design ...................................................... 6
1.4 Literature Review and Criteria Evaluation activities .......... 7
1.5 Criteria Evaluation and Methods Comparison activities .... 7
1.6 Methods Comparison and Methods Classification activities .. 8
1.7 Methods Classification and Methods Evaluation activities ... 8
1.8 Research design and multi-method approach ...................... 9

2.1 SPL Activities (Clements and Northrop, 2001) ................. 12
2.2 Core Asset Development (Clements and Northrop, 2001) .... 13
2.3 Product Development (Clements and Northrop, 2001) ......... 14
2.4 Service-Oriented Lifecycle (Gu and Lago, 2007) ............. 25

3.1 Systematic Mapping Process by Petersen et al. (2007) ...... 27
3.2 Screening process ..................................................... 31
3.3 Intensity of the research activity in Service-Oriented Product Line Engineering (SOPLE) area .................................. 36
3.4 Journal, conference and workshop papers per year .......... 39
3.5 Motivation of integrating SOC and SPL ......................... 39
3.6 Research methods ...................................................... 41
3.7 Research methods per year .......................................... 42
3.8 Top seven application domains ..................................... 43
3.9 Application type ....................................................... 44
3.10 SOC topics that are influenced by the SOPLE research .... 45
3.11 SPL topics that are influenced by the SOPLE research .... 46
3.12 Proposal types ......................................................... 46
3.13 SOC topics x SPL topics x proposal types ..................... 47

4.1 Proactive approach (Krueger, 2002) .............................. 52
4.2 Reactive approach (Krueger, 2002) ............................... 53
4.3 Extractive approach (Krueger, 2002) ............................. 54
4.4 Bastos (2011) survey process ...................................... 57
4.5 Decision making process ............................................. 67
5.1 Company products portfolio. ........................................... 77
5.2 Kang et al. (2008) identification process. .......................... 79
5.3 Lee et al. (2008) identification process. .............................. 79
List of Tables

3.1 Search String ................................................. 29
3.2 Automatic Search - Results .................................. 30
3.3 Manual Search - Journals .................................... 30
3.4 Manual Search - Conferences ................................ 31
3.5 Ranking of countries that have published about SOPLE .... 37
3.6 Papers per journal, conference and workshop ............ 38
3.7 Enterprise focus .............................................. 42
4.1 Methods Comparison - Part I .................................. 69
4.2 Methods Comparison - Part II ................................ 70
4.3 Methods Comparison - Part III .............................. 71
4.4 Weight assignment: all criteria .............................. 71
4.5 Similarity analysis: Top-down scenario ....................... 72
4.6 Similarity analysis: Bottom-up scenario ..................... 72
4.7 Similarity analysis: Hybrid scenario ......................... 72
4.8 Decision model: top-down scenario ......................... 73
4.9 Decision model: bottom-up scenario ......................... 74
4.10 Decision model: hybrid scenario ............................ 75
A.1 Overview of the existing methods ......................... 107
A.2 Legend: strategy .............................................. 108
A.3 Legend: output type .......................................... 109
C.1 Weight assignment: service granularity .................... 118
C.2 Weight assignment: output artifacts ......................... 118
C.3 Weight assignment: techniques/activities ................... 118
C.4 Weight assignment: orchestration vs. coreography ........ 119
C.5 Weight assignment: research method ......................... 119
C.6 Weight assignment: research rigor .......................... 119
C.7 Weight assignment: ........................................... 119
C.8 Weight assignment: reuse .................................... 119
C.9 Weight assignment: Variability .............................. 119
C.10 Weight assignment: Business aspects ...................... 119
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>BPM</td>
<td>Business Process Management</td>
</tr>
<tr>
<td>CBD</td>
<td>Component-Based Development</td>
</tr>
<tr>
<td>COBOL</td>
<td>COrmon Business Oriented Language</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>DCOM</td>
<td>Distributed Component Object Model</td>
</tr>
<tr>
<td>EAI</td>
<td>Enterprise Application Integration</td>
</tr>
<tr>
<td>EJB</td>
<td>Enterprise Java Beans</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>OO</td>
<td>Object-Orientation</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>ROI</td>
<td>Return On Investment</td>
</tr>
<tr>
<td>RPC</td>
<td>Remote Procedure Call</td>
</tr>
<tr>
<td>SIM</td>
<td>Service Identification Method</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SPL</td>
<td>Software Product Line</td>
</tr>
<tr>
<td>SO</td>
<td>Service-Orientation</td>
</tr>
<tr>
<td>SOA</td>
<td>Service-Oriented Architecture</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SOC</td>
<td>Service-Oriented Computing</td>
</tr>
<tr>
<td>SOPLE</td>
<td>Service-Oriented Product Line Engineering</td>
</tr>
<tr>
<td>SOSE</td>
<td>Service-Oriented Software Engineering</td>
</tr>
<tr>
<td><strong>WSDL</strong></td>
<td>Web Service Description Language</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td><strong>XML</strong></td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
1

Introduction

1.1 Motivation

Nowadays, businesses require constant changes, and the companies must adapt their processes and reduce costs according to market conditions and customer requirements. This pressure is also present in the software that supports the business processes, since companies need to decrease development costs, improve software quality and maximize productivity.

Software reuse is a way to achieve these business goals. It is the process of creating software systems from existing software rather than building software systems from scratch (Krueger, 1992). Software Product Line (SPL) engineering establishes a systematic software reuse strategy, through which organizations identify common functionality and variability among applications within a domain, and build reusable assets to benefit future development efforts (Pohl et al., 2005).

On the other hand, Service-Oriented Computing (SOC) has emerged to achieve the IT efficiency and business agility. SOC is a computing paradigm that utilizes services as fundamental elements to develop applications (Papazoglou and Georgakopoulos, 2003).

In order to achieve the benefits highlighted by these two areas, putting SPL and SOC together have started to receive attention by researchers and practitioners (Bartholdt et al., 2008; Rusk and Gasevic, 2008). Furthermore, they can address issues of each other, and this combination of SPL and SOC is called Service-Oriented Product Line Engineering (SOPLE) (Trujillo et al., 2007a; Günther and Berger, 2008; Parra et al., 2009).

Regarding SPL, SOC can achieve dynamic reconfiguration, since in most of the SPL approaches (Clements and Northrop, 2001; Pohl et al., 2005) all variations are instantiated before a product is delivered to the customers, making it difficult for them to make any
changes to the product. In addition, for SOC, SPL can cover several issues, such as: an approach to manage variation among multiple service-oriented systems, determining relevant configurations of services to the users’ current context, maintaining the system integrity after configuration changes, and service identification (Lee et al., 2008; Cohen, 2010).

The last cited SOC issue, service identification, is the target of this work. The service identification is one of the first activities in the modeling of a service-oriented solution (Inaganti and Behara, 2007; Kim et al., 2008) and it consists of determining candidate services. This is not a trivial task, and several methods were proposed to address the identification activity (Gu and Lago, 2010). In addition, the mistakes made during the identification can propagate problems to the next activities of the service-oriented development process.

In the literature, there are several proposals (Erradi et al., 2007; Arsanjani et al., 2008; Kang et al., 2008; Lee et al., 2008) of service identification for several contexts and domains. However, an unified method for identifying services has not yet been reached (Gu and Lago, 2010). Thus, this dissertation aims to investigate which methods can be applied in a SOPLE context.

1.2 Scope

Regarding the intersection between service-oriented practices and product line techniques analyzed in a mapping study (see Chapter 3), this dissertation attempts to investigate the application of service identification methods in order to build service-oriented product lines, its advantages, drawbacks and the existing challenges in this field.

In this context, this research was designed based on a set of questions that guide all dissertation activities in the whole process. Sections 1.4 and 1.5 will explain the defined research questions and the research process/design, respectively.

1.2.1 Context

This dissertation is part of the RiSE Labs\(^1\) (Almeida et al., 2004), formerly called RiSE Project, whose goal is to develop a robust framework for software reuse in order to enable the adoption of a reuse program. RiSE Labs is influenced by a series of areas, such as software measurement, architecture, quality, environments and tools, and so on, in order

\(^1\)http://labs.rise.com.br
to achieve its goal. The influence areas are showed in Figure 1.1.

Figure 1.1 RiSE Labs Influences

Based on these areas, the RiSE Labs is divided in several different projects related to software reuse, as shown in Figure 1.2:

- **RiSE Framework:** Involves reuse processes (Almeida et al., 2004), component certification (Alvaro et al., 2006) and reuse adoption processes (Garcia et al., 2008).

- **RiSE Tools:** Research focused on software reuse tools, such as the Admire Environment (Mascena, 2006), the Basic Asset Retrieval Tool (B.A.R.T) (Santos et al., 2006), which was enhanced with folksonomy mechanisms (Vanderlei et al., 2007), semantic layer (Durao, 2008), facets (Mendes, 2008) and data mining (Martins et al., 2008), and the Legacy InFormation retrieval Tool (LIFT) (Brito, 2007), the Reuse Repository System (CORE) (Melo et al., 2008), and the Tool for Domain Analysis (ToolDAy) (Lisboa et al., 2007; Lisboa, 2008).

- **RiPLE:** Stands for RiSE Product Line Engineering Process and aims at developing a methodology for Software Product Lines, composed of scoping (Moraes, 2010), requirements engineering (Neiva, 2009), design (Diorenzo et al., 2009; Cavalcanti et al., 2011), implementation, test (Neto, 2010; Machado, 2010), and evolution management (Oliveira, 2009).
1.3. OUT OF SCOPE

- **SOPLE**: Development of a methodology for Service-Oriented Product Lines, based on the fundamentals of the RiPLE (Ribeiro et al., 2011). This dissertation is part of the SOPLE Project.

- **MATRIX**: Investigates the area of measurement in reuse and its impact on quality and productivity;

- **BTT**: Research focused on tools for detection of duplicated change requests (Cavalcanti, 2009; Cavalcanti et al., 2009).

- **Exploratory Research**: Investigates new research directions in software engineering and its impact on reuse;

- **CX-Ray**: Focused on understanding the C.E.S.A.R. ², and its processes and practices in software development.

1.3 Out of Scope

This work is part of a complete framework (Almeida et al., 2004; Alvaro et al., 2006; Garcia et al., 2008) for software product lines development that has been designed. However, some SPL and SOC topics are not described in this work, and some specific issues regarding SOPLE are not considered in the scope of this dissertation, such as:

²http://www.cesar.org.br/
1.4. RESEARCH QUESTIONS

- **Other activities in the service-oriented design process.** This work focuses on service identification. Other activities in the service-oriented design process are not covered in this dissertation, such as: market scan, business modeling, service development, service testing, service deployment, and so on (Erl, 2005; Gu and Lago, 2007).

- **Single systems.** This work is concerned to investigate the behavior of the service identification methods in SPL scenarios, characterizing a SOPLE context - since it combines Service-Oriented Software Engineering (SOSE) and SPL aspects. We do not investigate the application of service identification methods in single system scenarios.

- **A new service identification method.** The goal of this work is to evaluate the existing service identification methods to determine the most suitable ones given a set of SPL scenarios. It does not address a new service identification method.

### 1.4 Research Questions

As part of the SOPLE Project, the main research question of this work can be stated as following:

**Can the available service identification methods be applied in the SPL context?**

This question was defined in the beginning of this work, considering the SOPLE area and focusing specifically in the service identification practice, that can be applied to SPL. Therefore, the idea is to investigate if the methods can be really used, identify and understand their strong points and drawbacks, and if possible, characterize the most suitable ones.

In the main question, the SPL context is analyzed (instead SOPLE context), because an assumption of this research is that the combination of a SOC technique (service identification) and SPL practices already characterizes an investigation on the SOPLE area.

In this sense, the main question was divided in three research questions:

- **RQ1. What are the most significant criteria to compare and classificate the service identification methods?**
1.5. RESEARCH DESIGN

Rationale: the recommendation of a suitable method is a result of comparing and classifying all existing service identification methods. It can be performed based on a set of criteria, and in this question, the goal is to collect the most significant criteria. Significant criteria are important to achieve a consistent comparison and classification, that will indicate the appropriate method for the stakeholder.

• **RQ2. What characterizes a suitable service identification method?**
  Rationale: this question aims to find out the desirable characteristics for a suitable method, and consequently, which criteria are used to compare and classify the service identification methods, in order to determine their suitability.

• **RQ3. Which are the most suitable methods given specific SPL scenarios?**
  Rationale: besides investigating if the methods can be applied in the SPL context, the goal is to provide to the stakeholders an appropriate recommendation of the most suitable methods given some SPL scenarios, such as: top-down, that considers only “high-level” artifacts to start building a product line, bottom-up, considering only source code, and hybrid, a mix of the previous scenarios. These recommendations can help the stakeholders to choose the appropriate service identification method according to their needs.

1.5 Research Design

The research process to answer the research questions is summarized in Figure 1.3. It consists of five steps: literature review, criteria evaluation, methods comparison, methods classification and methods evaluation. These steps will be explained in details next.

![Figure 1.3 Research Design.](image-url)
• **Literature Review**

The first activity comprises of identifying the existing methods in the literature as well as the existing criteria to compare and classify the methods (Figure 1.4). It is an input for the criteria evaluation and methods comparison activities, and it will be explained in details afterwards (Chapter 4).

![Figure 1.4 Literature Review and Criteria Evaluation activities.](image)

• **Criteria Evaluation**

This activity consists of analyzing the existing criteria provided from the literature review, in order to select the most significant criteria - according to expert opinion - to compare and classify the service identification methods (Figure 1.5). It is an input for the methods comparison activity, and it will be explained in the Chapter 4.

![Figure 1.5 Criteria Evaluation and Methods Comparison activities.](image)

• **Methods Comparison**

In this step, the existing methods identified in the literature review and the most significant criteria from the criteria evaluation activity are used to perform a criteria-based methods comparison. The result will be an input for the methods classification step (Figure 1.6).

• **Methods Classification**

The methods classification activity uses the comparison results and criteria (with their relevances) of the previous steps to rank, through decision models, the most suitable methods according to some defined SPL scenarios.
1.5. RESEARCH DESIGN

Figure 1.6 Methods Comparison and Methods Classification activities.

Figure 1.7 Methods Classification and Methods Evaluation activities.

- **Methods Evaluation**

  The accuracy of the classifications provided by the resulting decision models will be evaluated in the methods evaluation activity. In this sense, some service identification methods will be taken to be evaluated using a case study (Runeson and Höst, 2009). They will be applied in real-world projects that correspond to the defined scenarios in the section 4.1.1.

1.5.1 A Multi-Method Research Methodology

A multi-method approach (Brewer and Hunter, 1989), also known as triangulation (Martin, 1982), combines different research methods. It can help to address the perceived weaknesses of single studies by attacking research problems “with an arsenal of methods that have non-overlapping weaknesses in addition to their complementary strengths” (Brewer and Hunter, 1989).

The multi-method research methodology, based on O’Leary (2010) and Bastos (2011), focuses on combining primary (e.g. case study, survey, expert opinion) and secondary (e.g. systematic review, mapping study) studies, in order to increase the availability of empirical knowledge based on the findings of the investigation.

This multi-method approach influenced the research design of this work. The empirical (primary and secondary) studies are incorporated in the research design as showed in Figure 1.8. The literature review is supported by the Gu and Lago (2010) systematic review, the criteria to compare and classify the service identification methods are
validated by the experts through a survey (Fink, 1995), and the case study research is performed aiming to evaluate some methods in a SPL scenario.

![Research design and multi-method approach.](image)

**Figure 1.8** Research design and multi-method approach.

## 1.6 Related Work

Boerner and Goeken (2009) proposed a comparison of five approaches for service identification, using a catalogue consisting of forty-two selected criteria in order to compare and give an overview of the identified approaches in their related literature. Some criteria have already been used by other researchers (e.g. service hierarchy, orchestration vs. choreography, granularity, industry sector, and so on.) (Erl, 2004; Josuttis, 2008), and others have been added to complement the existing ones.

Kim and Doh (2009) evaluated their service identification approach comparing it with three other approaches. They used a set of five criteria: background & starting point, service classification scheme, covering of SOA design phases, characteristics, and application of process models for service identification.

These work did not perform a systematic review to select the relevant primary studies. Furthermore, the criteria were defined by the researchers opinions, and there is not an evaluation of the appropriate criteria to analyze, in a comprehensive way, the service identification methods.

## 1.7 Main Contributions

Aiming to answer the research questions, this work provided some research contributions:

- We performed a SOPLE mapping study to understand the characteristics and research gaps of the area.
1.8. DISSERTATION STRUCTURE

- This work extended the Gu and Lago (2010) literature review on service identification methods in order to select papers that were not considered in the systematic review.
- A set of evaluated criteria to compare and classificate service identification methods.
- A detailed comparison involving the existing service identification methods collected from the literature.
- Three decision models considering three SPL scenarios. These decision models will help the stakeholder to select the most suitable method according to his needs.
- A case study considering two service identification methods in a real-life SOPLE context.

1.8  Dissertation structure

The remainder of this dissertation is organized as follows:

- **Chapter 2** reviews the essential topics used throughout this work: Service-Oriented Computing, Software Product Lines and Service Identification Methods.
- **Chapter 3** presents the mapping study performed to investigate the benefits, drawbacks and challenges faced by the emerging SOPLE area.
- **Chapter 4** explains in details the three analysis activities of the research design: literature review, criteria evaluation and methods comparison.
- **Chapter 5** describes the remaining activity of the research design: the methods evaluation activity, i.e., the case study. It will present in details: the protocol, data collection, data analysis, outcomes, and so on.
- **Chapter 6** provides the concluding remarks. It discusses our contributions, limitations, and outline directions for future work.
2

An Overview on Software Product Lines and Service-Oriented Computing

This Chapter concerns the understanding of three important topics for this dissertation: Software Product Lines, Service-Oriented Computing and Service Identification. Section 2.1 discusses SPL, its characteristics, development processes and benefits. Section 2.2 explains the SOC aspects, development processes and benefits of this area, and Section 2.3 addresses the state-of-the-art in service identification. Finally, Section 2.4 presents a summary of this Chapter.

2.1 Software Product Lines (SPL)

2.1.1 Introduction

In the 70s, Parnas (1976) proposed the concept of product families, and after that, roughly twenty years later, the concept of software product lines was introduced. The move towards Software Product Lines (SPL) is strongly based on economic considerations, and many adoptions of the SPL approach are driven by cost and time-to-market concerns (Linden et al., 2007).

The SPL principles emerged from automobile manufactures, which enable mass production cheaper than individual product creation. These manufactures use a common platform to derive products that can be customized to specific customers or market segments needs (Clements and Northrop, 2001). In a software engineering perspective, the combination of mass customization, large-scale production, and the use of a common platform to derive products, resulted in the software product line engineering paradigm (Pohl et al., 2005).
An SPL can be defined as a set of similar software intensive systems that share a collection of common features satisfying the needs of specific customers or market segments. Those similar software systems are developed from a set of core assets, comprised of documents, specifications, components, and other software artifacts that can be reusable during the development of each system in the product line (Clements and Northrop, 2001).

The goal of the software product line engineering is to exploit the commonality of a set of similar systems, while managing variability among these systems with the purpose of developing a family of customized products faster and cheaper than creating each individual product separately (Gomaa, 2004). In this sense, the software product lines development paradigm uses a specific development process, explained next.

### 2.1.2 The SPL Development Process

Pohl et al. (2005), Clements and Northrop (2001) proposed a set of activities to create a software product line. In essence, the two approaches comprises the activities of domain engineering and application engineering.

Building a product line involves core asset development (domain engineering) and product development (application engineering) using the core assets. Both activities have technical and organizational management (Clements and Northrop, 2001), as showed in Figure 2.1.

![Figure 2.1 SPL Activities (Clements and Northrop, 2001).](image)

The core asset development is an iterative activity to establish a production capability for products. Figure 2.2 shows these processes, as well as the outputs and the necessary
inputs. There are several inputs for this process, such as: product and production constraints, architectural styles, design patterns, application frameworks, production strategy and, finally, legacy systems.

![Diagram](image.png)

**Figure 2.2** Core Asset Development (Clements and Northrop, 2001).

In addition three main outputs are generated: *product line scope* (also called scoping), *core assets* and the *production plans*. The first output, scoping, describes the products that will constitute the product line and (if exists) the products that the product line is capable of including. This description can be simple or detailed, including, for example, a list of products, or features, operations, quality attributes and other necessary properties.

The core assets are the basis for the development of products. It includes an architecture that specifies the structure of the products and software components that are developed for systematic reuse. Some examples of other assets are performance models, architecture evaluation results, test plans, test cases, and so on. The core assets should have also associated attached process that specifies how they will be used in the products.

A production plan describes how products are produced from the core assets. It is the overall scheme of how the individual attached processes of the core assets can be fitted together to build a product. The production plan is the link among the core assets in order to utilize them effectively and within the constraints of the product line.

As showed in Figure 2.3, *product development* is the activity responsible for building products in a product line. This activity depends on the artifacts built in the core assets development (product line scope, core assets, and production plan) and the requirements specification for individual products. Those requirements are a variation of the generic product description contained in the product line scope.
2.1. SOFTWARE PRODUCT LINES (SPL)

The creation of products may have a feedback effect on the output artifacts of the core assets development and the requirements for specific products. This happens because some insights can be identified just during the product development process.

Finally, the management activity is performed in two levels: technical and organizational. The technical management ensures (for core assets development and product development activities) the definition of the processes for the product line, the engagement of the groups in the required activities, and the progress tracking. The organizational management must optimize the organizational structure and ensure that the organizational units receive the right resources in sufficient amounts.

2.1.3 The Benefits

Software product lines techniques offer a set of benefits for companies applying it (Clements and Northrop, 2001; Pohl et al., 2005; Linden et al., 2007), as follows:

- **Cost reduction**

  When the core assets are reused in several products, this implies in cost reduction for each product. However, before the core assets reuse, investments are necessary for creating them. In addition, the way in which they will be reused has to be planned in advance in order to provide managed reuse.

- **Quality improvement**
The core assets are reviewed and tested in many products. Thus, they have to prove their proper functionality in more than one specific product. The quality assurance implies a significantly higher chance of detecting and correcting faults, resulting in the higher quality of all products.

- **Reduction of time-to-market**

  Time-to-market in a product line is initially higher, because core assets have to be built first. However, after this building step, the time-to-market is reduced due the reuse of core assets across the products in the product line.

- **Other benefits**

  Other benefits provided from software product lines are the abilities to: maintain market presence, sustain unprecedent growth, increase customer satisfaction, and, enable mass customization.

As this work involves the combination of software product lines and service-oriented computing, the next sections present the main aspects of **SOC**.

## 2.2 Service-Oriented Computing (SOC)

### 2.2.1 Introduction

**SOC** is a computing paradigm that brings several successful concepts from previous technologies used in real-world systems. Basically, the roots of service-orientation can be found in three areas: programming paradigms, distributed computing and business computing (Krafzig *et al.*, 2004).

Since 70s, the programming languages incorporated the concept of abstractions from details of software functionality. The first used technique to achieve abstraction was the functional decomposition, and C0mmon Business Oriented Language (COBOL) was the first language to support this technique.

Based on the growing software complexity, the concept of software components was introduced, and a similar concept was introduced in programming languages, called encapsulation. Both focus on hiding the function internal details.

The significant increase of abstraction is an important step towards service-orientation. Abstraction is one of the key principles of **SOC** and it is a result of the influence leveraged for the abstraction in programming paradigms.
2.2. SERVICE-ORIENTED COMPUTING (SOC)

The distributed computing was available on the business environments in the 1990s, and several core-application remoting technologies were created such as Remote Procedure Call (RPC), Common Object Request Broker Architecture (CORBA), Enterprise Java Beans (EJB), and Distributed Component Object Model (DCOM) (Emmerich and Kaveh, 2001). In that time, we could see also the emergence of a large number of additional distributed computing middleware solutions, addressing transaction management, messaging, Enterprise Application Integration (EAI) (Gorton and Liu, 2004), security, and so on.

Nevertheless, this variety of technologies caused a problem: the middleware heterogeneity. The Extensible Markup Language (XML) became popular in the mid 1990s as a middleware-independent format for exchanging data among different applications. However, the XML (Kuchling, 1998) flexibility requires higher-level data structure and messaging formats.

Recognizing this issue, Microsoft invented XML-based Web Services, creating Simple Object Access Protocol (SOAP) (Curbera et al., 2002), that enables an easy server-to-server communication. Microsoft later developed an interface definition language for SOAP, called Web Service Description Language (WSDL) (Curbera et al., 2002). Web Services has influenced SOC and can be a platform to perform it.

In addition to the technical aspects (programming languages and distributed systems), business computing also influenced SOC concepts. SOC aims at leveraging data and business logic that resides in many applications, offering flexible adaptation without imposing the use of a single middleware or EAI platform.

2.2.2 Some Definitions

There are several terms and definitions in the literature about SOC concepts. However, the key terms considered in this dissertation are: Service-Oriented Computing (SOC), Services, and Service-Oriented Architecture (SOA).

Service-Orientation has been regarded the next step after Object-Orientation (OO) and Component-Based Development (CBD) (Szyperski, 2002; Pallos, 2001), providing a different - higher - abstract level from the object or component (Kim and Doh, 2007).

SOC is the computing paradigm that utilizes services as fundamental elements for developing applications (Huhns and Singh, 2005; Papazoglou and Georgakopoulos, 2003; Zhang et al., 2005). Services are self-describing, open components that support rapid, low-cost composition of distributed applications (Papazoglou and Georgakopoulos, 2003). They are autonomous, platform-independent entities that can be described, published,
discovered, and loosely coupled in novel ways (Papazoglou et al., 2007).

The keystone, for services, is the SOA (Papazoglou et al., 2007). It establishes an architectural model that aims to enhance the efficiency and agility of an enterprise by positioning services as the primary means (Erl, 2007).

### 2.2.3 The Principles

Papazoglou et al. (2007) argue that service-orientation is a design paradigm comprised of a set of principles, which are: contract, loose coupling, abstraction, reusability, autonomy, statelessness, discoverability and composability. The principles, based on Erl (2005); Papazoglou et al. (2007); Erl (2007), are discussed in details next.

- **Service contract**

Service contracts are a focal point of service design because they are central to everything that services do. A service contract establishes the terms of engagement, providing technical constraints and requirements, as well as semantic information, which the service owner wishes to make public.

This contract consists of a group of service description documents, each one describing a part of the service. For example, a Web service contract comprises the following documents: WSDL definition, XML schema definition, WS-Policy description and a Service Level Agreement (non-technical).

The goal of contracts in a service-oriented environment is to enable services with a meaningful level of natural interoperability and allow the purpose and capabilities of services to be more easily and intuitively understood.

- **Service coupling**

Software coupling represents the relationship between two pieces of software (programs, components, and so on.). A level of coupling is comparable to a level of dependency. This principle emphasizes the reduction of coupling between the parts of a service-oriented solution.

The loose coupling is advocated between a service and its consumers, as well as the service contract and its implementation. The goal is to work towards a state where service contracts increase independence from their implementation, and services are increasingly independent from each other.

This promotes an environment with services and consumers evolved over time with minimal impact on each other.
2.2. SERVICE-ORIENTED COMPUTING (SOC)

- **Service abstraction**
  Abstraction aims at hiding some information about a program. Examples of its application are the “black box” concept, Application Program Interface (API)s, and middleware. The principle of service abstraction attempts at achieving the right balance of hiding information in a service-oriented environment.

  The other principles emphasize the need to publish more information in the service contract. Moreover, the goal of this principle is to keep the quantity and details of the contract content concise and balanced. Further, it prevents unnecessary access to additional service details.

- **Service reusability**
  One of the more fundamental principles in achieving goals of SOC is reusability. Software reuse is the process of creating software systems from existing software rather than building them from scratch (Krueger, 1992). Designing software to be reusable increases the Return On Investment (ROI), and introduces changes to the traditional application delivery lifecycle.

  The service reusability principle has two main goals. Firstly, it allows the service logic to be repeatedly leveraged over time, and also it allows to achieve an increasingly high return on the initial investment of delivering the service.

  Secondly, it increases business agility on the organizational level by enabling the rapid fulfillment of future business automation requirements through wide-scale service composition.

- **Service autonomy**
  Autonomy represents the ability to self-govern. An autonomous piece of software has the freedom and control to make its own decisions without external involvement. Autonomy requires a more isolated program implementation in order to increase more the independence among parts of the software.

  Services are autonomous, and the goal of the service autonomy is to increase the amount of control a service has over its runtime environment.

- **Service statelessness**
  State refers to the general condition of something. A software can have different states because of its involvement in a runtime activity. Each state is represented
and described by data, and the state management represents the processing of this information.

Services minimize resource consumption by deferring the state management only when necessary. The goals of the service statelessness are to increase service scalability and to improve the potential for service reuse.

• **Service discoverability**

Discoverability refers to the action of something to be effectively searched for and located. Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted. Discoverability information in a service-oriented environment is a combination of the content in a service contract and meta data in the corresponding registry record.

The goal of this principle is to express the purpose and capabilities of the service, thus, they can be interpreted by humans and software programs.

• **Service composability**

Composition is the ability to break a problem in small pieces to connect them, solving the problem proposed. Examples of software composition are the assembly of program libraries, object hierarchy and so on.

Service also are composition participants, and in nature, are composable. All the goals of service reusability are applied to service composability, because service composition often turns out to be a form of service reuse.

### 2.2.4 The Service-Oriented Development Process

Gu and Lago (2007) claim that there is no consensus for a service life-cycle model in the literature. They evaluated a number of proposals and pointed to a well-defined sequence of steps divided into three phases: design time, runtime and change time.

Design time refers to the life cycle of a service before it is available for use. During the runtime phase, services are put into production and the implementations start to work. The change time focuses on the life cycle of a service when adjustments have to be made when business requirements change.

As showed in Figure 2.4, the proposed model consists of three rows and three columns. The rows are partitioned by the stakeholders: service provider, service broker and service consumer (application provider), and each row comprises service life cycle activities associated with the stakeholders.
The columns are partitioned by design time, runtime and change time. Each column comprises service life cycle activities across the stakeholders. Each of these activities is linked if the output of an activity is the input of another one. This is an iterative and incremental process. The horizontal links imply the interactions within a single stakeholder; the vertical links imply the interactions across the stakeholders.

The service identification, focus of this dissertation, is an analysis and modeling activity (Fareghzadeh, 2008), because it is an activity concerned to analyze existing software development artifacts to identify services, as a part of the service modeling. Therefore, based on the Gu and Lago (2007) life-cycle model, this activity is performed at the design time phase through service providers (during the service design step), that decide which are the services delivered for service consumers. This dissertation has a special focus on the design time activities, specifically service identification (one of the service design tasks).

### 2.2.5 The Benefits

**SOC** help companies in different ways, depending on their goals and the way in which **SOC** is applied (Erl, 2005). Erl (2007) argues that “the visio behind service-oriented computing is extremely ambitious and therefore also veru attractive to any organization interested in truly improving the effectiveness of its IT enterprise”. This vision is mainly based on a set of benefits for the company that successfully adopts service-orientation. A list of these benefits (Erl, 2005, 2007) is described next:

- **Increased business and technology domain alignment**

  **SOC** introduces a design paradigm that promotes abstraction in many levels. One of the most effective means, which functional abstraction is applied, is the establishment of service layers that accurately encapsulate and represent business models.

  This is accomplished by incorporating a structured analysis and modeling process, which requires the involvement of business experts in the definition of service candidates. The resulting service designs are capable to aligning automation technology with business intelligence on an unprecedented level.

  In addition, the fact that services are designed to be intrinsically interoperable, directly facilitates business change. As business processes are augmented in response to various factors, services can be reconfigured into new compositions that reflect the changed business logic.
2.2. SERVICE-ORIENTED COMPUTING (SOC)

- **Cost savings**
  Two major sources for cost savings are the project costs and maintenance costs. Lower costs are achieved optimizing the use of resources needed for updating or modifying code, integrating modules and testing.
  
  In addition, due to the simplification of the application landscape, the streamlining of the code base and technology independence, future changes can be made more easily, reducing maintenance costs.

- **Agility**
  Agility refers to the efficiency with which an organization can respond to changes. Service-orientation results in the creation of services that are standardized and reusable and, therefore, agnostic to parent business processes and specific application environments.
  
  Services as reusable Information Technology (IT) assets can be repeatedly composed into different configurations. Moreover, the time to change or add business processes is reduced because the IT projects can be completed with less development effort.

- **Reuse and the resulting benefits**
  Firstly, services are not confined to the project for which they are developed, since they can be reused in other projects. Secondly, SOC contributes to runtime reuse, where services can be reused at runtime by linking them to other services into sub-systems.
  
  Finally, reuse of code can reduce testing effort and improves the quality of artifacts. Instead of developing different services with similar functionalities, only one service is developed and tested.

- **Independence from technology**
  Focus on technology can be a major problem in IT projects. Resources spent discussing technology could be better spent designing and optimizing business functionality. SOC helps to shift the attention from technological to business issues, providing a decoupling of the technology and business aspects. It enables choosing the “best of breed” of products, and the introduction of new technologies does not require a makeover of the business processes and services.
2.3 Service Identification

The service identification is one of the first activities in the modeling of a service-oriented solution (Inaganti and Behara, 2007; Kohlmann and Alt, 2007), and it seems neglected in current research work (Wang et al., 2005). It consists of determining candidate services to a service-oriented environment, taking only one or a set of artifacts as the starting point. The goal is to identify services keeping a business alignment, with the right service granularity.

Service granularity refers to the service size and the scope of functionality a service exposes. If the service is coarse-grained, the size of exchanged messages grows and sometimes might carry more data than needed, and requires more complex interfaces, representing more possibilities to be affected by change. On the other hand, fine-grained services can require an amount of exchanged messages more than acceptable (Erradi et al., 2007). The problem of identifying right-grained services has not been addressed in the literature, in terms of formal methodologies and tools that allow the designer to generate and evaluate alternative designs (Jain et al., 2004; Erradi et al., 2007).

Furthermore, service identification is not a trivial task, and the errors made during the identification can flow down through detailed design and implementation activities, the next steps of the service-oriented development process (Inaganti and Behara, 2007).

There are three approach types of service identification: top-down, bottom-up, and “meet-in-the-middle” (Inaganti and Behara, 2007; Fareghzadeh, 2008; Ricca and Marchetto, 2009). These approaches are basically related to the input artifact that will be used in the process.

The top-down approach considers “high-level” artifacts (requirements, business processes, feature models, and so on) to determine the candidate services in a service-oriented solution. A common strategy is the business process driven approach. Such an approach identifies services by decomposing the business processes into tasks (or sets of tasks) that provide a right-grained functionality and that might be reusable (Mani et al., 2008).

The bottom-up approach identifies services by utilizing the information systems in place. This approach enable architects to identify the redundancies at the source code and data across the enterprise applications portfolio. The redundancies can be potential service candidates, and at the data level, CRUD (Create, Read, Update and Delete) based operations would be probably appropriate (Inaganti and Behara, 2007). Finally, a “meet-in-the-middle” (also called hybrid) approach combines bottom-up and top-down strategies.
Besides the different approaches, there is a lack of common understanding of what services are and which goals must be achieved (Gu and Lago, 2010). Due to this, the existing approaches for service identification differ significantly one from another. Unfortunately, an unified method for identifying services has not yet been reached, instead, a variety of heterogeneous approaches (Erradi et al., 2007; Arsanjani et al., 2008; Kang et al., 2008; Lee et al., 2008) have been proposed.

Considering the heterogeneity of the service identification methods, Gu and Lago (2010) performed a systematic review and identified thirty studies addressing it, as well as the types of inputs, outputs and processes of them. They argue that the variety of input types, outputs and processes explains why practitioners often face difficulties to select a service identification method that fits the practitioners needs and copes with the available resources.

In addition, they made some observations about future research directions. Firstly, IT services that leverage business processes and underlying IT infrastructure require more attention, because in ten of the thirty approaches, business process is a common input for the services identification. Next, services for internal use and external consumption should be differentiated due to their different characteristics, i.e., IT services and business services have different needs, and the integration among them should pay a specific attention. Finally, non-functional requirements should be explicitly considered due to their importance through the entire service life cycle.

### 2.4 Chapter Summary

In this Chapter, we discussed about some of the main involved topics of this work: Software Product Lines, Service-Oriented Computing and Service Identification. There are some SPL development processes (Clements and Northrop, 2001; Pohl et al., 2005) in order to create SPLs and benefits of using them, such as: cost reduction, quality improvement, reduction of time-to-market, and so on.

Service-Oriented Computing is a discipline for developing software systems that follows a set of principles (Papazoglou et al., 2007): contract, loose coupling, abstraction, reusability, autonomy, statelessness, discoverability and composability. There are some development process (Gu and Lago, 2007) to create service-oriented systems as well as benefits of using SOC technique (Erl, 2005, 2007), such as: increased business and technology domain alignment, cost savings, agility, and so on.

Service Identification is one SOC issue. It is not a trivial task, and the errors made
during the identification can flow down through detailed design and implementation activities. An unified method for identifying services has not yet been reached, and heterogeneous approaches have been proposed.

Next Chapter presents a mapping study concerned to investigate the combination of SPL and SOC areas, called Service-Oriented Product Line Engineering (SOPL E). The goal is to understand the motivation of combining activities of both areas, indentify drawbacks, benefits and research gaps in this emergind field.
Figure 2.4 Service-Oriented Lifecycle (Gu and Lago, 2007).
Putting SPL and SOC together have begun to receive attention by researchers and practitioners, and this combination is called SOPLE. Both areas share business goals in common, and they can address issues of each other. The goal of this mapping is to understand the motivation of combining techniques of both areas as well as identify drawbacks, benefits and research gaps in this emerged field.

This Chapter is organized as follows: Section 3.1 presents the methodology to perform this mapping study and Section 3.2 discusses the rationale of the research questions. Section 3.3 describes the criteria for inclusion and exclusion of studies. Section 3.4 defines the search strategy to find out the relevant SOPLE papers (primary studies), and Section 3.5 presents the results of this search. Section 3.6 describes the schemes to classify the papers in terms of facets. Section 3.7 presents the mapping study results, and Section 3.8 discusses some findings of this activity. Finally, Section 3.9 describes the threats to validity and Section 3.10 presents a summary of this Chapter.

### 3.1 Methodology

This Chapter presents a systematic mapping study in the SOPLE field in order to select and analyze the relevant primary studies in this area, integrating the outcomes, trying to generalize them, and identifying existing issues. This mapping is based on Kitchenham and Charters (2007) and Petersen et al. (2007) guidelines.

Figure 3.1 presents steps and outcomes for the systematic mapping process. The first
step defines the research questions of the mapping study. Secondly, the primary studies are identified through a search strategy. The third activity excludes primary studies that are not relevant to answer the research questions, based on inclusion and exclusion criteria. Next, the goal is to build a classification scheme, helping to categorize the extracted data. Finally, the extracted data is synthetized and reported for the stakeholders. A pilot of the research protocol was performed aiming to find mistakes in the mapping process. The resulting activities will be discussed in details through next Sections.

![Systematic Mapping Process by Petersen et al. (2007).](image)

### 3.2 Research Questions

The Easterbrook (2007) work - providing guidelines to create research questions - helped to create the mapping study research questions. These questions and their rationales are described as following:

- **RQ1. What is the intensity of the research activity in the SOPLE area?**
  
  Rationale: The graphical representation of the amount of papers per year indicates the existence of an increasing or a decreasing in the research activity. In addition, this question aims to identify the countries, journals and conferences of the selected papers.

- **RQ2. What are the goals of integrating SOC and SPL practices?**
  
  Rationale: This question analyzes the motivation to integrate SOC and SPL areas, and what are the provided benefits and possible shortcomings of this integration.

- **RQ3. What are the applied research methods in the SOPLE area?**
  
  Rationale: The goal is to identify the applied research methods (e.g. experimental study, case study, survey) and how they are evolving over the years.
3.3 INCLUSION AND EXCLUSION CRITERIA

- **RQ4. What are the domains and application types addressed for the SOPLE area? Which proposals are applied on enterprises?**

  Rationale: SOC and Service-Oriented Architecture (SOA) can be applied in different domains (e.g. medical domain, financial domain) and contexts (e.g. web system, embedded system). In addition, this question identifies studies that are developed in academic and enterprise environments.

- **RQ5. Which SOC topics are influenced in the SOPLE research?**

  Rationale: The goal is to identify the SOC topics (e.g. service management, service design, service discovery) that are used to solve SPL issues, i.e., that are applied in the SPL engineering.

- **RQ6. Which SPL topics are influenced in the SOPLE research?**

  Rationale: This question is the opposite of the prior question. The goal is to identify the SPL topics (e.g. variability managing, product derivation) that are used to solve SOC issues, i.e., that are applied in the SOC development lifecycle.

- **RQ7. What are the types of proposals in the SOPLE area?**

  Rationale: Each study has a different goal. For instance, there are papers providing tools to automatize processes, or metrics measuring some aspects, and so on. This question aims to identify which are the types of proposals in order to achieve the intended goals. These proposal types can be: tool, process, method, metric, model, and so on.

---

3.3 Inclusion and exclusion criteria

The selection criteria are defined aiming to have a set of consistent definitions about the inclusion and exclusion of studies in the mapping. This procedure also reduce the likelihood of bias, and the criteria can be refined during the search process. The criteria are described as follows:

- **Inclusion Criteria:** all studies published on journals and conferences which address SOPLE; only papers written in the english language; studies that proposes the integration of SPL and SOC areas. In general, this integration comprises SPL papers that address SOC techniques as well as SOC papers that address SPL techniques in order to fill the gaps for the context defined in the paper (e.g. a SOC paper using feature modeling to handle the variability in service-oriented systems).
• **Exclusion Criteria:** short papers and grey literature such as white papers, books, and so on; all studies published on journals and conferences, which address SOPLE; studies that do not consider a service as a black box, self-contained, stateless, loosely coupled, interoperable and reusable entity with well-defined contracts; studies discussing only SOC issues without mention SPL concepts or studies discussing SPL issues without considering SOC concepts; studies focusing only on Web Services without addressing SOC aspects and principles; a duplicate study must be excluded. If a study has been published in more than one conference, workshop or journal, we will consider the paper containing more useful information according to the mapping research questions.

### 3.4 Search strategy

This mapping study considered two types of search: automatic and manual. The sources of evidence regarded in the automatic search are the electronic databases. Brereton *et al.* (2007) argue that we need to search many different electronic sources, since no single source finds all the primary studies.

Thus, the electronic databases included in this study are: IEEE XPlore, ACM Digital Library, ScienceDirect, Scopus and Engineering Village. The search string applied in each database present some differences, however, in essence, the search string used is showed in Table 3.1.

```
("service oriented" OR "service-oriented" OR "SOA" OR "SOC" OR "SPL" OR "service design" OR "service based" OR "service-based" OR "service engineering" OR "service analysis" OR "service implementation" OR "service-orientation" OR "service orientation" OR "service variability" OR "service family" OR "service customization" OR "service composition" OR "service contract" OR "service grid" OR "service delivery" OR "service orchestration" OR "service development" OR "service systems" OR "service identification")
AND
("SPL" OR "SPL" OR "product line" OR "product family" OR "domain engineering" OR "application engineering" OR "variability" OR "feature analysis" OR "feature modelling" OR "feature modeling" OR "core asset")
```

**Table 3.1** Search String.

These search terms are based on the research questions of this study. They were tested and calibrated though a pilot in the search engines, and the results are described in the Table 3.2.
3.5. SCREENING OF PAPERS

<table>
<thead>
<tr>
<th>Electronic database</th>
<th>Amount of papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM Digital Library</td>
<td>449</td>
</tr>
<tr>
<td>IEEE Xplore</td>
<td>282</td>
</tr>
<tr>
<td>Scopus</td>
<td>1153</td>
</tr>
<tr>
<td>ScienceDirect</td>
<td>264</td>
</tr>
<tr>
<td>Engineering Village</td>
<td>444</td>
</tr>
</tbody>
</table>

Table 3.2 Automatic Search - Results.

The electronic databases might index all relevant papers for this study, however, there can be shortcomings in this automatic search process. For instance, Brereton et al. (2007) claim that indexing database are inadequate to support complex boolean searches. Kitchenham and Charters (2007) argues that automated searches find more studies than manual restricted searches, but they may be of poor quality.

Hence, a manual search was performed. This activity consists of searching within relevant international journals and conferences of the SPL and SOC areas. The list of journals and conferences used in the manual search can be seen in Tables 3.6 and 3.4. The manual search returned 58 candidate studies.

<table>
<thead>
<tr>
<th>Journals</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM Transactions on Software Engineering and Methodology</td>
</tr>
<tr>
<td>Journal of Systems and Software</td>
</tr>
<tr>
<td>Communications of the ACM</td>
</tr>
<tr>
<td>IEEE Software</td>
</tr>
<tr>
<td>IEEE Transactions on Services Computing</td>
</tr>
<tr>
<td>IEEE Transactions on Software Engineering</td>
</tr>
<tr>
<td>International Journal of Services and Standards</td>
</tr>
<tr>
<td>International Journal of Services Technology and Management</td>
</tr>
<tr>
<td>International Journal of Web Services Research</td>
</tr>
<tr>
<td>Service Oriented Computing and Applications</td>
</tr>
</tbody>
</table>

Table 3.3 Manual Search - Journals.

3.5 Screening of Papers

The screening process is composed by three filters in order to select the relevant primary studies. Figure 3.2 shows what was considered in each filter and the amount of studies remaining after applying each one. The screening process was performed by three
3.5. SCREENING OF PAPERS

<table>
<thead>
<tr>
<th>Conferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM SAC Track on Service Oriented Architecture and Programming (SOAP)</td>
</tr>
<tr>
<td>IEEE International Conference on Service-Oriented Computing and Applications (SOCA)</td>
</tr>
<tr>
<td>IEEE International Conference on Services Computing (SCC)</td>
</tr>
<tr>
<td>IEEE International Conference on Web Services (ICWS)</td>
</tr>
<tr>
<td>International Conference on Service Oriented Computing (ICSOC)</td>
</tr>
<tr>
<td>International Conference on Software Engineering (ICSE)</td>
</tr>
<tr>
<td>International Conference on Software Reuse (ICSR)</td>
</tr>
<tr>
<td>Software Product Line Conference (SPLC)</td>
</tr>
<tr>
<td>Software Product-Family Engineering (PFE)</td>
</tr>
<tr>
<td>World Congress on Services (SERVICES)</td>
</tr>
<tr>
<td>Workshop on Service-Oriented Architectures and Software Product Lines (SOAPL)</td>
</tr>
</tbody>
</table>

Table 3.4 Manual Search - Conferences.

researchers.

The first filter selected the relevant papers based on inclusion criteria regarding title and abstract of the studies. After excluding duplicated studies found in the automatic and manual search, 114 papers have remained to be evaluated (second filter). The third filter applied the exclusion criteria in all resulting papers of the second filter, analyzing the introduction, conclusion and other parts of the text. At the end, 43 studies remained to participate of the data extraction, analysis and synthesis.

Ten randomly selected studies were analyzed by all involved researchers in order to calibrate the understanding of the aspects (research questions, inclusion/exclusion criteria, classification schemes, and so on) defined in the protocol - a pilot study. After that, each
3.6 Classification scheme

We classified the studies in a set of facets, as described by Petersen et al. (2007). Hence, six facets were created, and they are explained as following.

3.6.1 Research method

This mapping study used an existing classification of research approaches by Wieringa et al. (2005), which classifies the research methods as: philosophical papers, experience papers, solution proposal, evaluation research, opinion paper, and validation research as described next.

- **Validation Research.** Techniques investigated are novel and have not yet been implemented in practice. Techniques used are for example experiments, i.e., work done in the lab.

- **Evaluation Research.** Techniques are implemented in practice and an evaluation of the technique is conducted. That means, it is shown how the technique is implemented in practice (solution implementation) and what are the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation). This also includes to identify problems in industry.

- **Solution Proposal.** A solution for a problem is proposed, the solution can be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution is shown by a small example or a good line of argumentation.

- **Philosophical Papers.** These papers sketch a new way of looking at existing things by structuring the field in form of a taxonomy or conceptual framework.

- **Opinion Papers.** These papers express the personal opinion of somebody whether a certain technique is good or bad, or how things should been done. They do not rely on related work and research methodologies.
3.6. CLASSIFICATION SCHEME

- **Experience Papers.** These papers express the personal opinion of somebody whether a certain technique is good or bad, or how things should been done. They do not rely on related work and research methodologies.

### 3.6.2 SOC Topics

These facet was created based on the keywording of abstracts process proposed by ?). This process is performed in two phases: firstly, the reviewers read the abstracts and look for keywords or concepts that reflect the paper contribution. Next, the set of keywords are combined to develop an understanding about the contribution of the research. The idea of this scheme is to verify which **SOC** topics were influenced in the studies.

- **Business Process Management (BPM).** BPM is a management approach in order to align the aspects of an organization with the wants and needs of its clients.

- **Quality of Service (QoS).** QoS in this context refers to the quality aspects offered by the developed services in a service-oriented environment.

- **Service management.** It refers to some aspects such as service composition, negotiation, service level agreements, and compliance.

- **Service design.** Specification and design of services are treated in the study.

- **Service customization and adaptation.** The study mentions some aspect of service customization and adaptation, variability techniques, and so on.

- **Infrastructure.** The study mentions aspects related to the hardware infrastructure needed for service-oriented systems.

- **Testing and Validation.** The study incorporates service testing and validation techniques.

- **Governance.** Governance aspects in service-oriented computing.

- **Web Services.** The study mentions the Web Services technology.

- **Grid Services.** The study mentions about service development over grid computing infrastructures.

- **Service Discovery and selection.** Techniques to discover and select service in a service-oriented environment.
3.6.3 SPL Topics

In the same process of the previous facet, this scheme was created based on the classification of the main conferences in the SPL using the keywording of abstracts. It identifies which SPL topics were influenced in the studies.

- **Variability resolution.** The study relates some way for variability resolution and it is closely related to product derivation, that is, how to handle the variability in the product derivation process.
- **Variability management.** The study involves some aspect of variability management.
- **Product derivation.** Techniques for product derivation.
- **Feature analysis.** Feature analysis techniques such as feature modeling, feature extraction, and so on.
- **Configuration.** How to model the configuration of the products.
- **Scoping.** Techniques and activities for the product line scoping phase.
- **Dynamic product line.** Product lines that binds the variability at runtime.
- **Product line architecture.** Activities to design a product line.

3.6.4 Proposal type

This facet was created using the keywording strategy. The idea is to classify the papers according to their proposal type, such as: process, tool, method, model, metric and discussion.

- **Process.** A process is a set of activities to develop a service-oriented product line. This process can include activities, roles, artifacts, and so on.
- **Tool.** A software tooling support is developed in order to support different aspects of problem under study.
- **Method.** A method is proposed which describes rules, guidelines and a sequence of steps to perform some activity to solve a problem.
- **Model.** A model or metamodel to represent some aspect of SOPLE artifacts.
3.7. RESULTS

- **Metric.** This category proposes metrics to evaluate different SOPLE aspects.
- **Discussion.** In this category, comparison of different characteristics of existing approaches or area overview is considered.

### 3.6.5 Application type

This facet also was developed using the keywording strategy, and identifies which application types the SOPLE techniques are applied.

- **Process-oriented system.** Process-oriented systems aims at managing business process of an enterprise.
- **Embedded system.** An embedded system is a computer system designed to do one or a few dedicated and/or specific functions often with real-time computing constraints.
- **Information system.** Information systems are implemented within an organization for the purpose of improving the effectiveness and efficiency of that organization.
- **Grid system.** In grid computing, software is run on a distributed computer consisting of multiple, general purpose computers connected by a network.
- **Context-aware system.** Context-aware computing refers to a general class of mobile systems that can sense their physical environment, and adapt their behavior accordingly.

### 3.7 Results

**RQ1. What is the intensity of the research activity in the SOPLE area?**

A total of 43 SOPLE papers were published between 2005 and 2010. It shows that Service-Oriented Product Line Engineering is a relatively new area, and according to the Figure 3.3, the research activity has grown over the years. In 2005 and 2006, there is only one publication addressing SOPLE, and this number has risen to six in 2007 and to eleven in 2010. The amount of papers per year did not decrease. Hence, SOPLE issues are are paying more attention by researchers over the years.

Regarding the countries involved in the SOPLE research, fifteen countries have published studies covered in this mapping, in four continents (America, Europe, Asia and
3.7. RESULTS

Figure 3.3 Intensity of the research activity in SOPLE area.

Oceania). Korea and Germany were the countries that most have published in the SOPLE area (seven papers, or 16.3% for each one). The amount of papers has a homogeneous distribution among the countries, with a standard deviation of 1.81. Table 3.5 summarizes the ranking of these countries.

This mapping covered 2 journals, 13 conferences and 6 workshops (see Table 3.6). Most of the published studies are conference papers (15, 35% of the total) against 25 (58% of the total) conference papers and 3 (7% of the total) journal papers. Regarding the top journal, top conference and top workshop which publish SOPLE studies, the Journal of Systems and Software (2 papers), International Software Product Line Conference (5 papers) and Workshop on Service-Oriented Architectures and Software Product Lines (10 papers) were the top ones, respectively.

Figure 3.4 shows the evolution in the amount of papers per year in journals, conferences and workshops. Journal papers were published from 2010, conference papers were not published in 2005, however they kept a growth between 2006 and 2009. Although workshop papers kept a growth between 2006 and 2008, they decreased in 2010.

RQ2. What are the goals of integrating SOC and SPL practices?

Analyzing the selected primary studies, there is a considerable number of papers investigating the integration of SPL and SOC concepts. Both techniques are being used to support each other, i.e., while SPL practices lead to a more systematic and planned reuse, a service-oriented architecture provides the flexibility required to develop adaptable and
3.7. RESULTS

<table>
<thead>
<tr>
<th>Countries</th>
<th>Amount of Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Korea</td>
<td>7</td>
</tr>
<tr>
<td>Germany</td>
<td>7</td>
</tr>
<tr>
<td>France</td>
<td>5</td>
</tr>
<tr>
<td>Brazil</td>
<td>4</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>4</td>
</tr>
<tr>
<td>Finland</td>
<td>3</td>
</tr>
<tr>
<td>China</td>
<td>3</td>
</tr>
<tr>
<td>Italy</td>
<td>3</td>
</tr>
<tr>
<td>United States</td>
<td>3</td>
</tr>
<tr>
<td>Spain</td>
<td>2</td>
</tr>
<tr>
<td>Canada</td>
<td>2</td>
</tr>
<tr>
<td>The Netherlands</td>
<td>2</td>
</tr>
<tr>
<td>Australia</td>
<td>2</td>
</tr>
<tr>
<td>Singapore</td>
<td>1</td>
</tr>
<tr>
<td>Ireland</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.5 Ranking of countries that have published about SOPLE.

configurable systems.

In this context, we could identify four main types of works combining SPL and SOC concepts as detailed next:

- Research works that take variability into account during the service-oriented development with the main intention of customizing services to specific customers or market segment needs, improving the service reusability, such as Díaz et al. (2003), van Gurp and Savolainen (2006), Chang and Kim (2007), Liu et al. (2007), Kang et al. (2008), Acher et al. (2008), Asadi et al. (2009), Bartholdt et al. (2008), Boffoli et al. (2008), Gimenes et al. (2008), Wang et al. (2008), Hesham Shokry (2008), Kim and Doh (2008), Boffoli et al. (2009), Istoan et al. (2009), Nunes et al. (2009), Park et al. (2009), Smith and Lewis (2009), Stollberg and Muth (2009), Sun et al. (2009), Ribeiro et al. (2010), Galster (2010), Kang and Baik (2010), Kattepur et al. (2010), Medeiros et al. (2010), Park et al. (2010), Sun et al. (2010) and Nguyen et al. (2011);

- Works that focus on the dynamic aspects of SOC, e.g. dynamic discoverability and binding of services, and uses service orientation as a way to develop dynamic software product lines that can be customized or configured during runtime, such as Lee et al. (2007), Mietzner and Leymann (2008), Kotonya et al. (2009), Lee
3.7. RESULTS

<table>
<thead>
<tr>
<th>Journals, Conferences and Workshops</th>
<th>Amount of Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workshop on Service-Oriented Architectures and Software Product Lines (SOAPL)</td>
<td>10</td>
</tr>
<tr>
<td>International Software Product Line Conference (SPLC)</td>
<td>5</td>
</tr>
<tr>
<td>International Conference on Convergence and Hybrid Information Technology (ICHIT)</td>
<td>3</td>
</tr>
<tr>
<td>International Conference on Software Reuse (ICSR)</td>
<td>3</td>
</tr>
<tr>
<td>IEEE Asia-Pacific Services Computing Conference (APSCC)</td>
<td>2</td>
</tr>
<tr>
<td>Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS)</td>
<td>2</td>
</tr>
<tr>
<td>European Conference on Software Architecture (ECSA)</td>
<td>2</td>
</tr>
<tr>
<td>International Conference on Web Services (ICWS)</td>
<td>2</td>
</tr>
<tr>
<td>Journal of Systems and Software (JSS)</td>
<td>2</td>
</tr>
<tr>
<td>ACM Symposium on Applied Computing (SAC)</td>
<td>1</td>
</tr>
<tr>
<td>International Conference on Computer and Information Science (ACIS)</td>
<td>1</td>
</tr>
<tr>
<td>International Conference on Computer and Information Technology (ICCT)</td>
<td>1</td>
</tr>
<tr>
<td>International Conference on e-Business Engineering (ICEBE)</td>
<td>1</td>
</tr>
<tr>
<td>International Conference on Grid and Cooperative Computing (GCC)</td>
<td>1</td>
</tr>
<tr>
<td>International Conference on Services Computing (SCC)</td>
<td>1</td>
</tr>
<tr>
<td>International Workshop on Dynamic Software Product Lines (DSPL)</td>
<td>1</td>
</tr>
<tr>
<td>International Workshop on Engineering Service-Oriented Applications</td>
<td>1</td>
</tr>
<tr>
<td>International Workshop on Feature-Oriented Software Development (FOSD)</td>
<td>1</td>
</tr>
<tr>
<td>International Workshop on Technologies for E-Services (TES)</td>
<td>1</td>
</tr>
<tr>
<td>International Workshop on Variability Modeling of Software-Intensive Systems (VaMoS)</td>
<td>1</td>
</tr>
<tr>
<td>Journal of IEEE Software</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.6 Papers per journal, conference and workshop.

et al. (2009), Mohabbati and Nima Kaviani (2009), Parra et al. (2009), Lee et al. (2010), Nguyen and Colman (2010), Yu et al. (2010) and Lee and Kotonya (2010);

- Studies that do not explicitly mention the purpose of integrating SPL and SOC concepts, but detail a comparison of the concepts presenting their similarities and differences, or discussing SPL, SOC and related concepts such as Ontologies, Process-Driven Development, and so on. These kind of studies are Raatikainen et al. (2007b), Raatikainen et al. (2007a), Günther and Berger (2008) and Bu-Qing et al. (2009); and

- Research works that uses SOC with the purpose of increasing interoperability with the purpose of integrating services developed by different companies, platforms or languages, such as Trujillo et al. (2007b).

Figure 3.5 summarizes these amount of papers according to the motivation of integrating SPL and SOC.

Most of the works (28 studies) focus on practices applied during the service-oriented development, and extend these practices to deal with variability, such as business process modeling (Boffoli et al., 2008). In addition, SPL practices are being used to plan and
3.7. RESULTS

Figure 3.4 Journal, conference and workshop papers per year.

![Figure 3.4](image1)

Figure 3.5 Motivation of integrating SOC and SPL.

![Figure 3.5](image2)

systematize the reuse of services to keep them useful for a long period of time or customize them.

On the other hand, SOC activities and characteristics of service orientation (e.g. the dynamism and interoperability of service-oriented architectures) are being used to create dynamic software product lines (10 studies). However, few papers are focusing on integration issues to compose services developed in different languages on any platform (1 study).

We could identify that some good practices of both SPL and SOC are being used together to support each other, e.g. the use of features models (such as in Díaz et al. (2003), Kang et al. (2008) and Kotonya et al. (2009)), process-oriented development (used in Boffoli et al. (2008) and Bu-Qing et al. (2009)), business process variability
3.7. RESULTS

(such as in Boffoli et al. (2008)) and service variability (used in Chang and Kim (2007), Díaz et al. (2003), Liu et al. (2007) and Kang et al. (2008)). In addition, we could also detect studies considering variability in non-functional requirements, such as Sun et al. (2009), and reusing existing components and services, i.e., the bottom-up approaches as described in Ribeiro et al. (2010).

Another point is the focus on BPM activities and adapting them to deal with variability, such as in Gimenes et al. (2008), Sun et al. (2010) and Boffoli et al. (2008). In this context, business processes are developed concerning variability and can be customized for specific customers according to their requirements. Another work is Kattepur et al. (2010), which analyzed the QoS of service compositions to ensure their Service Level Agreement (SLA).

We could also detect studies concerned about the integrity and consistency of dynamically composed services, such as Lee et al. (2007), Raatikainen et al. (2007b) and Lee et al. (2010). Since not all possible variants can be constructed correctly, it is important to verify service compositions composed dynamically to ensure that they have a valid configuration. We could also identify studies addressing the reconfiguration of services dynamically, such as Lee and Kotonya (2010).

RQ3. What are the applied research methods in the SOPLE area?

A considerable number of studies are solution proposals (21 studies, 49% of the total), in which a solution for a problem is proposed. This solution can be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution are shown by a small example or a good line of argumentation.

A great amount of evaluation research were performed (13 studies, 30% of the total), in which techniques are implemented in practice and an evaluation of the technique. In this sense, it shows how the technique is implemented in practice (solution implementation) and the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation). It also includes the identification of problems in industry. Figure 3.6 summarizes the intensity of papers in relation of their research method used to evaluate their proposals.

Empirical studies in the SOPLE area have increased over the years. Figure 3.7 shows this evolution, in which the number of papers performing evaluation research increased from one in 2008 to six in 2009. These numbers can indicate an increasing of research rigor of the area.
3.7. RESULTS

RQ4. What are the domains and application types addressed for the SOPLE area? Which proposals are applied on enterprises?

The results of this question intends to identify the domains (e.g. medical domain, financial domain) and application types (process-oriented system, embedded system, information system, grid system, and context-aware system) of the SOPLE studies.

The domain perspective were also identified in the studies, such as: Customer Relationship Management (CRM), health care system, automotive system, medical imaging, web store, business hotel, e-marketplace, house automation system, SCM (Supply Chain Management), selling system, virtual office, travel agency, product data management, and so on. As showed in Figure 3.8, the SOPLE application domains are heterogeneous, and a great number of papers did not consider any application domain.

In addition, the application type perspective (showed in Figure 3.9), information system was the application type applied in most studies (20 papers, 46.5% of the total). Other application types were also applied such as process-oriented system (5 studies, 11.6% of the total), embedded system (3 studies, 7% of the total), context-aware system (also 3 studies) and grid system (1 study, 2.3% of the total). However, the researchers could not define the application type of many studies (15 papers, 27.9% of the total).

The majority of the studies are applied in an enterprise environment (62.8%). This result can be related to characteristics of the SOC, and one of its benefits is the increased business domain alignment (Erl, 2005, 2007). Since services focusing on business value,
they link the gap between business and IT. An increased number of companies are turning to SOC in order to consolidate and repurpose legacy applications and combine them with new applications (Barros and Dumas, 2006). For instance, information systems today are not computational islands, but rather systems that need to communicate and interoperate over the Internet or corporate intranets (Sun et al., 2010).

Indeed, none of the studies have considered enterprise architecture aspects. Enterprise architectures align the strategies of enterprises with their business processes and their (business and IT) resources (Wegmann, 2003; Zachman, 1987). Table 3.7 shows the studies that addresses SOPLE applications in enterprise environments.

<table>
<thead>
<tr>
<th>Focus on enterprise</th>
<th>Amount of Papers</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>27</td>
<td>62.8%</td>
</tr>
<tr>
<td>No</td>
<td>14</td>
<td>32.6%</td>
</tr>
<tr>
<td>Not clear</td>
<td>2</td>
<td>4.7%</td>
</tr>
</tbody>
</table>

Table 3.7 Enterprise focus.

RQ5. Which SOC topics are influenced in the SOPLE research?

This question addresses the relationship between service-oriented topics and the software product line area. The identified SOC topics in the classification scheme were: BPM for service-oriented systems, quality of service, service management (includes specific topics, such as Service composition, negotiation, compliance, and so on), service design
activities (requirements and specification of services), service customization and adaptation, infrastructure (physical structure of service-oriented systems), testing and validation, governance and adoption, web services, grid services and service discovery/selection.

The most addressed topic, as showed in Figure 3.10 (one study can address more than one SOC topic), is service design (27 studies). Other topics also had a considerable number of publications, such as: service management (22 studies), web services (18 studies), service customization/adaptation (17 studies), quality of service (14 studies), service discovery/selection (12 studies) and BPM (9 studies). On the other hand, infrastructure (3 studies), testing and validation (2 studies) and grid services (2 studies) presented few evidence, and a governance/adoption study was not found.

**RQ6. Which SPL topics are influenced in the SOPLE research?**

This question addresses the relationship between software product line topics and the Service-Oriented Software Engineering area. The identified SPL topics in the classification scheme were: variability resolution (techniques to resolve variabilities), variability management, product derivation, feature analysis and modeling, configuration, scoping, dynamic product line and product Line architecture. One study can address more than one SPL topic.

The most addressed topic is variability management, as showed in Figure 3.11 (38 papers). Feature analysis (24 papers) is the most used technique to handle the variabilities. Variability resolution (17 papers) also had a considerable number of publications. On
RQ7. What are the applied proposal types in the SOPLE area?

The proposal types, as defined before, are: process, tool, method, model and metric (see Section 3.6). Method is the most addressed proposal type that was identified in this mapping, present in 44% of the papers (33 studies). Model and tool also were well-addressed proposal types, with 20 and 13 papers, respectively. Process (5 papers) and metric (4 papers) presented a few amount of publications. Figure 3.12 summarizes this information.

3.8 Analysis

This Section presents a synthesis of the collected evidence based on the mapping study research questions. Figure 3.13 synthesizes three analyzed aspects of this work: SOC topics, SPL topics and the relationship between each area with the respective proposal types. There are some regions in the chart that present a considerable research activity, and other area presenting an absence of studies.

For example, in the relationship between SOC topics and proposal types, the papers propose methods to address topics such as quality of service, service management, service...
3.8. ANALYSIS

Figure 3.10  SOC topics that are influenced by the SOPLE research.

design, service customization, web services and service discovery. For SOC topics, we can see a significant activity in the proposals of models. Moreover, proposals like metrics, tools and process present a poor research activity.

The most addressed SPL topics are variability resolution, variability management and feature analysis, as described previously. In addition, the most applied proposal types considering these SPL topics are models, methods and tools.

The evidence collected through this mapping study can provide some assumptions for the SOPLE area. Each assumption is described in details as follows:

• **SOPLE is not a different engineering approach.** It was perceived that the focus of the primary studies is to address specific issues in service-oriented engineering with product line activities or issues in product line development with service-oriented activities. It means the proposals incorporate specific activities of the other area in the development process. In short, there is not a new engineering approach that covers the whole software development life-cycle. The evidence shows SOPLE is not a “merging” of the two areas to create a different approach, instead, is the practice of incorporating some activities from one area to the another one.

The questions RQ2, RQ5 and RQ6 presents evidence for this assumption. In the second research question, the main motivation of integrating SPL and SOC areas addresses the variability in SOC. In this sense, SOC applications are using SPL activities in order to solve their variability handling issues. RQ5 and RQ6 presents the SPL and SOC topics that are influenced by SOPLE. The analysis of these two
3.8. ANALYSIS

Figure 3.11 SPL topics that are influenced by the SOPLE research.

Figure 3.12 Proposal types.

questions points the studies are addressing specific SPL and SOC activities, and they do not address a whole new engineering approach.

• **SOPLE is an emerging area.** The SOPLE emergence can be justified through the increased number of publications over the years. The RQ1 analysis provides evidence of this growth.

• **SOPLE is moving to a mature research field.** The RQ1 results shows a decreasing number of workshop papers and an increasing number of journal and conference papers. The assumption is that papers published in workshops have not the same level of maturity compared to journal and conference papers, i.e., journals and conferences have a greater rigor for publishing than workshops - in terms of research methos, evaluation of the proposal in a real-world scenario, and so on.
3.9. THREATS TO VALIDITY

Figure 3.13 SOC topics x SPL topics x proposal types.

- **SOPLE studies needs to improve the research rigor.** The RQ3 evidence shows a great number of publications applying small examples to evaluate their proposals, and some papers have not any type of evaluation. A research rigor improvement can be achieved, for example, through systematic methods for evaluation, such as: case study, experiments, and so on.

3.9 Threats to Validity

There are some threats to the validity of this mapping study, as well as the mitigation strategies for each threat.

- **Research questions.** The defined questions might not have covered all aspects of the SOPLE area. It was mitigated through discussions among the authors and external experts in the area in order to calibrate the questions. Regarding these discussions, we attempted to deeply address the most common issues in the field.

- **Publication bias.** Regarding the publication bias, we cannot guarantee that all relevant primary studies were selected. It is possible that some relevant studies
were not chosen during the search process. In order to mitigate it, we performed the automatic search, and complemented it performing manual search to collect all primary studies in this field.

- **Classification of the studies.** During the data extraction process, the studies were classified based on our judgment. In order to mitigate this threat, the classification process was performed using peer review.

- **Contribution type classification scheme.** The contribution type classification scheme can present threats to validity. The primary studies used the terms: technique, Modeling, method, technology, discussion and metric. However, an absence of general usage rules for these terms was observed.

### 3.10 Chapter Summary

This Chapter presented a systematic mapping study for the SOPLE field, based on Kitchenham and Charters (2007) and Petersen et al. (2007) guidelines, in order to select and analyze the relevant primary studies in this area. Seven research questions were created and provided some evidence about this field, such as: SOPLE is not a different engineering approach (instead, is a combination of activities from the two areas), SOPLE is an emerging area (its amount of publications is growing), SOPLE is moving to a mature research field (the amount of conference and journal papers is growing) and SOPLE studies needs to improve their research rigor.

Next Chapter presents, in details, four activities of the research design: literature review, criteria evaluation, methods comparison and methods classification. In addition, the involved research methods (systematic review and survey with expert opinion) are described in details.
This chapter addresses four activities of the research design that were defined to provide recommendations of the most suitable methods given the SPL scenarios. These activities, explained in the next sections, are: literature review (4.1), criteria evaluation (4.2) methods comparison (4.3), and methods classification (4.4).

4.1 Literature Review

The literature review is concerned to identify the existing service identification methods as well as the available criteria to compare and classify the methods. This activity is based on the Gu and Lago (2010) systematic review, and it is complemented with other sources of evidence, such as: manual searching (or hand searching) regarding some key conferences and the “snowballing” (Petticrew and Roberts, 2006).

Gu and Lago (2010) systematic review, the first source of evidence, presents a holistic view of the existing service identification methods. They examined 237 studies, and 30 of them were selected as primary studies. They provide a comparison of the methods regarding a set of aspects, as follows (The comparison table containing the collected service identification methods in the review and these aspects can be seen in Appendix A):

- **Input type**: artifact used as starting point of the method (e.g. business process, source code, use case, and so on).

- **Output format**: type of the identified service (e.g. business service, data service, and so on).
4.1. LITERATURE REVIEW

- **Output type:** the way to describe a service (e.g. informal service specification, service model, service implementation, and so on).

- **Strategy:** approach or plan for identifying services (e.g. business process decomposition, goal driven, component-based, and so on).

- **Technique:** technical procedures or actions taken to accomplish some defined tasks in a method (e.g. algorithm, guidelines, analysis, and so on).

- **Validation:** the form of validation (e.g. examples, case study, experiment, and so on).

The second source, manual search, consists of searching studies in specific conferences and journals determined by the researchers involved in this work, and does not have any support of automatic indexing system. This search complemented the systematic review performing a screening of papers in some Service-Oriented Computing and Software Product Lines conferences.

The goal was to cover the absence of a manual search in the Gu and Lago (2010) review, recovering methods until 2011. The target conferences were: SPLC (Software Product Line Conference), VaMoS (International Workshop on Variability Modelling of Software-intensive Systems), ICSR (International Conference on Software Reuse) and IEEE SCC (International Conference on Service Computing).

The search considered papers that address service identification methods, and the first filter, that analyzes the paper title and abstract, returned 13 candidate primary studies. However, the full reading of the papers excluded some papers and the remaining ones already were identified in the systematic review. Therefore, at the end of the manual search, there were not new primary studies.

The “snowballing” technique was used as the last source of evidence. It consisted of using the retrieved studies to identify other studies. In some papers of the systematic review, there were some informal reviews (not performed in a systematic way), and they provided some candidate primary studies. The reviews are: Jain et al. (2004), Klose et al. (2007), Boerner and Goeken (2009) and Kohlborn et al. (2009). These informal reviews, at the end, provided 2 new primary studies for this research.

In summary, 32 papers were identified. However, two papers of the systematic review were excluded because one paper was not found, and another paper did not propose a service identification method to be analyzed for this research. Therefore, 30 papers were analyzed in this work: Jain et al. (2004), Zhang and Yang (2004), Chen et al.
4.1. LITERATURE REVIEW

In addition to the methods, a set of criteria used for comparison and classification were identified. Basically, four studies reported the criteria: Klose et al. (2007), Boerner and Goeken (2009), Kim and Doh (2009) and Gu and Lago (2010). These criteria will be discussed in Section 4.2.

4.1.1 SPL Scenarios

Another task performed during the literature review was the definition of SPL scenarios. This definition meets the research question RQ1 (Chapter 1, Section 1.4), where it claims the most suitable service identification methods given some SPL scenarios. The scenarios are based on three SPL adoption approaches, defined in Krueger (2002): proactive, reactive and extractive. These approaches are described as follows.

The proactive approach to software product lines is like the waterfall approach to traditional software engineering (Cheatham and Crenshaw, 1991). Software development teams analyze, architect, design, and implement all product variations on the foreseeable horizon up front. Once the production line has been implemented, all that remains is to create product instances as needed. This approach might suit organizations that can predict their product line requirements well into the future and that have the time and resources for a long waterfall development cycle (Figure 4.1).

The high level tasks in the proactive approach are as follows:

1. Perform scoping to identify the variations to be supported in the product line;
2. Model the product line architecture to support all products in the production line;
3. Design the common and variant parts of the system; and
4. Finally, implement the common and variant parts of the system.

With the reactive approach, the organization incrementally grows their software production line when the demand arises for new products or new requirements on existing
products. This approach works in situations where you cannot predict the requirements for product variations well in advance or where organizations must maintain aggressive production schedules with few additional resources during the transition to a product line approach.

This incremental approach offers a quicker and less expensive transition into software mass customization. The reactive approach is like the spiral (Boehm, 1986) or extreme programming approach (Succi et al., 2001) to traditional software engineering. Software development teams analyze, architect, design, and implement one or several product variations on each development spiral. This approach showed in Figure 4.2 is appropriate when the requirements for new products in the production line are somewhat unpredictable.

The high level tasks for incrementally adding a new product using the reactive approach are:

1. Characterize the requirements for the new product relative to what is currently supported in the product line;

2. It is possible that the new product is currently within the scope of the current product line. If so, skip to step 4;
3. If the new product is not in scope, perform the “delta engineering” to the production line on any or all of the declarations, automata, common software, and definitions in order to extend the scope to include the new requirements; and

4. Create the product definition for the new product by selecting values for each of the feature declaration parameters.

![Diagram of Reactive Approach](image)

**Figure 4.2** Reactive approach (Krueger, 2002).

The *extractive approach* reuses one or more existing software products for the product line’s initial baseline. With the extractive approach, the organization capitalizes on existing custom software systems by extracting the common and varying source code into a single production line. After the production line has been populated, product instances are created.

The feature declarations, product definitions, and automata are created as the commonality and variation is identified during the extraction (Figure 4.3). This approach is very effective for an organization that wants to quickly transition from conventional to software product line engineering. The high level tasks for the extractive approach are as follows:

1. Identify commonality and variation in the existing systems;
2. Sub-steps:
   - create a single copy of the common software;
   - create feature declarations that model the scope of variation among the existing systems;
   - encapsulate variation points into automata;
   - program the automata logic to map declaration parameter values to variant selections in the automata; and
   - create the product definitions for the desired product instances by selecting values for each of the feature definition parameters.

![Extractive approach (Krueger, 2002).](image)

Based on these approaches and their relative assets, three SPL scenarios were identified: top-down, bottom-up and hybrid. They are described next:

- **Top-down scenario:** the first step of the proactive approach is to perform domain analysis and scoping (Clements and Northrop, 2001) to identify the variations to be supported in the product line, and the reactive approach initially characterizes the requirements for the new product relative to what is currently supported in the product line.

In general, these activities produce only “high-level” artifacts, that is, they do not consider “low-level” artifacts (such as source code) before the SPL implementation. Therefore, the available assets to identify services in the top-down scenario are: feature model, requirements, use case and business process.
4.2. CRITERIA EVALUATION

- **Bottom-up scenario:** The bottom-up scenario considers the extractive adoption model, because the used assets in the initial tasks to build Software Product Lines are extracted from the existing/legacy systems (source code). Thus, the available asset to identify services is the source code.

- **Hybrid scenario:** The hybrid scenario considers the two previous scenarios, consequently the two adoption models. Therefore, the available assets are “high and low level”, such as: feature model, requirements, use case, business process and the source code.

## 4.2 Criteria Evaluation

This activity involves to select the most significant criteria to perform a comparison and classification of the Service Identification Method (SIM)s. One of the literature review goals was to find out studies that define and use criteria to compare the service identification methods. The available criteria from literature were found in Klose et al. (2007), Boerner and Goeken (2009), Kim and Doh (2009) and Gu and Lago (2010).

The evaluation consists of three sub-activities: exclusion of duplicated criteria, merging of specific criteria, and a survey asking about the most significant ones.

Firstly, the duplicated criteria from the literature were filtered. Thereafter, criteria that were considered too specific were grouped in a new criterion, in a higher level of abstraction. For instance, vendor dependency and customer satisfaction (Boerner and Goeken, 2009) were grouped in the economic aspects criterion. The resulting list is composed of:

- **Service granularity:** granularity of the identified service (e.g. coarse, middle, fine, coarse to fine, and so on).

- **Understanding of services:** what the approach considers about what is a service (e.g. business process oriented, feature oriented, as module, and so on).

- **Direction of analysis:** three types: top-down, bottom-up and hybrid. The top-down analysis involves domain analysis, business process analysis or other forward method as starting point. A bottom-up analysis investigates code wrapping or refactoring strategies as starting point. The hybrid analysis is a mix of the two previous approaches.
4.2. CRITERIA EVALUATION

- **Input artifacts**: the artifact used to identify the services (e.g. feature model, use cases, business processes, and so on).

- **Output artifacts**: the output format of the identified services (e.g. service model, service implementation, and so on). Similar to output type (Gu and Lago, 2010).

- **Techniques/activities**: technical procedures or actions taken to accomplish certain defined tasks in a method (e.g. algorithm, guidelines, analysis, and so on).

- **Orchestration vs. choreography**: the way of managing the service environment. Orchestration implies a central instance that coordinates all activities of a process. The result of orchestrated services can itself be described as a service. Choreography means that services are called by other services and there is not one steering unit (Boerner and Goeken, 2009).

- **Research method**: the way to validate the identification approach (e.g. case study, example, and so on).

- **Research rigor**: the formalism of the validation (e.g. only implicit, special focus, mentioned, and so on).

- **Business/economic aspects**: business or economic aspects considered by the identification method (e.g. legal compliance, value creation, consideration of strategic aspects, and so on).

- **Industry sector**: the context where the method is applied (e.g. financial services, production, and so on).

Finally, a survey was applied to find out the most significant criteria to analyze the methods given some specific SPL scenarios. Next sub-section (4.2.1) will explain the details of the survey.

### 4.2.1 Survey

Survey is a comprehensive research method for collecting informations to describe, compare or explain knowledge, attitudes and behavior (Fink, 1995). The purpose of this method is to produce quantitative descriptions, by asking questions where their answers constitute the data to be analyzed (Fowler, 2002).
This survey with expert opinion is based on the process (Figure 4.4) defined in Bastos (2011), that structured systematically six steps following different guidelines (Kitchenham and Pfleeger (2002), Mosleh et al. (1987) and Li and Smidts (2003)).

**Setting the Objectives and Survey Design**

In the first step, the objectives are defined, and after that, the survey design is defined (second step). The objective of this survey is to determine, according to the expert opinions, which are the most significant criteria to compare and classify the SIMs. The experts answered a questionnaire containing all selected criteria, asking about what is the relevance of each criterion.

**Developing and Evaluating the Survey Instrument**

The third step involves the development of the survey instrument, which are usually questionnaires. In the literature review, eleven criteria were selected to perform this evaluation. However, some criteria were merged in the survey questionnaire in order to reduce the overall time to answer the questions. Two criteria, research method and research rigor, were merged in the research rigor criterion. Moreover, understanding of services and service granularity have a strict relationship (for example, domain oriented services in most cases tend to be coarse grained). Thus, they were merged in the understanding of services and their granularities criterion.

As a result of this refinement, the survey contains a set of ten questions. Nine questions ask the experts the relevance of each criterion, that is rated through a five-level scale, whose values can be: definitely low, probably low, relative, probably high or
definitely high. The last question asks the experts to propose a criterion not defined in the questionnaire that can be relevant to compare and classify the service identification methods. The questionnaire can be seen in Appendix B.

In the fourth step, the survey instrument is evaluated, and making eventual modifications. A test was performed in order to calibrate the survey. Two RiSE members, researchers in the SOPLE field, participated in this calibration process. The results were satisfactory, with a response time of seven minutes in average. However, the participants had some comments considering the objective of the survey explained before questions (see Appendix B), and some corrections were made.

**Expert Opinion Process**

Next, the experts are selected, contacted, and the data is collected and aggregated. At the end, the data is analyzed to verify the most important criteria based on expert opinions. Our process for expert opinion was inspired in Mosleh et al. (1987) and Li and Smidts (2003). The expert opinion steps are: problem statement, experts selection, opinions elicitation, opinion aggregation and analysis sensibility.

The problem statement refers to the background and problem need to be clearly systematized and defined. The mapping study on SOPLE and the service identification literature review provided a detailed information about background and the problem definition.

The experts selection need to be identified based on a set of criteria, which should include the credibility, knowledge ability, and dependability of experts. We selected two types of experts: the first set of participants should have expertise on service identification, and the authors of all service identification proposals identified in the literature review were invited to participate in this survey. The second group comprises of researchers and practitioners of the SOPLE area.

We performed two surveys, and the main difference among them is the context where the SIM is applied. The first survey evaluated significant criteria of SIMs that are applied in Software Product Lines (called SPL survey), and the second one, evaluated important criteria for SIMs applied in single systems (called SS survey). The goal was to verify if the expert opinion changes if the scenario is different. Appendix B shows details (invitation letter, chronology and questions) of the two surveys. Fifteen SOPLE experts were selected for the SPL survey, and twenty-seven SOC experts with experience of applying SIM in single systems were selected for the SS survey.

- **SOPLE experts**: Sebastian Günther, Jaejoon Lee, Dirk Muthig, Gerald Kotonya,
Doo-Kwon Baik, Laurence Duchien, Klaus Schmid, Robert Krut, Sholom Cohen, Ali Arsanjani, Patricia Lago, Frank Van der Linden, René Boerner, Klaus Pohl and Hassan Gomaa.


The opinions elicitation poses the right question and ensures conditions conducive to an elicitation exercise. Our elicitation is performed through a survey containing ten questions, asking the experts the relevance of nine criteria for comparing and evaluating service identification methods and one question asking about some additional criteria that can be considered.

The goal of the opinion aggregation is to reach an aggregated opinion, that is usually a scalar value representing the scale of each alternative. In this sense, the criteria relevance is rated the defined five-level scale (definitely low, probably low, relative, probably high or definitely high). Finally, the analysis sensibility makes sensitivity analysis based on the aggregated opinion.

After the pilot study, the survey was sent to the experts. The period of interaction with the experts lasted about five months, obtaining six responses for the SPL survey (40% of the total) and nine responses for the SS survey (33% of the total). The criteria were evaluated through a hypothesis test, specifically the Student-T test (Montgomery, 2006) was used in this analysis because the population mean is unknown.

Using the Student test, five-scale values used in the survey were converted to numbers: -2 to definitely low, -1 to probably low, 0 to relative, 1 to probably high and 2 to definitely high. These set of experts responses (numbers) for each criteria built a sample to be tested. The goal was to use the samples for each criterion to verify if the null hypothesis ($H_0$) can be rejected and the alternative hypothesis accepted, that is, if that criterion significance has a tendency to be at least relative.

\[
H_0 : \mu \leq 0 \\
H_1 : \mu > 0
\]

95% confidence interval
4.2. CRITERIA EVALUATION

At the end, the survey analysis determines all criteria at least relative in the survey are part of the most important criteria to compare and classify the methods (research question RQ3, Chapter 1, Section 1.4).

Survey Analysis

The results of the SPL survey showed that all the proposed criteria are significant. The range of values can vary from -2 to 2. The first criterion, understanding of services and their granularities, had the average value of 1.66, i.e., all involved experts pointed this criterion as high or definitely high relevant. The direction of analysis an average value of 1.33, and five of six experts pointed it as high or definitely high significant, and one expert opinion is that this criterion has a low significance.

The input artifacts criterion presented an average value of 1.5, and all involved experts pointed this criterion as high or definitely high relevant. The techniques/activities=strategies criterion had an average of 1, two experts evaluated it as a relative relevance and the other ones pointed it as all high or definitely high relevant. Four experts evaluated the output artifacts criterion as definitely high relevant, one expert pointed it as relative and another one was evaluated as low relevant (average value of 1.1). The orchestration vs. choreography and business/economic aspects criteria presented three relatively high, two high and one relative relevances (average value of 1.3).

The research rigor criterion had one low, one relative, one high and two definitely high relevances (average value of 0.8). The industry sector had one low, three relative, one high and one definitely high relevances (average value of 0.33). In summary, all proposed criteria in the survey were pointed as significant according to the hypothesis tests results.

Furthermore, the experts (37% of the total) proposed two new criteria: Reuse (reuse strategies considered by the method) and Variability (any variability handling technique is adopted).

In the SS survey, the understanding of services and their granularities, had the average value of 1.55, i.e., eight of nine involved experts pointed this criterion as high or definitely high relevant. The direction of analysis had an average value of 0.6, because two experts evaluated it as a low relevance, two of them evaluated this criterion as relatively relevant, and the other ones pointed it as all high or definitely high relevant.

The input artifacts presented an average value of 1.33, and eight of nine involved experts pointed this criterion as high or definitely high relevant. The techniques/activities= strategies criterion had an average of 0.75, because three experts evaluated it as a low
relevance, and the other ones pointed this criterion as high or definitely high relevant. Four experts evaluated the output artifacts criterion as high or definitely high relevant, and five of them evaluated this criterion as relatively or low relevant (average value of 0.44).

The orchestration vs. coreography criteria presented three low, one relative, four high and one relatively high relevances (average value of 0.33). Research rigor had three relative, four high and one definitely high relevances (average value of 0.8). The industry sector had four low, three relative, one high and one definitely high relevances (average value of -0.11). The business/economic aspects criterion had one low, one relative, four high and two definitely high relevances (average value of 1). In summary, all proposed criteria in the survey were pointed as significant according to the hypothesis tests results. According to the hypothesis tests results, only the Industry sector criterion is not significant.

Moreover, the experts proposed some specific criteria such as business rules and business alignment, therefore, these new criteria were merged in the business/economic aspects criterion. Merging the two survey results, we decided to keep the industry sector criterion (excluded in the SS survey), because there is not enough evidence that this criterion is not significant, since it is not rejected in the SPL survey.

One characteristic of the two surveys is that none of the criteria were evaluated as definitely low relevant. The final set of significant criteria is: Service granularity, Understanding of services, Direction of analysis, Input artifacts, Output artifacts, Techniques/activities, Orchestration vs. coreography, Research method, Research rigor, Business/economic aspects, Industry sector, Reuse and Variability.

### 4.3 Methods Comparison

The comparison was performed taking the selected methods (Section 4.1), and comparing them using the most significant criteria (Section 4.2). The resulting comparison is separated in three tables.

Table 4.1 shows the criteria service granularity, understanding of services, direction of analysis, input artifacts and output artifacts. Table 4.2 presents four criteria: variability, reuse, business/economic aspects and industry sector. Table 4.3 shows techniques/activities, orchestration vs. coreography, research method and research rigor, the four remaining criteria.

The first criterion, service granularity (Table 4.1), refers to the service size and the scope of functionality a service exposes. Identifying right-grained services has not been
addressed in the literature, in terms of formal methodologies and tools in order to generate and evaluate alternative designs (Jain et al., 2004; Erradi et al., 2007).

In this comparison, most of the methods consider middle and coarse-grained services as the right granularity. Even methods that identify fine-grained services take consideration about the coarse-grained ones. Only Zhang and Yang (2004) considers the fine-grained as the right granularity. This evidence indicates that middle to coarse-grained services has been largely adopted as a reference for right granularity.

There are different ways for understanding of services in the service identification proposals. The service identification can be oriented to: domain, feature, business process, use case or as module. Service-Oriented Computing has a strong business alignment (Erl, 2005), and the comparison evidence shows that the most common understanding of services is oriented to business processes and business goals, with 20 studies addressing it (66% of the total).

The direction of analysis can differ in top-down, bottom-up and hybrid. Most of the methods take considerations of the top-down direction, it means, the approaches consider “high-level” artifacts as starting point to identify services. Twenty-eight of the thirty methods have a hybrid or top-down direction of analysis (93% of the total). This tendency is strictly related to the understanding of services criterion, because 26 methods use “high-level” artifacts as starting point, such as: business process, feature, use case, application domain, and so on. Only Baghdadi (2006) and Aversano et al. (2008) follows strictly a bottom-up direction.

The input artifacts criterion has a strong relationship with the understanding of services and the direction of analysis criteria. In general, top-down approaches consider “high-level” input artifacts (application domain, business process, and so on.). Furthermore, the input artifact is related to the way of the service is understood. For example, eight methods (Wang et al. (2005), Inaganti and Behara (2007), Dwivedi and Kulkarni (2008), Jamshidi et al. (2008), Mani et al. (2008), Azevedo et al. (2009), Bianchini et al. (2009), Kim and Doh (2009), Yousef et al. (2009)) with business process as input artifact has an understanding of services oriented by business process.

In the output artifacts criterion, results are heterogeneous. The types of output artifacts are: list of services, informal service specification, formal service specification, service implementation, and service model (Gu and Lago, 2010). The most common generated output artifact is the informal specification (12 approaches, 40% of the total), that can be a textual description of the identified services, and the least addressed is the list of services (3 approaches, 10% of the total), consisting of a list of names of candidate ones.
The variability criterion (Table 4.2) is a key aspect in the Software Product Lines Engineering (Clements and Northrop, 2001; Pohl et al., 2005). Surprisingly, only four of the thirty methods concern variability in the proposals. Chen et al. (2005), Kang et al. (2008) and Lee et al. (2008) perform feature analysis to model variability, and the feature model is an input for the service identification activity. The difference is that Chen et al. (2005) start from the source code to build a feature model, and Kang et al. (2008) and Lee et al. (2008) consist of top-down approaches. Kim et al. (2008) model variable goals and scenarios to analyze the future changes for supporting business agility.

SPL is a systematic software reuse technique (Pohl et al., 2005). The experts pointed reuse as a criterion that should be used, and a great number of studies take considerations of reusability in the proposals (18 studies, 60% of the total). There are some adopted reuse strategies, and the most common are reusing existing assets such as legacy code (Chen et al. (2005), Wang et al. (2005), Zhang et al. (2005), and so on.), and identify services that are suitable for reuse (Jain et al. (2004), Dwivedi and Kulkarni (2008), Lee et al. (2008), Cho et al. (2008), and so on.), i.e., the reusability principle of the Service-Oriented Computing (Erl, 2007).

The business/economic aspects and industry sector criteria have different considerations among the methods. Some informations in the comparison can be highlighted, such as Kohlmann and Alt (2007) and Arsanjani et al. (2008), addressing a great number of business/economic aspects (see Table 4.2), as well as the financial (Erradi et al. (2007), Kohlmann and Alt (2007), Arsanjani et al. (2008), Dwivedi and Kulkarni (2008)) and production (Klose et al. (2007), Azevedo et al. (2009), Bianchini et al. (2009), Kohlborn et al. (2009)) domains as the most addressed industry sectors.

The proposed methods address different techniques/activities to perform service identification (Table 4.3). There is a homogeneous distribution among algorithm, analysis, guidelines and ontology techniques. However, pattern and information manipulation have been addressed for one paper, Baghdadi (2006) and Aversano et al. (2008), respectively.

Eighteen papers (60% of the total) did not discussed about the orchestration vs. coreography criterion. However, ten papers (33% of the total) chose orchestration as technique to manage the service-oriented environment, and two papers addressed coreography (0.67% of the total).

The last two criteria are concerned to the evaluation of the proposed service identification methods, that are research method and research rigor. The criteria are related, however the results have not shown this sinergy. Most of the approaches applied case studies, real-life projects, surveys or comparisons to evaluate methods (21 papers, 70%
of the total), and few proposals presented examples or did not evaluated the methods (9 papers, 30% of the total). In contrast, few proposals pay special attention to provide a rich description of the research method (5 papers, 16.7%). Most of the approaches mentioned the research method, or this description is only implicit (25 papers, 83.3% of the total).

### 4.4 Methods Classification

The classification activity uses the previous comparison (Section 4.3) as an input to provide recommendations of the most suitable methods according to the defined SPL scenarios. This recommendation is expressed through decision models (one decision model for each scenario). The decision models classify a method according to its suitability, and identify the activities of each method that are performed before the identification activity.

The resulting decision models can help stakeholders to choose, according to their experienced scenario, which is the suitable method and which activities are desired to perform before the service identification. The decision model building process is specified in the sub-Section 4.4.1, and the resulting decision models are explained in the sub-Section 4.4.2.

#### 4.4.1 Decision Models Building

Bad decisions can be one reason for failures. In this sense, this work used a set of decision models aiming help stakeholders to choose a suitable service identification method for his context. The built decision models will guide stakeholders through the decisions needed to make choices among the identified service identification methods. It is based on Weiss et al. (2008) decision model, that is somewhat different usage of the term than in traditional decision theory (Schoemaker, 1982).

Building decision models presents some challenges. In this work, the first challenge was to determine a classification of the suitable methods using the methods comparison. The adopted alternative was to use recommendation techniques, and the chosen method was the vector space model (Salton et al., 1975), since it is largely used for information filtering (Musto, 2010), information retrieval (Fautsch and Savoy, 2010), and so on.

The idea of the vector space model is to verify the similarity between one entity and the other ones, represented by vectors in a vector space. The result is expressed by similarity coefficients, values between 0 (zero) and 1 (one), and the method closer to 1
is the most similar. In this case, the similarity was measured between a defined ideal method and the existing methods.

The vectors are expressed by numeric values. Consequently, the answers in the comparison table were converted to numbers. They represent the respective weights of each answer, because our assumption is that different answers of a criterion have different significances (weights). Furthermore, each criterion has a particular significance, and as a consequence, they will also have weights assigned. The weight assignment is optimized using the pairwise comparison technique (Koczodaj et al., 1997), in order to reduce the bias of assigning weights - by the researcher judgement - without a systematic process.

The ideal method has its answers with the highest weight for each criterion, and the similarity coefficient will determine how near is the existing method of the ideal one. Pairwise comparison and similarity analysis with vector space model will be explained next.

Pairwise Comparison

The pairwise comparison technique helps to determine a ranking of a group of items (Koczodaj et al., 1997). This is often used in processes to assign weights of evaluation criteria, because it is difficult to tackle all criteria once and discuss what weights should be assigned to each criterion.

In summary, this is a “divide-and-conquer” method where all criteria are analyzed in pairs, verifying the significance of a criterion related to another one. At the end, a certain criteria is compared to the rest, and the appropriate weight is applied to the criterion. The resulting criteria weight assignment is showed in Table 4.4 (the weight assignment for each criterion answer can be seen in the Appendix C). This Table shows the weights for each criterion. Variability, reuse and service granularity received the greater values in the pairwise comparison calculation.

Similarity Analysis

The similarity analysis, as mentioned previously, was performed using the vector space model. This analysis used all weights defined for each criterion that were defined through the pairwise comparison. The calculation of the similarity coefficient follows the formula:

\[
sim(Q, D) = \frac{\sum_{i=1}^{N} w_{Q,i}^{}D_{i}^{}Q_{i}^{}}{\sqrt{\sum_{i=1}^{N} w_{Q,i}^{2}D_{i}^{2}}} \sqrt{\sum_{i=1}^{N} w_{Q,i}^{2}Q_{i}^{2}}
\]

\[Q = \text{ideal method}\]
4.4. METHODS CLASSIFICATION

\[ D = \text{analyzed method} \]
\[ a_{i,X} = \text{answer of the } "i" \text{ criterion for the } "X" \text{ method} \]
\[ w = \text{weight of the criterion} \]
\[ N = \text{number of criteria} \]

An assumption of this research is that in a certain SPL scenario, some methods are not suitable. The factor that determines this exclusion is the input artifact (e.g. feature model, business process, source code) and direction of analysis (top-down, bottom-up and hybrid) of the service identification method. For instance, methods that consider only the code as input artifact cannot be considered in top-down scenarios, which consider “high-level” artifacts as input for the SPL process.

Based on this assumption, the similarity calculation was divided according to the scenarios (top-down, bottom-up and hybrid). Regarding the formula and all weights determined in the pairwise comparison, the similarity results are showed in the tables 4.5, 4.6 and 4.7. Kang et al. (2008), Aversano et al. (2008) and Erradi et al. (2007) were rated with the highest similarity values compared to the other methods in their respective scenarios: top-down, bottom-up and hybrid.

4.4.2 Resulting Decision Models

The resulting decision models are presented according to the SPL scenarios: top-down (Table 4.8), bottom-up (Table 4.9) and hybrid (Table 4.10).

Table 4.8, the top-down decision model, shows the ranking of suitable top-down service identification approaches and the additional activities to perform service identification. These activities are: feature analysis, ontology, business process modeling, domain analysis, use case specification, sequence diagram modeling, data flow diagram modeling, UI design specification, business goal modeling, scenario modeling, cost metrics and requirements specification. Business process modeling is the most required activity to perform service identification in the top-down scenario (7 papers, 46.7% of the total).

In Table 4.9, the bottom-up decision model, the additional activity of the suitable methods is only the legacy code analysis. Finally, the hybrid decision model (Table 4.10) has the following additional activities: business process modeling, use case specification, legacy code analysis, enterprise modeling, financial networking modeling, domain analysis, goal modeling, enterprise architecture modeling and feature analysis. Business process modeling and legacy code analysis are the most required activities to perform
service identification in the hybrid scenario.

Regarding these three decision models, the stakeholder can choose the suitable method for his experienced scenario. The decision making process consists of three tasks: choosing the input artifacts, choosing the required activities and choosing the method by ranking. It is summarized in Figure 4.5.

![Figure 4.5 Decision making process.](image)

The first choice is about the involved input artifacts in the identification method. It will determine which decision model the stakeholder should pay attention. For instance, if the stakeholder is using the source code as input, he should look at the decision model of the bottom-up scenario. In the second decision, the stakeholder concerns choosing the method that fits with the desired activities performed before service identification. He will identify the method that requires his desired activities previously to perform the service identification.

For instance, assuming that the stakeholders chose a top-down scenario and he is interested in methods that perform use case specification, three methods are the candidate ones (Table 4.8): Jain et al. (2004), Huayou et al. (2009) and Kim and Doh (2007).

The last choice is the method ranking. The stakeholders will choose a method that has the best position in the ranking of the most suitable methods considering the experienced scenario. Following the previous example, the stakeholder can choose the Jain et al. (2004) method, because this is the best method considering the similarity analysis (Jain et al. (2004) has ranking value as 3, against 6 of Huayou et al. (2009) and 13 of Kim and Doh (2007)).

The key idea, at the end of these three choices, is to provide a recommendation of the most suitable method through a comprehensive set of informations generated by this
work.

4.5 Threats to Validity

We cannot guarantee all service identification methods and evaluation criteria were included. The strategy to mitigate it was to consider Gu and Lago (2010) systematic review, and perform a manual search in the main SOC and SPL conferences. In addition, secondary search was performed, in order to get methods and criteria looking at references of the primary studies already included (see Section 4.1).

In the comparison table (Section 4.3), the answers for each criterion were determined by researchers. The mitigation strategy was to extract the answers for each criterion more than once, i.e., repeatedly. In some cases, the answers were replaced by a more consistent answer.

Furthermore, the weights determined in Section 4.4 were defined by researcher. However, these weights, created to support the calculation of the most suitable methods in decision models, were defined using pairwise comparison. This technique is more reliable than determining the weights all at once through a not systematic way.

4.6 Chapter Summary

This Chapter presented four steps of the research design: literature review, criteria evaluation, methods comparison and methods classification. The literature review identified thirty service identification methods and a set of criteria to be evaluated in the criteria evaluation activity, that performed a survey with expert opinion in order to verify the most important criteria to compare and classificate them. The comparison synthesizes the results of analyzing all methods in terms of the thirteen most important criteria. At the end, it is presented a recommendation of the most suitable service identification methods through decision models.

Next Chapter presents a case study to evaluate two service identification methods in a health care software product line.
<table>
<thead>
<tr>
<th>Method/Criterion</th>
<th>Service granularity</th>
<th>Understanding of services</th>
<th>Direction of analysis</th>
<th>Input artifacts</th>
<th>Output artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jain et al. (2004)</td>
<td>Middle</td>
<td>Domain oriented</td>
<td>Top down</td>
<td>A list of services</td>
<td>Formal service specification</td>
</tr>
<tr>
<td>Zhang et al. (2004)</td>
<td>Fine</td>
<td>As module</td>
<td>Hybrid (focus on bottom up)</td>
<td>Service model</td>
<td>Service model</td>
</tr>
<tr>
<td>Wang et al. (2005)</td>
<td>Coarse</td>
<td>Business process oriented</td>
<td>Hybrid (focus on bottom up)</td>
<td>Mix</td>
<td>Formal service specification Informal service specification</td>
</tr>
<tr>
<td>Zhang and Yang (2004)</td>
<td>Coarse</td>
<td>As module</td>
<td>Hybrid (focus on bottom up)</td>
<td>Mix</td>
<td>Informal service specification</td>
</tr>
<tr>
<td>Chen et al. (2005)</td>
<td>Fine</td>
<td>As module</td>
<td>Bottom up</td>
<td>Legacy system</td>
<td>Service model</td>
</tr>
<tr>
<td>Wang et al. (2005)</td>
<td>Coarse</td>
<td>Business process oriented</td>
<td>Hybrid (focus on bottom up)</td>
<td>Mix</td>
<td>Service model</td>
</tr>
<tr>
<td>Zhang et al. (2005)</td>
<td>Middle</td>
<td>Business process oriented</td>
<td>Hybrid (focus on bottom up)</td>
<td>Mix</td>
<td>Service model</td>
</tr>
<tr>
<td>Baghdadhi (2006)</td>
<td>Middle</td>
<td>As module</td>
<td>Bottom up</td>
<td>Mixed</td>
<td>Service model</td>
</tr>
<tr>
<td>Erradi et al. (2007)</td>
<td>Coarse</td>
<td>Business process oriented</td>
<td>Bottom up</td>
<td>Mixed</td>
<td>Service model</td>
</tr>
<tr>
<td>Inaganti and Behara (2007)</td>
<td>Middle</td>
<td>Business process oriented</td>
<td>Bottom up</td>
<td>Mixed</td>
<td>Service model</td>
</tr>
<tr>
<td>Kim and Doh (2007)</td>
<td>Middle</td>
<td>Business process oriented</td>
<td>Bottom up</td>
<td>Mixed</td>
<td>Service model</td>
</tr>
<tr>
<td>Kohlborn et al. (2009)</td>
<td>Middle</td>
<td>Business process oriented</td>
<td>Hybrid (focus on bottom up)</td>
<td>Mixed</td>
<td>Service model</td>
</tr>
<tr>
<td>Ricca and Marchetto (2009)</td>
<td>Middle</td>
<td>Business process oriented</td>
<td>Hybrid (focus on bottom up)</td>
<td>Mixed</td>
<td>Service model</td>
</tr>
<tr>
<td>Yousef et al. (2009)</td>
<td>Coarse</td>
<td>Business process oriented</td>
<td>Hybrid (focus on bottom up)</td>
<td>Mix</td>
<td>Service model</td>
</tr>
<tr>
<td>Yun et al. (2009)</td>
<td>Middle</td>
<td>Business process oriented</td>
<td>Top down</td>
<td>Legacy system</td>
<td>Service model</td>
</tr>
</tbody>
</table>

69
### Table 4.2 Methods Comparison - Part II.

<table>
<thead>
<tr>
<th>Method/Criterion</th>
<th>Variability</th>
<th>Reuse</th>
<th>Business/economic aspects</th>
<th>Industry sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jain et al. (2004)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Identifying services that are suitable for reuse</td>
<td>Maintenance and operation costs, SOA controlling</td>
</tr>
<tr>
<td>Zhang and Yang (2004)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Business functions are identified to be reusable</td>
<td>Insurance claims domain</td>
</tr>
<tr>
<td>Chen et al. (2005)</td>
<td>Feature modeling</td>
<td>Not considered</td>
<td>Identifying services that are suitable for reuse</td>
<td>Management Information System</td>
</tr>
<tr>
<td>Wang et al. (2005)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Zhang et al. (2005)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Functions in the legacy system should be identified in order to be reused in the target service</td>
<td>Not considered</td>
</tr>
<tr>
<td>Baghdadi (2006)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Web services reused in composing e-business solutions</td>
<td>Database applications</td>
</tr>
<tr>
<td>Erradi et al. (2007)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Existing software assets, generalized set of services</td>
<td>Value creation, Maintenance and operation costs, Testing effort for new functionality</td>
</tr>
<tr>
<td>Inaganti and Behara (2007)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Kim and Han (2007)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Klose et al. (2007)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Kohlmann and Ah (2007)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Financial services</td>
</tr>
<tr>
<td>Aersanjani et al. (2008)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Financial services</td>
</tr>
<tr>
<td>Klose et al. (2007)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Financial services</td>
</tr>
<tr>
<td>Dwivedi and Kulkarni (2008)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Financial services</td>
</tr>
<tr>
<td>Fareghzadeh (2008)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Jamshidi et al. (2008)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Reusability and reuse efficiency metrics</td>
<td>Not considered</td>
</tr>
<tr>
<td>Kang et al. (2008)</td>
<td>Feature modeling</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Threat evaluation system</td>
</tr>
<tr>
<td>Kim et al. (2008)</td>
<td>Variable goal-scenario model</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Hotel reservation system</td>
</tr>
<tr>
<td>Lee et al. (2008)</td>
<td>Feature analysis</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Virtual office of the future</td>
</tr>
<tr>
<td>Matti et al. (2008)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Cho et al. (2008)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Azevedo et al. (2009)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Degree of reuse criterion for service identification</td>
<td>Production</td>
</tr>
<tr>
<td>Brunelli et al. (2009)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Production</td>
</tr>
<tr>
<td>Chen et al. (2009)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Production</td>
</tr>
<tr>
<td>Huq et al. (2009)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Kim and Han (2009)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Kohlbren et al. (2009)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Roca and Marchetto (2009)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
<tr>
<td>Vassili et al. (2009)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Healthcare</td>
</tr>
<tr>
<td>Yun et al. (2009)</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
<td>Not considered</td>
</tr>
</tbody>
</table>
Table 4.3 Methods Comparison - Part III.

<table>
<thead>
<tr>
<th>Method/Criterion</th>
<th>Techniques/activities</th>
<th>Orchestration vs. coreography</th>
<th>Research method</th>
<th>Research rigor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jain et al. (2004)</td>
<td>Algorithm</td>
<td>Orchestration</td>
<td>Evaluated</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Chen et al. (2005)</td>
<td>Analysis</td>
<td>Not clear</td>
<td>Case study</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Wang et al. (2005)</td>
<td>Algorithm</td>
<td>Orchestration</td>
<td>No</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Zhang et al. (2005)</td>
<td>Algorithm</td>
<td>Choreography</td>
<td>Evaluated</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Baghdada (2006)</td>
<td>Pattern</td>
<td>Not clear</td>
<td>No</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Erradi et al. (2007)</td>
<td>Guidelines</td>
<td>Choreography</td>
<td>Case study</td>
<td>Defined/used</td>
</tr>
<tr>
<td>Inagantii and Behara (2007)</td>
<td>Guidelines</td>
<td>Not clear</td>
<td>No</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Kim and Doh (2007)</td>
<td>Algorithm</td>
<td>Not clear</td>
<td>Example</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Klose et al. (2007)</td>
<td>Guidelines</td>
<td>Orchestration</td>
<td>Case study</td>
<td>Special focus</td>
</tr>
<tr>
<td>Kohlmann and Alt (2007)</td>
<td>Guidelines</td>
<td>Orchestration</td>
<td>Project</td>
<td>Defined/used</td>
</tr>
<tr>
<td>Ansanjamii et al. (2008)</td>
<td>Guidelines</td>
<td>Not clear</td>
<td>Case study</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Nwessanne et al. (2008)</td>
<td>Information manipulation</td>
<td>Not clear</td>
<td>Project</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Dwivedi and Kelkarini (2008)</td>
<td>Algorithm</td>
<td>Orchestration</td>
<td>Example</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Fareghzadeh (2008)</td>
<td>Analysis</td>
<td>Orchestration</td>
<td>Case study</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Jamshidi et al. (2008)</td>
<td>Algorithm</td>
<td>Not clear</td>
<td>Evaluated</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Kang et al. (2008)</td>
<td>Ontology</td>
<td>Not clear</td>
<td>Case study</td>
<td>Defined/used</td>
</tr>
<tr>
<td>Kim et al. (2008)</td>
<td>Guidelines</td>
<td>Orchestration</td>
<td>Case study</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Lee et al. (2008)</td>
<td>Analysis</td>
<td>Orchestration</td>
<td>Case study</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Mani et al. (2008)</td>
<td>Algorithm</td>
<td>Not clear</td>
<td>Case study</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Cho et al. (2008)</td>
<td>Analysis</td>
<td>Orchestration</td>
<td>Example</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Azevedo et al. (2009)</td>
<td>Algorithm</td>
<td>Not clear</td>
<td>Case study</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Bianchini et al. (2009)</td>
<td>Ontology</td>
<td>Not clear</td>
<td>Case study</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Chen et al. (2009)</td>
<td>Ontology</td>
<td>Not clear</td>
<td>Case study</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Huayou et al. (2009)</td>
<td>Guidelines</td>
<td>Not clear</td>
<td>Example</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Kim and Doh (2009)</td>
<td>Algorithm</td>
<td>Not clear</td>
<td>Evaluated</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Kohlboen et al. (2009)</td>
<td>Analysis</td>
<td>Not clear</td>
<td>No</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Ricca and Marchetto (2009)</td>
<td>Guidelines</td>
<td>Orchestration</td>
<td>Case study</td>
<td>Mentioned</td>
</tr>
<tr>
<td>Yousef et al. (2009)</td>
<td>Ontology</td>
<td>Not clear</td>
<td>Case study</td>
<td>Only implicitly</td>
</tr>
<tr>
<td>Yun et al. (2009)</td>
<td>Guidelines</td>
<td>Not clear</td>
<td>Example</td>
<td>Mentioned</td>
</tr>
</tbody>
</table>

Table 4.4 Weight assignment: all criteria.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variability</td>
<td>9</td>
</tr>
<tr>
<td>Reuse</td>
<td>7</td>
</tr>
<tr>
<td>Service granularity</td>
<td>7</td>
</tr>
<tr>
<td>Techniques/activities</td>
<td>4</td>
</tr>
<tr>
<td>Research method</td>
<td>5</td>
</tr>
<tr>
<td>Research rigor</td>
<td>5</td>
</tr>
<tr>
<td>Business/economic aspects</td>
<td>5</td>
</tr>
<tr>
<td>Output artifacts</td>
<td>2</td>
</tr>
<tr>
<td>Industry sector</td>
<td>2</td>
</tr>
<tr>
<td>Orchestration vs. coreography</td>
<td>1</td>
</tr>
</tbody>
</table>
### 4.6. CHAPTER SUMMARY

**Table 4.5** Similarity analysis: Top-down scenario.

<table>
<thead>
<tr>
<th>Method</th>
<th>Similarity value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Kang et al., 2008)</td>
<td>0.972</td>
</tr>
<tr>
<td>(Jamshidi et al., 2008)</td>
<td>0.949</td>
</tr>
<tr>
<td>(Jain et al., 2004)</td>
<td>0.930</td>
</tr>
<tr>
<td>(Yun et al., 2009)</td>
<td>0.898</td>
</tr>
<tr>
<td>(Azevedo et al., 2009)</td>
<td>0.893</td>
</tr>
<tr>
<td>(Huayou et al., 2009)</td>
<td>0.891</td>
</tr>
<tr>
<td>(Yousef et al., 2009)</td>
<td>0.889</td>
</tr>
<tr>
<td>(Mani et al., 2008)</td>
<td>0.887</td>
</tr>
<tr>
<td>(Bianchini et al., 2009)</td>
<td>0.877</td>
</tr>
<tr>
<td>(Lee et al., 2008)</td>
<td>0.866</td>
</tr>
<tr>
<td>(Kim et al., 2008)</td>
<td>0.860</td>
</tr>
<tr>
<td>(Kim and Doh, 2009)</td>
<td>0.854</td>
</tr>
<tr>
<td>(Kim and Doh, 2007)</td>
<td>0.851</td>
</tr>
<tr>
<td>(Cho et al., 2008)</td>
<td>0.778</td>
</tr>
<tr>
<td>(Dwivedi and Kulkarni, 2008)</td>
<td>0.763</td>
</tr>
</tbody>
</table>

**Table 4.6** Similarity analysis: Bottom-up scenario.

<table>
<thead>
<tr>
<th>Method</th>
<th>Similarity value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Aversano et al., 2008)</td>
<td>0.965</td>
</tr>
<tr>
<td>(Baghdadi, 2006)</td>
<td>0.845</td>
</tr>
</tbody>
</table>

**Table 4.7** Similarity analysis: Hybrid scenario.

<table>
<thead>
<tr>
<th>Method</th>
<th>Similarity value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Erradi et al., 2007)</td>
<td>0.924</td>
</tr>
<tr>
<td>(Kohlmann and Alt, 2007)</td>
<td>0.919</td>
</tr>
<tr>
<td>(Klose et al., 2007)</td>
<td>0.894</td>
</tr>
<tr>
<td>(Chen et al., 2009)</td>
<td>0.889</td>
</tr>
<tr>
<td>(Arsanjani et al., 2008)</td>
<td>0.878</td>
</tr>
<tr>
<td>(Kohlborn et al., 2009)</td>
<td>0.820</td>
</tr>
<tr>
<td>(Ricca and Marchetto, 2009)</td>
<td>0.811</td>
</tr>
<tr>
<td>(Zhang et al., 2005)</td>
<td>0.810</td>
</tr>
<tr>
<td>(Chen et al., 2005)</td>
<td>0.778</td>
</tr>
<tr>
<td>(Zhang and Yang, 2004)</td>
<td>0.758</td>
</tr>
<tr>
<td>(Inaganti and Behara, 2007)</td>
<td>0.755</td>
</tr>
<tr>
<td>(Wang et al., 2005)</td>
<td>0.754</td>
</tr>
<tr>
<td>(Fareghzadeh, 2008)</td>
<td>0.741</td>
</tr>
<tr>
<td>Method/Activity</td>
<td>1</td>
</tr>
<tr>
<td>------------------------</td>
<td>---</td>
</tr>
<tr>
<td>Requirements specification</td>
<td></td>
</tr>
<tr>
<td>Cost metrics</td>
<td></td>
</tr>
<tr>
<td>Scenario modeling</td>
<td></td>
</tr>
<tr>
<td>Business goal modeling</td>
<td></td>
</tr>
<tr>
<td>UI design specification</td>
<td></td>
</tr>
<tr>
<td>Data flow diagram modeling</td>
<td></td>
</tr>
<tr>
<td>Sequence diagram modeling</td>
<td></td>
</tr>
<tr>
<td>Class diagram modeling</td>
<td></td>
</tr>
<tr>
<td>Use case specification</td>
<td></td>
</tr>
<tr>
<td>Domain analysis</td>
<td></td>
</tr>
<tr>
<td>Business process modeling</td>
<td></td>
</tr>
<tr>
<td>Ontology</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature analysis</td>
<td>Yes</td>
</tr>
<tr>
<td>Ranking</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.9 Decision model: bottom-up scenario.

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Method/Activity</th>
<th>Legacy code analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aversano et al. (2008)</td>
<td>YES</td>
</tr>
<tr>
<td>2</td>
<td>Baghdadi (2006)</td>
<td>YES</td>
</tr>
<tr>
<td>Feature analysis</td>
<td>Enterprise architecture modeling</td>
<td>Asset analysis</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.10** Decision model: hybrid scenario.
The final step of this work was performed through a case study research. The goal of using this research method is to investigate a contemporary phenomenon within its real-life context, describing and analyzing a bounded system (or case) (Yin, 2008). Case study is characterized by the search for understanding, where the researcher is the primary instrument of data collection and analysis, an inductive investigation strategy, and the end product being richly descriptive (Merriam, 2009).

In this case, the interest is to investigate the service identification activity (the contemporary phenomenon, sub-section 5.1.3) in a product line of medical systems, within its real-life context (sub-section 5.1.2). The protocol, presented in the section 5.1, will explain all the definitions around the case study research, and its results and findings are presented in Section 5.2.

5.1 Case Study Protocol

The structure of this case study protocol is based on Runeson and Höst (2009), Robson (2002) and Maimbo (2003) guidelines. The next sub-sections address the protocol definitions including: objective (5.1.1), the bounded system or case (5.1.2), units of analysis (5.1.3), case study research questions (5.1.4), data collection instruments (5.1.5) and data analysis procedures (5.1.6).

5.1.1 Objective

The goal of applying this case study is to investigate the accuracy of the decision models in terms of recommendation for the most suitable methods. Thus, some methods classified in a decision model will be selected, and their performance will be evaluated in a real-life
5.1. CASE STUDY PROTOCOL

context.

5.1.2 The Case

The company has worked with software development for the medical domain since 1994 and it is located at Salvador, Brazil. It was created offering strategic and operational solutions integrated for hospitals, clinics, labs, and private doctor’s offices having more than 50 clients spread over the country.

The company has 51 employers and four products. The product A is a product of 35 modules or sub-domains, and manages the whole area of a hospital, from financial to patient aspects. The product B, composed of 28 modules, performs the clinical management supporting activities related to medical exams, diagnostics, and so on. The product C has 28 modules and integrates a set of features to manage labs of clinical pathology. Finally, the product D is the web product and is composed of 11 modules to manage the tasks and routines of a doctor’s office. Figure 5.1 shows the relation among the products.

![Figure 5.1 Company products portfolio.](image)

The company decided to transform the set of similar products in an SPL, aiming to take all advantages and benefits of using this technique. They performed the scoping process, and the resulting artifacts are the feature model (Kang et al., 2002) and the product map (DeBaud and Schmid, 1999). The feature model represents features of the product line and the variation of them among products. The product map shows which

---

77
features are present in each product. Furthermore, they are intended to build the core assets guided by a service-oriented approach. It characterizes a SOPLE context - since the company is integrating SPL and SOC techniques - that will be evaluated in this case study.

5.1.3 Units of Analysis

The literature review provided thirty service identification methods that could be evaluated. However, it is not possible to evaluate all methods timely, thus, the decision making process was performed to choose some service identification methods. The first selection criterion is the consideration of methods that address explicitly variability handling, because variability is a key concern in Software Product Lines (Pohl et al., 2005; Clements and Northrop, 2001).

Additionally, another selection criterion determines the required activities and the input artifacts to perform service identification in a real-life context. In this industrial case, the available artifacts are the feature model and product map, and the required activity before service identification is the feature analysis. Performing the decision making process, two service identification methods will be analyzed: Kang et al. (2008) and Lee et al. (2008). They are described as following:

- **Kang et al. (2008) method.** It proposes a solution that identifies services using a ontology in the product line context. They analyze some service identification methodologies in order to define services from the feature model, considering various extensions of feature-oriented domain analysis. Figure 5.2 shows the identification process.

The Kang et al. (2008) identification process consists of three steps:

1. Grouping features: features are grouped by their relations. According to the progress of grouping features, a group unit evolves as a service candidate.
2. Refining service candidate: the service candidate is needed to refine for proper granularity. Service boundary can also refine service candidates.
3. Estimating identified service: Estimating identified service which supports reuse level and extension of service according to both the abstraction degree of development and coupling degree. among services.
• **Lee et al. (2008) method.** This method is based on the feature analysis technique that enables to identify services of a service oriented system. Figure 5.3 shows the identification process.
The Lee et al. (2008) identification process consists of four steps:

1. A feature analysis organizes product family features into an initial model. Within the feature model, the subsequent binding analysis identifies binding units and determines their relative binding times among each others (Lee and Kang, 2004).

2. The service analysis consumes the results of the previous step. Each binding unit is analyzed to determine its service category (i.e., orchestrating service or molecular service) with respect to the particular family at hand. The behaviors of services can be described best by workflows performed by the system users. Furthermore, the context may vary and thus dynamic reconfigurations of product variants are expected.

3. The mass of low level services, called atomic services, are grouped into richer services as required by the family. These richer services are (virtually) composed of atomic services and thus we call them as ‘molecular’ services.

4. The high level services, called orchestrating services, are specified first as workflows and their constituting tasks. Then, their pre/post conditions, invariants, and service interfaces are specified. Finally, the system integration and deployment activity form a product and the orchestrating services provide services to users by integrating and parameterizing the molecular services at runtime.

5.1.4 Case Study Research Questions

The research questions evaluate the case in two perspectives: the software engineer applying the method, and the domain expert analyzing benefits and shortcomings of the identified services.

The software engineer perspective analyzes the technical aspects of the approach, and the domain expert perspective provides informations about how the identified services address the main SPL characteristics (reusability and variability) as well as business aspects. Thus, the following questions were defined:

1. What are the benefits and shortcomings of the approach according to the software engineer perspective?

Rationale: this question investigates the weak and strong aspects of the approach, during its application. These informations are analyzed according to the software
engineer point of view applying the method.

2. **How suitable are the identified services according to the domain expert perspective?**

   Rationale: after collecting informations during application of the methods (by researcher), this question analyzes aspects of the resulting services, according to the domain expert point of view. This main question was divided in three sub-questions, one for each aspect: business needs, reusability and variability.

   (a) **Do the identified services meet the business needs?**

   Rationale: the goal is to verify the business alignment of the identified services, i.e., if the services achieve the business needs of the company specifically, such as value creation, consideration of strategic aspects, and so on.

   (b) **How reusable are the identified services?**

   Rationale: this question verifies if the identified services can be reused among the products (how reusable they are) according to the domain expert opinion.

   (c) **How clear the services express the variability among products?**

   Rationale: in this question, the domain expert evaluates if the identified services represent the existing variability among systems, and what is the relationship between the feature model and services in terms of variability.

### 5.1.5 Data Collection

Yin (2008) discusses about six sources of evidence for case study research: documentation, archival records, interviews, direct observations, participant-observation and physical artifacts. The suitable sources for this research are **documentation and interview**.

Documentary information is likely to be relevant to every case study topic. The use of documents corroborate and augment evidence from other sources, and the researcher can make inferences from documents. They contain exact names, references, and details of an event (Yin, 2008).

The available **documentation** in this case are the output artifacts and activities step-by-step description by the SIMs. They will be evaluated by the researcher in order to investigate the benefits and shortcomings of the approach. In addition, the output artifacts will be analyzed by the domain expert to assess the method suitability to the company needs.
5.1. CASE STUDY PROTOCOL

*Interview* is one of the most important sources of case study information. This instrument is targeted, that is, focuses directly on case study topic. Additionally, interviews are insightful, providing perceived casual inferences ([Yin, 2008](#)).

The type of *interview* adopted in this case study is the semistructured interview. In this type, all questions are more flexibly worded or the interview is a mix of more and less structured questions. The largest part of the interview is guided by a list of questions or issues to be explored ([Merriam, 2009](#)).

The goal is to interview the domain expert in order to identify advantages and drawbacks of the SIMs, in terms of business alignment, reusability and variability. The asked questions in the interview of this case study are presented in Appendix D.

5.1.6 Data Analysis

The strategy for data analysis is relying on theoretical propositions ([Yin, 2008](#)). These propositions reflected the research questions, reviews of literature, and shaped the data collection plan. They help to focus on certain data and to ignore other data.

The propositions for this case study are based on literature review (Section 4.1) and methods comparison (Section 4.3) activities. The literature review presented advantages and disadvantages of the service identification methods, coinciding with the first research question that will be analyzed in this case study.

Other propositions are evidenced in the methods comparison. The analysis using some comparison criteria (business/economic aspects, reuse and variability) suggests that some approaches consider business, reusability and variability aspects to identify services. This theoretical proposition is the foundation for three sub-questions that will be analyzed in this case study.

This strategy is complemented by performing a specific technique, explanation building. The goal is to analyze the case study data building an explanation about the case. According to [Yin (2008)](#), the better case studies are the ones in which explanations have reflected some theoretically significant propositions.

In this case study, the propositions raised from literature review and methods comparison will support the research questions as well as data collection procedures, and the explanation building will analyze how the collected data reflects these propositions.
5.2 Results and Findings

The software engineer (in this case, the software engineer was the author of this work) performed the Kang et al. (2008) and Lee et al. (2008) methods in order to provide the required documentation to be analyzed in this case study. This documentation consists of the list of identified services of each approach, and a brief description of these services.

The time needed to understand the concepts and background of the methods and the application of these approaches was, in average, two days. In addition, the software engineering documented the benefits and drawbacks of using these methods, that will be described in next Section.

The interview was performed with the technical leader of the company, that has more than ten years in the medical domain. The interview guide was created aiming a short meeting with the domain expert, since he could not provide more than thirty minutes for the meeting. As a result, the interview had an average time of eighteen minutes (one interview for the two methods).

The methods were performed using the feature model containing eighty-four features (all existing features in that time for the case study application). In addition, the contact with the methods authors (Kang et al., 2008; Lee et al., 2008) was established in order to obtain the maximum information as possible. Detailed description of case study results are presented in the next sub-sections (5.2.1 and 5.2.2).

5.2.1 Kang et al. (2008) method

Kang et al. (2008) method was applied without further information. The authors were contacted and did not have additional documents, because they stopped investigating the subject after the publication. At the end, an amount of nine services were identified: FinancialService, LaboratoryService, SurgeryCenterService, DoctorsOfficeService, PatientRecordService, PreAdmissionService, AdmissionService, SchedulingService and StockService. Additional information about the services cannot be revealed in consequence of the term of confidentiality.

The approach presents consistent definitions about services and grouping features, i.e., there is no possibility for the software engineers take different meanings of the same definition. Three formal definitions are well-described:

- **Definition 1.** (A Service in Product Line) A Service is end-user visible functionality and non-functionality of product being offered to users in order to satisfy their
5.2. RESULTS AND FINDINGS

- **Definition 2.** (Grouping Features (GF)) represents the degree of feature cohesiveness for a group of features as a set $F$.

- **Definition 3.** (Service boundary ($S_b$)) is a semantic distance from key feature to boundary features. It is measured from a key feature $F_R$ to the other features $F_i$. $F_R$ relates with the number of features $F_i$ as a semantic radius $R$.

In contrast, software engineers must understand the definitions of this method, expressed in semi-formal notation. It can consume an additional time to learn these specifications to apply the identification activity. Thus, it can be an adoption barrier.

The interview with the domain expert points that the expected service granularity for the company was not addressed by this approach. For the expert, the identified services are too coarse grained, i.e., the involved functionalities in a service are larger than expected. He argues that in each service, there are many product features with a considerable development effort.

However, the product variability are well represented according to the domain expert. The variability expressed in the feature model is reflected in the identified services, and the expert could see that some services are mandatory for all products, and other services can be optional in the product derivation.

Another aspect analyzed for him is the service reusability. According to the domain expert opinion, all the identified services in the Kang et al. (2008) approach can be reused for more than one product. In summary, the expert observed that the candidate services follow the characteristics of the feature model in systematizing the variability modeling and providing a large-scale reuse.

### 5.2.2 Lee et al. (2008) method

The Lee et al. (2008) method also was applied without further information. This paper was an output from an industry project and the authors are afraid that no other material is available. The resulting workflow services are: BillingService, BuyingService, LaboratoryService, PatientRecordService, InternacaoService, AgendamentoService and StockService. As mentioned before, additional information about the services cannot be revealed in consequence of the term of confidentiality.

The definition of molecular and workflow services allows a dubious interpretation. Molecular services are self-contained (control + computation) and stateless from service
user’s point of view, and workflow services perform a dependable orchestration of molecular services.

In this sense, another software engineer can see some identified workflow services as molecular services, since he has a different notion about the granularity of workflow and molecular services. Nevertheless, the concepts of these kind of services are easy to understand, and software engineers can learn this method in a reasonable time.

In another point of view, the domain expert argued that the identified services has the right granularity for the company business needs. According to him, the workflow services can orchestrate a set of molecular services, because these molecular services can implement the self-contained and fine-grained functionalities of the products portfolio, that was not covered for the previous approach.

The variability are also well represented in this method according to the expert. As discussed in the previous approach, the variability modeled in the feature model influences the identified services, and there are some mandatory services, and optional/alternative ones.

In terms of service reusability, the expert argued that the Lee et al. (2008) approach can also be reused for more than one product. However, comparing atomic and molecular services, the expert identified the atomic services are more reusable, because they implement self-contained functionalities, and workflow services are too fine-grained and consequently more difficult to be reused.

5.3 Threats to Validity

The case study quality (or validity) denotes the trustworthiness of the results, to what extent the results are not biased by the researchers subjective point of view. The quality must be addressed during all the previous phases of the case study (Runeson and Höst, 2009).

Yin (2008) uses a classification scheme that describes the aspects of validity and threats to validity of a study. There are four aspects: construct validity, internal validity, external validity and reliability. Hereafter will be explained how these aspects are addressed as well as the threats are mitigated in this case study.

- **Construct validity.** This aspect of validity reflects to what extent the operational measures that are studied really represent what the researcher has in mind and what is investigated is conform to the research questions. For example, if the constructs
discussed in the interview questions are not interpreted in the same way by the researcher and the interviewed persons, there is a threat to the construct validity. In order to mitigate this threat, this research is regarding multiples sources of evidence (documentation and interview information), as a way of encouraging convergent lines of inquiry.

- **Internal validity.** This aspect of validity is of concern when causal relations are examined. This case study built the research design aiming to anticipate the possible inferences made in this study, as an adopted strategy to improve internal validity.

- **External validity.** This aspect of validity is concerned with to what extent it is possible to generalize the findings, and to what extent the findings are of interest to other people outside the investigated case. The findings obtained in this research are supported by theory, that must be tested by replicating the findings in other cases.

- **Reliability.** This aspect is concerned with to what extent the data and the analysis are dependent on the specific researchers. Hypothetically, if another researcher later on conducted the same study, the result should be the same. We developed a detailed protocol, in order to clarify all activities of this case study.

### 5.4 Chapter Summary

This Chapter presented the information about the performed case study: objective, the case, units of analysis, data collection instruments, data synthesis procedures, results and the main findings of this activity.

Identifying services is not a trivial task (Inaganti and Behara, 2007), however Kang et al. (2008) and Lee et al. (2008) provide a guide to facilitate the identification of right-grained services using feature model and product map artifacts. Furthermore, the approaches achieved the benefits related to the business needs of the company, reusability and variability modeling.

Next Chapter will present the related work, future work and concluding remarks of this dissertation.
Conclusion

This dissertation verified a set of service identification methods in order to investigate the most significant approaches for three SPL scenarios: bottom-up, top-down, and hybrid. The research design to address it consisted of five activities: literature review, criteria evaluation, methods comparison, methods classification and methods evaluation.

The literature review identified thirty methods regarding three sources of evidence, Gu and Lago (2010) systematic review, manual search in four conferences, and “snowballing” search (or secondary search). Furthermore, this task selected existing comparison and classification criteria.

The second activity, criteria evaluation, used the comparison and classification criteria to verify which are the most important aspects to compare and classify the methods. An expert opinion survey was performed, and the results pointed that the experts considered all criteria important. Moreover, they suggested two other criteria: reuse aspects and variability.

All selected SIMs were compared based on the resulting list of significant criteria. In addition to provide an in-depth comparison of the methods, this activity was used as an input for the methods classification activity.

The methods classification was expressed through decision models, one for each SPL scenario. The goal was to indicate a ranking of suitable SIMs for top-down, bottom-up and hybrid scenarios. The decision model is showed for the stakeholder in terms of similarity analysis (how near the method is of a ideal methods), and mandatory activities performed before service identification.

At the end, two methods were evaluated through a case study performed in a medical domain. The analyzed methods (Kang et al. (2008) and Lee et al. (2008)) provided systematic steps facilitating the identification activities, presenting benefits in terms of business, reusability and variability modeling for the company.
6.1 Research Contributions

The main contributions of this research are described as follows:

• **SOPLE mapping study.** This mapping study is based on Petersen et al. (2007) and Kitchenham and Charters (2007) guidelines answering a set of seven research questions. It provided some evidence regarding the SOPLE area.

• **An extended literature review on service identification methods.** Although our literature review is based on Gu and Lago (2010) systematic review, this work extended the results performing manual search and the “snowballing” technique. These activities helped to find two relevant service identification methods (Erradi et al. (2007) and Arsanjani et al. (2008)) that were not covered for the systematic review.

• **A set of evaluated criteria to compare and classify service identification methods.** The literature review provided a set of criteria, and this work analyzed the most important ones. This activity was supported by a survey with expert opinion. At the end, thirteen important criteria were used for comparison and classification of the service identification methods.

• **A detailed comparison involving the existing service identification methods collected from the literature.** The most important criteria were used to perform a detailed comparison of the existing service identification methods. The stakeholders can use this comparison to investigate the specific characteristics of each method and eventually select a suitable approach for him.

• **A set of decision models considering three SPL scenarios.** Our classification uses the comparison results as an input to provide a reasonable recommendation of the suitable methods according to the stakeholder needs. The recommendation is expressed through decision models, that classifies a method according to its suitability, and identifies the required activities to perform the identification activity.

• **An evaluation of two service identification methods in a real-life SPL context.** The case study research investigates a contemporary phenomenon within its real-life context, describing and analyzing a bounded system (Yin, 2008). The case study findings can help stakeholders to evaluate the behavior of the suitable service identification methods in a product line context. Furthermore, it can help researchers to
perform case studies in different real-life contexts aiming to increase the amount of empirical studies in the service identification field.

6.2 Future Work

This work can be seen as an in-depth analysis of service identification methods for software product lines. Due to the time constraints imposed on the master degree, other activities cannot be performed in order to improve the confidence of this research. Therefore the following activities should be performed as future work:

- **Performing other recommendation techniques.** In this work, only the vector space model (Salton et al., 1975) was applied to verify the most suitable methods. Thus, it is important to investigate more recommendation techniques in order to analyze them and select the most appropriate one, aiming to provide a suitable recommendation for the stakeholders.

- **Case studies applied in bottom-up and hybrid scenarios.** In the performed case study, the industrial case was based on a top-down scenario. Bottom-up and hybrid scenarios were not evaluated in this work. Therefore, more case studies should be performed. The goal is to identify cases that fit in these remaining scenarios, select the most suitable methods according to our decision making process, and perform the case studies.

- **Evaluating service design and implementation.** This work is concerned to evaluate the impact of the service identification methods in the service design and implementation activities, identifying benefits and drawbacks of using these approaches.

6.3 Concluding Remarks

The goal of this work was to investigate the application of service identification methods in Software Product Lines. It is motivated by an emerging area, Service-Oriented Product Line Engineering (SOPLE), that combines activities from Service-Oriented Computing and Software Product Lines issues such as: dynamic reconfigurability, variability handling, service identification, and so on.

This research used a multi-method approach to identify the available service identification methods that can be applied in a SPL context, and to determine the most suitable
6.3. CONCLUDING REMARKS

methods given top-down, bottom-up and hybrid scenarios. Three research methods were combined to address it: mapping study (and systematic review), survey with expert opinion and case study.

Our findings helped to conclude that the service identification methods can be applied in SPLs, and the evidence show that there is not requirements to create a method that are not covered for the existing service identification approaches. In addition, there are some evidence suggesting that these methods meets the business needs of the company, filling the gap between technical and business aspects. Some methods also concern reusability strategies and variability handling in the identification activity.


97


Medeiros, F. M., de Almeida, E. S., and de Lemos Meira, S. R. (2010). Designing a set of service-oriented systems as a software product line. *Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS)*, 0, 70–79.


Appendices
<table>
<thead>
<tr>
<th>Method</th>
<th>Input type</th>
<th>Strategy</th>
<th>Output format</th>
<th>Output type</th>
<th>Technique</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jain et al. (2004)</td>
<td>Application domain</td>
<td>W2</td>
<td>WS</td>
<td>A list of services</td>
<td>Algorithm</td>
<td>Evaluated</td>
</tr>
<tr>
<td>Wang et al. (2005)</td>
<td>Business process</td>
<td>W1</td>
<td>PS+BS+IS</td>
<td>Service model</td>
<td>Algorithm</td>
<td>No</td>
</tr>
<tr>
<td>Chen et al. (2005)</td>
<td>Legacy system</td>
<td>W7</td>
<td>WS</td>
<td>Service implementation</td>
<td>Analysis</td>
<td>Case study</td>
</tr>
<tr>
<td>Zhang et al. (2005)</td>
<td>Legacy system</td>
<td>W7</td>
<td>WS</td>
<td>Service implementation</td>
<td>Algorithm</td>
<td>Case study</td>
</tr>
<tr>
<td>Baghdadi (2006)</td>
<td>Data</td>
<td>W3</td>
<td>DS</td>
<td>Formal service specification</td>
<td>Pattern</td>
<td>No</td>
</tr>
<tr>
<td>Klein et al. (2007)</td>
<td>Application domain</td>
<td>W1&amp;4</td>
<td>BS+DS+IS+CS</td>
<td>Informal service specification</td>
<td>Guidelines</td>
<td>Case study</td>
</tr>
<tr>
<td>Agung et al. (2007)</td>
<td>Business process</td>
<td>W1</td>
<td>BS+IS</td>
<td>Service model</td>
<td>Guidelines</td>
<td>No</td>
</tr>
<tr>
<td>Kim and Dol (2007)</td>
<td>Use case</td>
<td>W2&amp;4</td>
<td>BS+CS</td>
<td>Informal service specification</td>
<td>Algorithm</td>
<td>Example</td>
</tr>
<tr>
<td>Amedon (2007)</td>
<td>Business process</td>
<td>W1</td>
<td>BS+CS</td>
<td>Service model</td>
<td>Analysis</td>
<td>Example</td>
</tr>
<tr>
<td>Fareghzadeh (2008)</td>
<td>Mix</td>
<td>W5&amp;7</td>
<td>BS+DS+IS+CS</td>
<td>Service model</td>
<td>Analysis</td>
<td>Case study</td>
</tr>
<tr>
<td>Atesi et al. (2008)</td>
<td>Application domain</td>
<td>W5</td>
<td>BS+CS</td>
<td>Service model</td>
<td>Guidelines</td>
<td>Case study</td>
</tr>
<tr>
<td>Man et al. (2008)</td>
<td>Business process</td>
<td>W1&amp;11</td>
<td>BS+DS+CS</td>
<td>Service implementation</td>
<td>Algorithm</td>
<td>Case study</td>
</tr>
<tr>
<td>Jamshidi et al. (2008)</td>
<td>Business process</td>
<td>W1&amp;3</td>
<td>PS+IS</td>
<td>Service model</td>
<td>Algorithm</td>
<td>Evaluated</td>
</tr>
<tr>
<td>Dwivedi and Kakkani (2008)</td>
<td>Business process</td>
<td>W1</td>
<td>PS+BS+IS+IS+BS+CS</td>
<td>A list of services</td>
<td>Algorithm</td>
<td>Example</td>
</tr>
<tr>
<td>Lee et al. (2008)</td>
<td>Feature</td>
<td>W2</td>
<td>BS</td>
<td>Informal service specification</td>
<td>Analysis</td>
<td>Case study</td>
</tr>
<tr>
<td>Kang et al. (2008)</td>
<td>Feature</td>
<td>W2</td>
<td>BS</td>
<td>Informal service specification</td>
<td>Ontology</td>
<td>Case study</td>
</tr>
<tr>
<td>Azevedo et al. (2008)</td>
<td>Legacy system</td>
<td>W7</td>
<td>WS</td>
<td>Service implementation</td>
<td>Information manipulation</td>
<td>Project</td>
</tr>
<tr>
<td>Che et al. (2008)</td>
<td>Mix</td>
<td>W1&amp;5</td>
<td>BS+DS+CS</td>
<td>Informal service specification</td>
<td>Analysis</td>
<td>Example</td>
</tr>
<tr>
<td>Kohlbacher et al. (2009)</td>
<td>Application domain</td>
<td>W2&amp;5</td>
<td>BS+DS+IS</td>
<td>Informal service specification</td>
<td>Analysis</td>
<td>No</td>
</tr>
<tr>
<td>Buchtal et al. (2009)</td>
<td>Business process</td>
<td>W1</td>
<td>BS</td>
<td>A list of services</td>
<td>Ontology</td>
<td>Case study</td>
</tr>
<tr>
<td>Violati et al. (2009)</td>
<td>Business process</td>
<td>W1</td>
<td>BS+DS+IS</td>
<td>Service model</td>
<td>Ontology</td>
<td>Case study</td>
</tr>
<tr>
<td>Azevedo et al. (2009)</td>
<td>Business process</td>
<td>W1</td>
<td>BS+DS+IS</td>
<td>Informal service specification</td>
<td>Algorithm</td>
<td>Case study</td>
</tr>
<tr>
<td>Kim and Dol (2009)</td>
<td>Business process</td>
<td>W1</td>
<td>PS+IS</td>
<td>Service implementation</td>
<td>Algorithm</td>
<td>Evaluated</td>
</tr>
<tr>
<td>Atesi et al. (2009)</td>
<td>Legacy system</td>
<td>W7</td>
<td>WS</td>
<td>Service implementation</td>
<td>Guidelines</td>
<td>Case study</td>
</tr>
<tr>
<td>Huynh et al. (2009)</td>
<td>Use case</td>
<td>W2&amp;5</td>
<td>BS+CS</td>
<td>Informal service specification</td>
<td>Guidelines</td>
<td>Example</td>
</tr>
<tr>
<td>Yun et al. (2009)</td>
<td>Data</td>
<td>W3</td>
<td>PS+BS+CS</td>
<td>Informal service specification</td>
<td>Guidelines</td>
<td>Example</td>
</tr>
<tr>
<td>Rocca and Marchisio (2009)</td>
<td>Mix</td>
<td>W1&amp;7</td>
<td>WS</td>
<td>Service implementation</td>
<td>Guidelines</td>
<td>Case study</td>
</tr>
<tr>
<td>Strategy</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W1 (Business Process Decomposition)</td>
<td>Decompose business process models that depict how the work is done within an organization.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W2 (Business Functions)</td>
<td>Decompose business function models that depict what an organization does.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W3 (Business Entity Objects)</td>
<td>Model services according to business object models.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W4 (Ownership and Responsibility)</td>
<td>Take the ownership of processes into consideration when identifying services.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W5 (Goal driven)</td>
<td>Decompose a company’s goals down to the level of services.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W6 (Component-Based)</td>
<td>Identifies services based components.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W7 (Existing Supply)</td>
<td>Identify services from the functionality provided by existing legacy applications.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W8 (Front-Office Application Usage Analysis)</td>
<td>Select a set of applications that support business processes and extracts comparable functions into a single service.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W9 (Infrastructure)</td>
<td>Keep the technical infrastructure into consideration when identifying services.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W10 (NFRs)</td>
<td>Use non-functional requirements as the primary input to identify services.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W11 (User interface)</td>
<td>Identify services based on the design of user interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output type</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business process service (BS)</td>
<td>A service that has the business logic or represents a business process, including task services, process services.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data service (DS)</td>
<td>A service that represents business centric entities, including information services, entity services.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Composite Service (CS)</td>
<td>A composition of multiple services.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IT service (IS)</td>
<td>A service that represents technology specific functionalities, including application services, software services, utility services and infrastructure services.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web service (WS)</td>
<td>A service that is implemented using the web service technology. This type is orthogonal to the other types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partner service (PS)</td>
<td>A service that is offered to external partners.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
B.1 SPL Survey

B.1.1 Invitation Letter

Service Identification Criteria for Software Product Lines Survey

Dear (expert name),

This questionnaire is part of the Service-Oriented Product Line Engineering (SOPLE) research conducted by the Reuse in Software Engineering (RiSE) Labs whose goal is to investigate Service Identification Methods for Software Product Lines.

The questionnaire is designed to discover relevant criteria of service identification methods for software product lines based on expert opinion. The results of the survey will be used to indicate suitable methods for some software product lines scenarios.

We chose you because consider you as an expert in the SOPLE area. All the information provided will be kept confidential and we will just publish the name of the experts in the study. We have no intention to judge your work since we are merely interested in learning about some aspects of your work.

Finally, the survey is composed of 9 objective questions and 1 subjective question (all the questions are alternative). These questions can be concluded in less than 7 minutes. Please, return it until Sep 15th, 2011.

Survey URL: https://spreadsheets.google.com/viewform?formkey=dGkzM0pQdGVGbTNtT2w2VmlYZkdvaUE6MQ

B.1.2 Chronology

• May 17th, 2011: Sent to the co-advisor
B.1. SPL SURVEY

- May 17th, 2011: Feedback from co-advisor
- May 18th, 2011: Pre-testing with RiSE members (Jorge, Flavio)
- May 19th, 2011: Response from Jorge e Flavio
- May 24th, 2011: Experts’ Invitation
- May 24th, 2011: Delivery status notification (failure): Robert Krut and Laurence Duchien / Office notice: Klaus Pohl and René Boerner
- Jun 15th, 2011: Experts’ reminder
- Jun 16th, 2011: Response - Frank Van der Linden
- Jun 23th, 2011: Response - Hassan Gomaa
- Jul 13th, 2011: Response - Jaejoon Lee
- Jul 20th, 2011: Response - Patricia Lago
- Aug 15th, 2011: Response - Sholom Cohen
- Sep 8th, 2011: Response - Dirk Muthig
- Sep 15th, 2011: Analysis

B.1.3 Questionaire

Service Identification Methods Comparison Criteria for Software Product Lines: A Survey

This survey investigates the significance and influence of specific criteria to identify services in some Software Product Lines (SPLs) scenarios. In previous research, criteria to identify services in single systems were defined. Based on some of them, we defined a list of candidate criteria for identification in the context of Software Product Line Engineering. For our research, your opinion is very important to refine them.

The defined Software Product Line scenarios are:

a. Top-down scenario

The first step of the proactive (1) adoption model is to perform domain analysis and scoping to identify the variation to be supported in the production line. The reactive (2) approach initially characterizes the requirements for the new product relative to what
is currently supported in the production line. In general, these activities produce only “high-level” artifacts, that is, they do not consider “low-level” artifacts (like source code) before the SPL implementation. Therefore, the available assets to identify services in the top-down scenario are: feature models, requirements and use cases.

b. Bottom-up scenario
The bottom-up scenario considers the extrative (2) adoption model, where the used assets in the initial tasks to build Software Product Lines are extracted from the existing/legacy systems (source code). Therefore, the available asset to identify services is source code.

c. Hybrid scenario
The hybrid scenario considers the two previous scenarios, consequently the tree adoption models. Therefore, the available assets are from “high and low level”, such as: feature models, requirements, use cases and source code.

1. The proactive adoption model to software product lines is like the waterfall approach to conventional software. You analyze, architect, design, and implement all product variations on the foreseeable horizon up front.

2. With the reactive approach, the organization incrementally grows their software mass customization production line when the demand arises for new products or new requirements on existing products.

3. With the extractive approach, the organization capitalizes on existing custom software systems by extracting the common and varying source code into a single production line.


QUESTIONS

• Your name:

• Understanding of services and their granularities
Some consider a service comprehensively, i.e. it represents a complete business process. On the other hand, service represents a single task. Service granularity refers to the service size and the scope of functionality a service exposes.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

- **Direction of the analysis (bottom-up, top-down or hybrid)**

Top-down approaches have high-level artifacts (design artifacts, functional/business requirements, etc.) as the start point. Bottom-up approaches consider source code (legacy systems) as the start point, and hybrid approaches combine the two previous approaches.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

- **Input artifacts**

e.g. feature models, requirements, use cases, source code (legacy systems), business processes and so on.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

- **Output artifacts (results)**

Artifact that is produced through performing the identification method.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

- **Techniques / activities / strategies**

Instructions that describes the course of action within an activity.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

- **Orchestration vs. Coreography**

Orchestration implies a central instance that coordinates all activities of a process. Choreography means that services are called by other services and there is not one steering unit.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )
• **Research rigor**

It corresponds to the applied methodologies and the evaluation/validation methods (e.g. case study, experiment, and so on).

Criteria significance/influence:
- Definitely low ( )
- Probably low ( )
- Relative ( )
- Probably high ( )
- Definitely high ( )

• **Industry sector**

Industry sector where the method is applied.

Criteria significance/influence:
- Definitely low ( )
- Probably low ( )
- Relative ( )
- Probably high ( )
- Definitely high ( )

• **Business/economic aspects**

e.g. value creation, consideration of strategic aspects, customer satisfaction, time-to-market, legal compliance, service level agreements, IT governance, and so on.

Criteria significance/influence:
- Definitely low ( )
- Probably low ( )
- Relative ( )
- Probably high ( )
- Definitely high ( )

• **Do you consider other criteria in the service identification phase? What is their significance/influence?**

Another criteria not cited above:

---

**B.2 SS Survey**

**B.2.1 Invitation letter**

Service Identification Criteria Survey

Dear (expert name),

This questionnaire is part of the Service-Oriented Product Line Engineering (SOPLE) research conducted by the RiSE (Reuse in Software Engineering) Labs whose goal is to investigate Service Identification Methods for Software Product Lines.

The questionnaire is designed to discover relevant criteria of service identification methods based on expert opinion, and the results will be used to evaluate the existing methods.
We chose you because consider you as an expert in the Service Identification area. All the information provided will be kept confidential and we will just publish the name of the experts in the study. We have no intention to judge your work since we are merely interested in learning about some aspects of your work.

Finally, the survey is composed of 9 objective questions and 1 subjective question (all questions are alternatives). These questions can be concluded in less than 6 minutes. Please, return it until Sep 15th, 2011.

Survey URL: https://spreadsheets.google.com/spreadsheet/viewform?formkey=dGtFUzhSUWZXcFdySI1BWT2ktdWNNZGc6MA

B.2.2 Chronology

- May 17th, 2011: Sent to the co-advisor
- May 17th, 2011: Feedback from co-advisor
- May 18th, 2011: Pre-testing with RiSE members (Jorge, Flavio)
- May 19th, 2011: Response from Jorge e Flavio
- Jun 15th, 2011: Experts’ Invitation
- Jun 15th, 2011: Delivery status notification (failure): Zhuopeng Zhang, Ralf Knackstedt, Ruimin Liu
- Jun 16th, 2011: Response - Fernanda Baião, Cinzia Cappiello, Flávia Maria Santoro
- Jun 16th, 2011: Warning message - Youcef Baghdadi
- Jun 20th, 2011: Response - Nafiseh Fareghzadeh
- Jul 8th, 2011: Experts’ reminder
- Jul 9th, 2011: Response - Luigi Cerulo
- Jul 12th, 2011: Response - Kyung-Goo Doh
- Jul 21th, 2011: Response - Alessandro Marchetto
- Jul 25th, 2011: Response - Naveen Kulkarni
- Jul 29th, 2011: Response - Rainer Alt
- Sep 15th, 2011: Analysis
B.2.3 Questionnaire

Service Identification Criteria Survey

This survey investigates the significance and influence of specific criteria to identify services. In previous research, criteria to identify services in single systems were defined. Based on some of them, we defined a list of candidate criteria for service identification. For our research, your opinion is very important to refine them.

QUESTIONS

• Your name:

• Understanding of services and their granularities

Some consider a service comprehensively, i.e. it represents a complete business process. On the other hand, service represents a single task. Service granularity refers to the service size and the scope of functionality a service exposes.

Criteria significance/influence:
Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

• Direction of the analysis (bottom-up, top-down or hybrid)

Top-down approaches have high-level artifacts (design artifacts, functional/business requirements, etc.) as the start point. Bottom-up approaches consider source code (legacy systems) as the start point, and hybrid approaches combine the two previous approaches.

Criteria significance/influence:
Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

• Input artifacts

e.g. feature models, requirements, use cases, source code (legacy systems), business processes and so on.

Criteria significance/influence:
Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

• Output artifacts (results)

Artifact that is produced through performing the identification method.

Criteria significance/influence:
Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )
• **Techniques / activities / strategies**

Instructions that describes the course of action within an activity.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

• **Orchestration vs. Coreography**

Orchestration implies a central instance that coordinates all activities of a process. Choreography means that services are called by other services and there is not one steering unit.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

• **Research rigor**

It corresponds to the applied methodologies and the evaluation/validation methods (e.g. case study, experiment, and so on).

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

• **Industry sector**

Industry sector where the method is applied.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

• **Business/economic aspects**

e.g. value creation, consideration of strategic aspects, customer satisfaction, time-to-market, legal compliance, service level agreements, IT governance, and so on.

Criteria significance/influence:
- Definitely low ( ) Probably low ( ) Relative ( ) Probably high ( ) Definitely high ( )

• **Do you consider other criteria in the service identification phase? What is their significance/influence?**

Another criteria not cited above:
Pairwise Comparison Results

Table C.1  Weight assignment: service granularity.

<table>
<thead>
<tr>
<th>Service granularity</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse</td>
<td>4</td>
</tr>
<tr>
<td>From coarse to fine</td>
<td>3</td>
</tr>
<tr>
<td>Middle</td>
<td>2</td>
</tr>
<tr>
<td>Fine</td>
<td>1</td>
</tr>
</tbody>
</table>

Table C.2  Weight assignment: output artifacts.

<table>
<thead>
<tr>
<th>Output artifacts</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service implementation</td>
<td>5</td>
</tr>
<tr>
<td>Formal service specification</td>
<td>4</td>
</tr>
<tr>
<td>Service model</td>
<td>3</td>
</tr>
<tr>
<td>Informal service specification</td>
<td>2</td>
</tr>
<tr>
<td>A list of services</td>
<td>1</td>
</tr>
</tbody>
</table>

Table C.3  Weight assignment: techniques/activities.

<table>
<thead>
<tr>
<th>Techniques/activities</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>7</td>
</tr>
<tr>
<td>Pattern</td>
<td>7</td>
</tr>
<tr>
<td>Information manipulation</td>
<td>7</td>
</tr>
<tr>
<td>Ontology</td>
<td>7</td>
</tr>
<tr>
<td>Guidelines</td>
<td>2</td>
</tr>
<tr>
<td>Analysis</td>
<td>1</td>
</tr>
</tbody>
</table>
Table C.4  Weight assignment: orchestration vs. coreography.

<table>
<thead>
<tr>
<th>Orchestration vs. coreography</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orchestration</td>
<td>1</td>
</tr>
<tr>
<td>Coreography</td>
<td>1</td>
</tr>
<tr>
<td>Not clear</td>
<td>0</td>
</tr>
</tbody>
</table>

Table C.5  Weight assignment: research method.

<table>
<thead>
<tr>
<th>Research method</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case study</td>
<td>4</td>
</tr>
<tr>
<td>Project</td>
<td>4</td>
</tr>
<tr>
<td>Evaluated</td>
<td>2</td>
</tr>
<tr>
<td>Example</td>
<td>1</td>
</tr>
<tr>
<td>No</td>
<td>0</td>
</tr>
</tbody>
</table>

Table C.6  Weight assignment: research rigor.

<table>
<thead>
<tr>
<th>Research rigor</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special focus</td>
<td>4</td>
</tr>
<tr>
<td>Defined/used</td>
<td>4</td>
</tr>
<tr>
<td>Mentioned</td>
<td>2</td>
</tr>
<tr>
<td>Only implicitity</td>
<td>1</td>
</tr>
</tbody>
</table>

Table C.7  Weight assignment: Industry sector.

<table>
<thead>
<tr>
<th>Industry sector</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Considered</td>
<td>1</td>
</tr>
<tr>
<td>Not considered</td>
<td>0</td>
</tr>
</tbody>
</table>

Table C.8  Weight assignment: reuse.

<table>
<thead>
<tr>
<th>Reuse</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Considered</td>
<td>1</td>
</tr>
<tr>
<td>Not considered</td>
<td>0</td>
</tr>
</tbody>
</table>

Table C.9  Weight assignment: Variability.

<table>
<thead>
<tr>
<th>Variability</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Considered</td>
<td>1</td>
</tr>
<tr>
<td>Not considered</td>
<td>0</td>
</tr>
</tbody>
</table>

Table C.10  Weight assignment: Business aspects.

<table>
<thead>
<tr>
<th>Business aspects</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Considered</td>
<td>1</td>
</tr>
<tr>
<td>Not considered</td>
<td>0</td>
</tr>
</tbody>
</table>
Case Study Interview

Interview guide

INITIAL PHASE

1) Explanation of the SOPLE area
   - Combination of SPL and SOC
   - Motivation for using services

2) Interview proposal

3) What are your responsibilities in this company?

RESEARCH QUESTIONS

4) Show the identified services of the method
   (Three aspects: reusability, variability and business alignment)

5) Can the identified services be used for more than one product?

6) Do the services express clearly the variability among the products?

7) Do the identified services meet the business needs of the company?

FINAL

Thanks for your attention.
Dissertação de Mestrado apresentada por Tassio Ferreira Vale à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "Using a Multi-Method Approach for Evaluating Service Identification Methods in Service-Oriented Product Lines" orientada pelo Prof. Silvio Romero de Lemos Meira e aprovada pela Banca Examinadora formada pelos professores:

Prof. Fernando José Castor de Lima Filho  
Centro de Informática / UFPE

Profa. Rossana Maria de Castro Andrade  
Departamento de Computação / UFAL

Prof. Silvio Romero de Lemos Meira  
Centro de Informática / UFPE

Visto e permitida a impressão.  
Recife, 12 de março de 2012

Prof. Nelson Souto Rosa  
Coordenador da Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco.
Vale, Tassio Ferreira
xv, 120 folhas: il., fig., tab.

Orientador: Silvio Romero de Lemos Meira.
Dissertação (mestrado) - Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2012.
Inclui bibliografia e apêndice.

1. Engenharia de software. 2. Linhas de produtos de software. I. Meira, Silvio Romero de Lemos (orientador). II. Título.

005.1 CDD (23. ed.) MEI2012 – 046