PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Universidade Federal do Rio de Janeiro

ROSANE SFAIR HUERGO

MDCSIM: a method to identify services using master data, logical data models and artefact-centric modeling approach.

DISSERTAÇÃO DE MESTRADO

Rio de Janeiro
2014

Instituto de Matemática

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais

ROSANE SFAIR HUERGO

MDCSIM: a method to identify services using master data, logical data models and artifact-centric modeling approach.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, como requisito parcial à obtenção do título de Mestre em Informática.

Orientador: Prof. Paulo de Figueiredo Pires, D.Sc

Rio de Janeiro
2014

H887 Huergo, Rosane Sfair

MDCSIM: a method to identify services using master data, logical data models and artifact-centric modeling approach / Rosane Sfair Huergo— 2014.

163 f.: il.: 31 cm.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti , Rio de Janeiro, 2014.

Orientador: Paulo de Figueiredo Pires

1. Arquitetura Orientada a Serviços. 2. Dados Mestres. 3. Identificação de serviços – Teses. I. Pires, Paulo de Figueiredo (Orient.). II. Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti. III. Título.

CDD:

ROSANE SFAIR HUERGO

MDCSIM: A METHOD TO IDENTIFY SERVICES USING MASTER DATA, LOGICAL DATA MODELS AND ARTIFACT-CENTRIC MODELING APPROACH.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, como requisito parcial à obtenção do título de Mestre em Informática.

Aprovada em 27 de feveiro de 2014.

Prof. Paulo de Figueiredo Pires, D.Sc, PPGI/UFRJ (Orientador)

Prof. Flavia Coimbra Delicato, D.Sc, PPGI/UFRJ

Prof. Leonardo Gresta Paulino Murta, D.Sc, UFF

Aos meus pais, meus grandes mestres, ídolos e verdadeiros amigos.

Ao meu marido André, companheiro de todas as horas, por seu apoio e paciência nos momentos adversos, encorajando-me a seguir adiante.

Ao pequeno Lucas, meu afilhado tão amado, por compreender minhas ausências.

Agradecimentos

Fazer um mestrado é empreender uma longa jornada, que exige grande dedicação e empenho. Como todo projeto de longo prazo, nos deparamos com diversas dificuldades e é nessas horas que a amizade, o apoio e a compreensão dos que estão a nossa volta revelam-se essenciais para a realização deste sonho.

Desta forma, gostaria de agradecer, primeiramente, aos meus pais e ao meu afilhado Lucas. Eles, sem dúvida, foram as pessoas que mais sentiram a minha ausência. Contar com a compreensão e carinho deles foi fundamental para a conclusão dessa jornada.

Agradeço especialmente ao meu marido, pelo companheirismo e apoio incondicional que sempre apresenta em todos os momentos da minha vida.

Ao meu professor orientador, Paulo Pires, por acreditar no meu potencial e proporcionar tantas oportunidades de aprendizado durante as reuniões de orientação e aulas.

À professora Flavia Delicato pelas inestimáveis contribuições e paciência nas inúmeras revisões dos artigos escritos durante esse trabalho.

Aos demais familiares e amigos pela torcida, incentivo e pelas conversas bem humoradas que tanto contribuíram para amenizar o cansaço e desânimo.

Por fim, agradeço aos meus gerentes, Eugenio Pedrosa e Luis Antônio Araújo, pelo incentivo e liberação das horas necessárias para dedicação a este trabalho.

Resumo

HUERGO, Rosane Sfair. **MDCSIM:** a method to identify services using master data, logical data models and artifact-centric modeling approach. 2014. 163 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

A etapa de identificação de serviços é um dos maiores desafios das organizações para a implantação da Arquitetura Orientada a Serviços. A maioria dos métodos de identificação de serviços (SIMs) existentes na literatura depende da decomposição dos processos para descrever o domínio de negócio. No entanto, a identificação de serviços exige um alto nível de detalhe na documentação dos processos, somente encontrado em organizações maduras em gestão por processos. Além disso, a análise de artigos que realizam revisões dos SIMs revelou diversas lacunas, tais como avaliação das perspectivas de negócio e de TI, identificação tanto de serviços de negócio como técnicos, avaliação da qualidade dos serviços candidatos e possibilidade de configuração dos métodos. O presente trabalho preenche essas lacunas propondo um SIM configurável (MDCSIM) que utiliza dados mestre, modelos lógicos de dados (obtidos a partir das bases de dados das organizações) e técnica de modelagem orientada a artefato. Dados mestres (conceitos de informação chave para o negócio, usados por diversos processos, unidades organizacionais e aplicações) podem ser usados como um insumo alternativo aos processos de negócio. Modelos lógicos de dados auxiliam na identificação dos atributos que compõem os dados mestres e contribuem para a elicitação da perspectiva de TI, bem como para identificação de serviços técnicos. A técnica de modelagem orientada a artefato é usada em conjunto com dados mestres para elicitar a perspecitiva de negócio e identificar serviços de negócio. MDCSIM também utiliza métricas para avaliar atributos de qualidade dos serviços candidatos e sua utilização é suportada por uma ferramenta chamada MDCSIM plug-in implementada com base na Arquitetura Orientada a Modelos (MDA). MDA foi escolhida, porque permite transformar modelos lógicos de dados e modelos centrados em artefato em modelos que descrevem serviços candidatos. Por fim, uma avaliação inicial do MDCSIM foi realizada comparando-se portifólios de serviços identificados pelo MDCSIM e por outros dois SIMs centrados em dados.

Palavras-chave: Arquitetura Orientada a Serviços (SOA), Arquitetura Orientada a Modelos (MDA); Dados Mestre, Método de Identificação de Serviços (SIMs), Modelagem Orientada a Artefato, Modelo Lógico de Dados.

Abstract

HUERGO, Rosane Sfair. **MDCSIM:** a method to identify services using master data, logical data models and artifact-centric modeling approach. 2014. 163 f. Dissertation (Master in Informatics) – Instituto de Matemática, Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

Service identification is one of the biggest challenges in implementing a service-oriented architecture. Current service identification methods (SIMs) rely on business process descriptions to elicit business perspective. However, service identification requires a level of business process documentation only found on organizations mature on business process modeling. Besides this, current service identification methods have several drawbacks such as analysis of both business and IT domains, identification of both business and software services, service quality assessment and method configurability. In this context, this work fulfills the aforementioned drawbacks proposing a configurable service identification method (MDCSIM) that uses master data, logical data models (obtained from organizations databases) and artifact-centric modeling technique. Master data (core enterprise information concepts, needed across different business processes, organizational units and applications across the organization) can be used as alternative input to business process. The logical data models aid the identification of master data attributes and contributes to the elicitation of IT perspective and identification of software services. Artifact-centric modeling technique is used together with master data to elicit business perspective and identify business services. MDCSIM also uses some metrics to assess service quality attributes. MDCSIM is supported by a tool named MDCSIM plug-in. Such tool was implemented based on Model-driven architecture (MDA). MDA enables the transformation of data logical models and artifact-centric models into models that describes candidate services. Finally, an initial assessment of MDCSIM is provided by comparing the service portfolios identified using MDCSIM and using other two data-focused SIMs.

Keywords: Artifact-centric modeling; Logical data model; Master data; Model-driven architecture (MDA); Service identification method (SIM); Service-Oriented Architecture (SOA).

List of Figures

List of Tables

List of Acronyms

ATL	ATLAS transformation language
CIM	Computation Independent Model
CRUD	Create, read, update and delete
DFD	Data flow diagram
ESB	Enterprise service bus
IDE	Integrated Development Environment
IT	Information Technology
MDA	Model driven approach
MDCSIM	Master data centric service identification method
MDM	Master data management
MEP	Message exchange patterns
MOF	Meta-Object Facility
OCL	Object Constraint Language
OMG	Object Management Group
PIM	Platform Independent Model
PSM	Platform Specific Model
ROI	Return of investment
SIM	Service identification methods
SLA	Service level agreement
SOA	Service-oriented architecture
SODA	Service-oriented design aspect
UDDI	Universal description, discovery and integration
UML	Unified Modeling Language
XMI	XML Metadata Interchange

Contents

Introduction

This chapter provides an overview of this work, presenting the motivations and discussing the research problem and the hypothesis. Next, the solution and the contributions are presented. This chapter ends with the dissertation structure.

1.1 Motivation

Organizations exist in a competitive environment. Today's dynamic markets, the pressure to improve quality and productivity made the adaptability critical to the survival of organizations. The Information Technology (IT), as an important tool for organizations, must provide flexible and adaptable systems architectures to support business changes. Therefore, the construction of interoperable elements, that could be organized to quickly meet business needs as described in the Service-Oriented Architecture (SOA), became a promising alternative to be considered.

The process of service-oriented modeling and architectural design consists of three general steps: identification, specification and realization of services [Arsanjani 2005]. The identification step aims to determine which services are appropriate to be implemented in a service-oriented architecture. In this phase, services are named candidate services. During the specification step, the service architecture is designed and its interface, messages and events are detailed. Finally, the service is codified and tested in the realization step. Only the identification step is within the scope of this work.

The identification step is one of the major challenges in designing and implementing a service-oriented architecture [Demirkan et al. 2009]. This challenge consists in predicting which services an enterprise will eventually need, and in defining which functions should be part of each service. The set of functions within a service should be defined considering the purpose of the service, i.e. the value the service will deliver to the business and service quality attributes. Another important issue to service identification is that software development usually does not happen from scratch. Thus, IT perspective represented by existing software assets must also be considered in this phase. The analysis of how business and technical requirements affect service’s functions is complex, therefore demanding a method to support the identification phase.

During the last decades, several service identification methods (SIMs) were proposed, but there is no consensus on the “best method” or a predominant approach to identify candidate services. Service identification methods are categorized in three possible identification strategies: Top-down, Bottom-up and Meet in the middle [Erl, 2007]. Top-down strategy identifies services from a business perspective. This strategy commonly uses business process as inputs to identify services [Gu and Lago 2010]. This kind of approach has generally failed to deliver the value promised, because most processes that organizations execute are not enough documented to enable a good service identification [Ponnalagu and Narendra, 2008]. Therefore, this kind of approach requires organizations to be first engaged in modeling business processes before adopting it. Otherwise, service identification effort will be time-consuming and hence will not scale to large business processes [Bell, 2010].

Bottom-up strategy identifies services from an IT perspective. This strategy is driven by functional analysis of existing software assets (applications, services repositories, databases and legacy documentation) to evaluate which functions should be exposed as services. It also identifies new services that can fulfill implementation gaps or meet new requirements and technical specifications [Bell, 2010]. Bottom-up approaches are more successful at delivering services in short-term, but they usually identify fine-grained services that have limited reuse or do not have direct value to the business.

Finally, Meet in the middle is a hybrid strategy that supports the examination of both business and IT perspectives. This strategy is the most aligned with the enterprises reality, since it considers existing software assets and quickly delivers recognizable benefits without neglecting the fact that services are designed for reuse and must be aligned with the business context. Nevertheless, according to current service identification surveys [Boerner et al. 2009, Birkmeier et al. 2009, Kohlborn et al. 2009, Gu and Lago 2010, Cai et al. 2011, Vale et al. 2012, Zadeh et al. 2012], only few methods can be classified as Meet in the middle approaches [Chaari, 2007], [Zimmermann, 2004], [Shan and Hua, 2006], [Cho et. al., 2008], [Patig and Wesenberg, 2009], [Inaganti and Behara, 2007], [Erradi, Anand and Kulkarni, 2006], [Arsanjani et. al. 2008], [Fareghzadeh, 2008]. Such methods use business process documentation as inputs to elicit business perspective. Therefore, they suffer from the same drawback of top-down approaches, namely, they require detailed process descriptions as inputs to service identification. Since many organizations are not mature enough in business process modeling, it is important to investigate alternatives to elicit the business perspective.

In addition to the analysis of the aforementioned surveys, the systematic review presented in the Appendix A and the snowballing review presented in Chapter 4 were conducted in order to analyze in depth the drawbacks and gaps in SIM field. Besides the lack of Meet in the middle approaches, the following drawbacks in SIMs were identified: (i) identification of both business services and software services, (ii) configurability of the method depending on utilization constraints within the organizations (e.g. unavailability of an input, the need to apply the method to small domains or small enterprises), (iii) analysis of non-functional requirements, (iv) service candidate quality assessment and (v) consideration of economic aspects in the identification phase. The analysis of non-functional requirements and the analysis of economical aspects were not addressed in this work. The drawbacks description and the detailing of how they are fulfilled in this work are presented in Chapter 4.

1.2 Problem Statement

Considering the complexity of the identification phase in a SOA process and the aforementioned drawbacks, this dissertation aims to answer the following research question:

"How to identify both business services and software services from other input rather than business process, eliciting both business and IT perspectives and providing a configurable method that assesses service candidate quality attributes?"

1.3 Research Hypothesis

Although data is recognized as a vital enterprise asset used to drive business decisions [Mosley et al. 2009] and databases are the most abundant software assets in the organizations, the currently available service identification methods use other software artifacts as source code and legacy documentation as inputs. Data can also be used to elicit the business perspective, but not in the low level of abstraction as stored in databases, instead they are used in a high level of abstraction as master data.

Master data is any information considered to play a key role in the operation of a business. Such information encompasses top-level abstraction concepts comprised by data used across different business processes, organizational units, and information systems [Dreibelbis et al. 2008, Loshin 2008]. These characteristics confer to the master data a great potential of reuse and relevance to business. Master data can also be correlated with IT perspective by identifying database tables that represent them or applications that manipulate their attributes. Hence they are relevant inputs to service identification methods, aiding the identification of services from both business and IT perspectives.

The relationship between master data and SOA was already mentioned by [Dreibelbis et al. 2008] in their Master Data Management (MDM) approach. Accordingly to the authors, master data management can act as a SOA enabler supporting the provision of information as a service. Master data management processes deal with master data definition and governance of its structure and semantics, aiding the specification of information services. This approach emphasizes the possibility to use master data as a central element in the business modeling in order to identify services.

Services can be understood as a set of tasks that process some data. Hence, service identification can be accomplished by modeling tasks and business rules involved in master data transformation (during the master data lifecycle). Therefore, services operations can be modeled as a set of tasks that process master data. This modeling paradigm is used in the artifact-centric modeling technique [Nigam and Caswell 2003]. In such technique, the business perspective is represented by lifecycle of artifacts or business concepts. Thus, the identification of business concepts is the key activity of this modeling paradigm, especially when used to identify services. This issue is accomplished by using master data and logical data models as inputs for such technique. Master data promotes identification of services that act in business relevant concepts and the identification of master data attributes from logical data models (obtained from existing databases) decreases the subjectivity in master data identification.

The usage of master data and artifact-centric modeling technique in a SIM promotes the segregation of capabilities by master data and by the capability type (behavioral or maintenance). This characteristic contributes to the identification of services in layers and therefore supports the customization of the SIM to promote an in-depth analysis of a business domain identifying all layers of services, or a “lightweight” analysis of the business domain to identify only services related to master data maintenance. The identification of services in the scope of distinct business scopes is the horizontal evolution of the service portfolio. Vertical evolution concerns the identification of services of different layers.

The segregation of capabilities by master data and by type also contributes for the identification of services with more quality (more cohesive and with the “right” granularity). Service quality is an important issue to SIMs, since the quality of the service can compromise the achievement of SOA goals. The assessment of service quality is important to improve the quality of the identified services and of the SIM itself.

Encouraged by the aforementioned characteristics, the following research hypotheses were elaborated:

H1: *Master data can be used as an alternative input to business process to elicit business perspective and identify business services in a service identification method.*

H2: *The usage of logical data models obtained from existing databases to identify master data attributes promotes the elicitation of the IT perspective and the identification of software services.*

H3: *The usage of master data and of the artifact-centric modeling technique enables the configuration of the SIM in order to evolve service portfolio horizontally or vertically.*

H4: *The analysis of the quality attributes coupling, cohesion, granularity and entity convergence can be used to assess the overall service portfolio quality.*

The service quality attributes coupling, cohesion, granularity and entity convergence (the extent to which a service focuses on processing operations of a specific business entity) can be directly correlated with SOA goals. Therefore, these attributes can be used to assess the service portfolio quality as detailed in section 2.3 Service-Oriented Architecture Goals.

1.4 Proposed Solution Overview

Service identification should be wide-ranged and consider multiples perspectives comprising business and technological issues [Bell 2010]. Business perspective is related to business goals and requirements and is generally structured into processes or business models that express rules, constraints and dependencies [Erl 2008]. IT perspective is the automation of the business perspective organized into various technology solutions.

Considering the perspectives that should be comprised in service identification and the hypotheses described in section 1.3 Research Hypothesis, the solution is the proposal of the Master Data Centric Service Identification Method (MDCSIM) that uses master data, logical data models (derived from databases) and artifact-centric modeling technique to identify candidate services. Besides, MDCSIM will also use the metrics proposed in [Ma et al. 2009] in order to assess candidate services quality (section 7.5.2 Quantitative Comparison) and a tool named MDCSIM plug-in (section 6.2 MDCSIM plug-in) to support service identification. MDCSIM will address gaps as the analysis of both business and IT perspectives, identification of both business services (Task and Process services aims to fulfill business tasks requirements) and IT services (Entity services that deal with issues related to master data storage and ensure master data integrity), method configurability in order to deal with different levels of SOA maturity in organizations and assessment of services quality attributes.

Due to time constraints related to a master degree achievement, some aspects that are related to this research will be left out of the scope of this work:

- Other phases of service-oriented modeling and architectural design: Our solution supports only the service identification phase. Nevertheless, all outputs produced by MDCSIM can be used as inputs for the specification phase, facilitating the addition of the specification and realization steps, in the future, to cover the complete lifecycle of the service design. MDCSIM plug-in is also constructed based on a flexible architecture enabling its extension in order to generate service implementations as detailed in Chapter 6.
- Identification of Utility services: The proposed method does not clearly provide steps to support Utility services identification. However, the artifacts produced within MDCSIM can support Utility services identification, and also the identified candidate services can reuse existing Utility services in the organization;
- Master data reconciliation: Despite of the usage of logical data models as inputs to MDCSIM, semantic and schema reconciliation techniques are not addressed in this work. Whether the same master data appear in more than one database, those tables must be joined into one logical model before the application of MDCSIM.

- Use of stored procedures to identify services: Besides the use of logical models extracted from databases in order to identify services, MDCSIM does not provide guidelines to analyze stored procedures. Business rules are extracted only from master data lifecycle.
- Non-functional requirements elicitation and the analysis of economical aspects: Service identification surveys [Boerner and Goeken 2009], [Gu and Lago 2010], [Kohlborn et al. 2009] concluded that future methods should consider non-functional requirements elicitation and economical aspects related to service creation and maintenance. These aspects were not analyzed within MDCSIM and are opportunities for future researches in service identification field.
- Improvements in service quality metrics: This work do not intends to evaluate or improve service quality metrics. Few works of metrics applicable in the identification phase were published hitherto. These works are limited, because they do not consider differences between each type of service. Nevertheless, they are useful to compare sets of services, and provide an overview of the portfolio quality.

1.5 Research Contributions

This work intends to provide the following contributions:

- A critical analysis of the service identification methods: This work presents an overview of several service identification methods published from 2002 until June 2013. The more relevant aspects to this work are summarized in the Chapter 4 Related Work and the complete systematic survey is presented in Appendix A;
- A method to identify services from master data using artifact-centric modeling technique (MDCSIM): It describes a set of sequential steps, roles and artifacts that aid software architects to identify candidate services from master data and logical data models. The proposed method also address some open issues identified in the literature;

- A tool to aid candidate services' identification (MDCSIM plug-in): The set of patterns used to derivate candidate services in MDCSIM were implemented in a tool to support service identification. This tool allows the manipulation of business domain models with large number of entities thus, contributing for the scalability of MDCSIM.

1.6 Dissertation Structure

This dissertation is structured as follows:

- **Chapter 1 – Introduction:** provides an overview of the dissertation, describing the motivation, the research problem, the hypothesis, the solution, the contributions and the structure of the dissertation.
- **Chapter 2 – An Overview of Service-Oriented Architecture:** introduces the field of Service-Oriented Architecture. Firstly the definition of SOA and its elements were presented. Next, we provide the definition of service, and finally we explain SOA goals and how do they affect service identification phase.
- **Chapter 3 - An Overview of Artifact-centric Modeling:** provides an overview of artifact-centric modeling paradigm, explains the elements that comprise such modeling technique and depicts its advantages to service identification**.**
- **Chapter 4 – Related work:** presents the summary of the systematic survey and snowballing review accomplished. Identifies the research gaps in service identification methods, describing which of them will be addressed by the proposed solution and how they are accomplished.
- **Chapter 5 – MDCSIM – Master Data Centric Service Identification Method:** introduces the service identification method (MDCSIM) that is developed based on the hypothesis and on some of the research gaps presented in Chapter 4. We depict MDCSIM steps, artifacts, roles and a service layer model used to aid service identification.
- **Chapter 6 – Implementation:** provides an overview of the model-driven approach justifying its utilization to automate MDCSIM. Presents the MDCSIM plug-in,

describing the technologies used and identifies which steps of the MDCSIM are supported by the tool.

- **Chapter 7 – Proof of Concept:** depicts a proof of concept of the MDCSIM using a real business scenario. Then a qualitative and quantitative comparison of the MDCSIM with two other data-focused methods is presented in order to have an initial assessment of MDCSIM.
- **Chapter 8 – Conclusion:** presents some conclusions about this dissertation with a review of its contributions and identifies future work opportunities.

1.7 Chapter Summary

This chapter provided an overview of this dissertation by describing the research problem, the solution and the contributions. It also presented the structure of the dissertation and the purpose of each chapter.

2 An Overview of Service-Oriented Architecture

This chapter introduces the field of Service-Oriented Architecture. Firstly the definition of SOA and its four basis abstractions are presented. Next, we provide the definition of service, and finally we explain SOA goals and how do they affect service identification phase.

2.1 SOA Definition

Service-Oriented Architecture (SOA) is an architectural paradigm in which applications are constructed from interoperable elements named services. More specifically, SOA is concerned with the independent construction of business-aligned services that can be combined into meaningful, higher-level business processes and solutions within the context of the enterprise [Rosen et al. 2008]. Services can be reused by end-user applications or other coarse-grained services [Endrei et al. 2004]. SOA is also comprised by the structure that supports services construction, deployment and communication independently of the platform where services are running.

SOA is based on four key abstractions: application frontend, service, service repository, and service bus [Krafzig, Banke and Slama 2004]. Figure 1 presents these abstractions and the elements that comprise them.

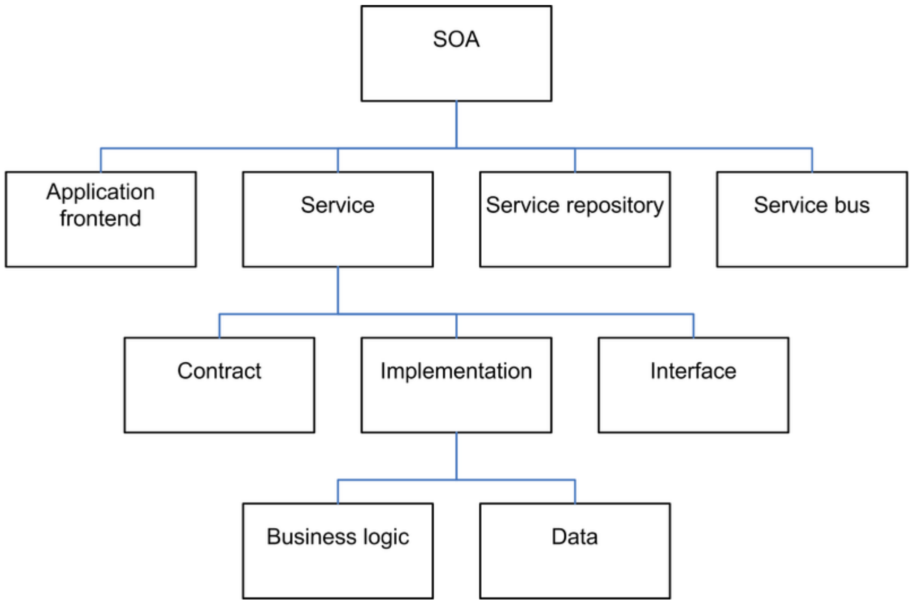

Figure 1: SOA abstractions. [Krafzig, Banke and Slama 2004]

The Application frontend constitutes the presentation layer. It initiates and controls all activities of the enterprise systems [Krafzig, Banke and Slama 2004]. Graphical user interfaces, such as a portal or a rich client are examples of applications frontends. Business users use application frontends in order to execute business process activities and monitor the execution of the services and processes. Services are comprised by a contract, an implementation and an interface. Services are detailed in the next section.

The service repository is a database that provides service metadata. Potential consumers access service repositories in order to identify which services can fulfill their needs. A service repository must provide facilities to discover services and information such as the service physical location, service provider, technical restrictions, security requirements and the service level. In SOA implementations using Web services, repositories are frequently constructed based on the Universal Description, Discovery and Integration (UDDI) pattern.

The Enterprise Service Bus (ESB) is a middleware that handles the communication between the several components of a SOA architecture. It decouples components from each other, allowing them to communicate without dependency on or knowledge of other components on the bus. The service bus can additionally offer features such as security, audit, transaction management, fault tolerance and support for various Message Exchange Patterns (MEPs). The ESB enables the integration between services, regardless of the technology and communication protocols that the services use.

2.2 Services

Services are the SOA building blocks. There are several definitions to services. From a business perspective, service can be defined as a discrete unit of business functionality [Rosen et al. 2008]. From the technical perspective a service can be defined as a software resource exposed and discovered via an interface [Arsanjani et al. 2008]. In this dissertation, we adopted [Bell 2010] definition "service is a software entity that offers business or technical capabilities". The capability offered depends on the type of service. For example, a Task service will typically support a business activity.

Services are comprised by an interface and an implementation. The interface specifies services capabilities. Service implementation is how the service provides its capabilities. The implementation may be based on existing applications, on orchestrating other services to combine their capabilities, on code written specifically for the service, or all of the above [Rosen et al. 2008]. Implementation should be hidden from consumers. Providers can change the implementation, nevertheless consumers will not be affected if the contract remains the same. This kind of encapsulation is a principle inherited from software component and from object oriented design. Nevertheless, differently from objects and components, services represent business functions and are designed to be used not only within a system scope, but within the whole organization or even inter organizations. Figure 2 summarizes the components of a service.

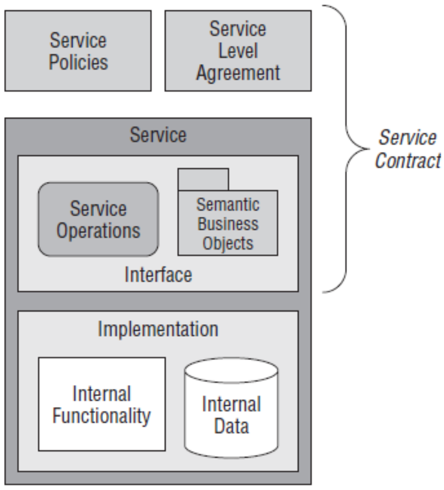

Figure 2: Service components. [Rosen et al. 2008]

The contract describes the service functionality and specifies how the consumers will interact with the providers. It comprises the interface and it may also describe quality attributes of the providers in order to define Service Level Agreements (SLA) [Erl 2007]. The functionality of the service is exposed by the service interface to clients that are connected to the service using a network. The service interface is divided into a behavior model and an information model [OASIS 2012]. The behavior model characterizes services operations, their temporal sequence and dependencies. The information model describes the syntax and semantics of the messages and data payloads, exception conditions and error handling in the event of faults.

The implementation fulfills the service contract. It consists of one or more artifacts such as programs, configuration data, and databases. The business logic that is encapsulated by a service is part of its implementation. The business logic and data are made available through service interfaces [Krafzig, Banke and Slama 2004].

2.3 Service-Oriented Architecture Goals

SOA was created as an answer to the growth of complexity of IT infrastructure. Legacy applications, typically developed with embedded business rules, duplicated code, tightly-coupled functions, and the demands for integrations made IT costs and deliver time high. SOA adoption can help organizations to increase organizational agility, increase the return of investment (ROI) and promote the alignment of business and IT domains [Erl 2007]. However, the real value of SOA only comes when reusable services are combined to create agile and flexible business processes [Rosen et al. 2008]. Therefore, a key step of developing service-based applications is to break required functionalities down into a set of services, and the challenge is to find an appropriated method to identify the "optimum" services. As business modeling and service interface designing are disconnected, the identified services do not always meet the business requirements and the quality attributes needed to satisfy SOA goals of increase organizational agility, increase ROI and promote the alignment of business and IT domains.

Organization agility is about being efficient to answer to changes. Services are reusable, interoperable, standardized and autonomous elements. These characteristics confer to services the ability to be easily composed in order to answer to new business requirements. As the number of services grows, time and cost of delivering solutions decrease, because more functions can be realized by service reuse. IT resources are also optimized by eliminating redundancy, consequently increasing the ROI. SOA also promotes the alignment of business and IT domains by using a design paradigm in several abstraction levels. Services can be organized in layers that encapsulate and represent business models, thus aligning automation technology and business intelligence [Erl 2007]. Figure 3 presents SOA goals, the ways to achieve them and the related service quality attributes.

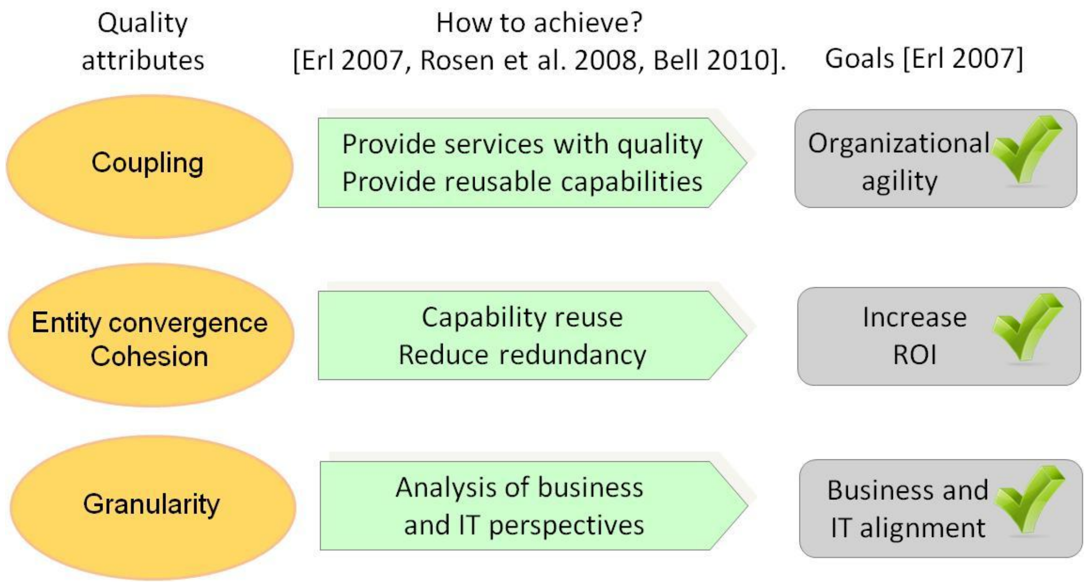

Figure 3: SOA goals and related service quality attributes.

Figure 3 shows the service quality attribute more related to each aforementioned goal. Nevertheless, a quality attribute can be related to more than one goal. Business agility can be favored by loosely-coupled services, because a service with low coupling is self-contained and independent, thus is more easily reused and composed to support new requirements. Cohesion and entity convergence (the extent to which a service focuses on processing operations of a specific business entity) cover the ROI increasing. Cohesion increases the comprehension of identified services, thereby simplifying reuse, maintenance and future enhancements. Services that encapsulate all actions of a business entity are more reusable, avoiding redundancy. Granularity covers the business alignment goal, because coarse-grained services offer rich functionality and have larger contribution to business processes.

Manage quality attributes is not simple because increasing one quality attribute can result in a negative impact in other attributes. For instance, fine-grained services are more reusable than coarse-grained services. Nevertheless, the utilization of fine-grained services can lead to a poor performance of the application, because of increasing communication trips. Therefore, a SIM must consider the purpose of the service candidate and promote services with quality in order to support SOA goals. The purpose of the service refers to the functionalities offered by the service in terms of service types (for instance, a service that provides CRUD operations is different from that one that provides infrastructure functions) and service quality refers to the balance of granularity, coupling, cohesion and entity convergence values of the identified service portfolio.

2.4 Chapter Summary

This chapter provided an overview of Service-Oriented Architecture presenting its definition and the four basis abstractions. The concept of service adopted in this dissertation was also explained. We concluded discussing SOA goals, the service quality attributes related to them and how do they affect the identification phase.

3 An Overview of Artifact-centric Modeling

This chapter provides an overview of artifact-centric modeling paradigm, describing its motivations. This chapter also explains the elements that comprise such modeling technique and depicts its advantages to service identification phase.

3.1 Artifact-Centric Modeling Technique

The activity-centric modeling paradigm models all the activities in the process and defines a control flow and a data flow over these activities. As the processes grow in size and complexity, it becomes increasingly difficult to understand the business behavior using these models [Alonso et al. 1997]. The approach to deal with this complexity is to use a hierarchical representation of business processes. Static and hierarchical representations of business processes are good for documenting business operations. Nevertheless, these representations do not lead to in-depth analysis and prediction of the behavior of the systems under dynamic conditions [Kumaran et al. 2008]. In response to this situation, another process modeling paradigm, in which business processes are modeled as intersecting lifecycles of artifacts, has been first proposed by IBM [Nigam and Caswell 2003].

The artifact-centric modeling approach is comprised by three steps: (i) identification of artifacts, (ii) development of the artifact informational model and (iii) creation of the artifacts lifecycles models. In the first step, key artifacts managed by the business process are identified. Discovering the “right” business artifacts is the key issue in this paradigm, because artifacts are the building blocks used to describe the business domain. Examples of artifacts include Customers, Payments and Invoices. In the second step, a detailed logical model of the data needed about each class of artifact and the relationships between artifacts is developed. In the last step, a workflow specification that represents the artifact lifecycle is modeled. The workflow specification can be optimized and mapped into a physical implementation in order to derive applications.

The artifact-centric modeling approach presents an attractive alternative as it helps to analyze and predict systems behavior using the lifecycle models of a few artifacts in a flat structure. An artifact model expressed in business-level terms can be automatically mapped onto a workflow engine to create a deployed system [Cohn and Hull 2009]. Besides, this paradigm comprises the informational model, which is a crucial aspect of virtually all software design approaches. Artifacts also provide a natural modularity and componentization of business processes. Each module implements the behavior of an artifact and manages the artifacts associated with that artifact. This approach to modularization leads to a new way to implement a process using SOA.

3.2 Artifact-Centric Modeling Elements

Differently from traditional activity-centric modeling paradigms, which often consider process modeling and data modeling separately, artifact-centric modeling technique defines business processes in terms of interacting artifacts. Artifacts are business-relevant objects that are created, evolved, and often archived as they pass through a business task. Artifacts combine both data aspects and process aspects into a holistic unit, and serve as building blocks from which business models and processes are constructed [Nigam and Caswell 2003]. Each business artifact is characterized by an information model and a lifecycle model.

The informational model describes artifacts attributes, artifact semantic and relationships between artifacts. Attributes store all the data needed for the lifecycle execution (data created, updated and deleted by services). Artifacts should be self-contained, in other words, all data needed by the artifact is present in the artifact. The lifecycle model describes all the stages in the possible evolution of the artifact, from inception to final disposition or archiving. Task that can be executed upon an artifact and when they can be executed are also depicted in the lifecycle. Lifecycles are usually described by using models based on state machine diagrams [Bhattacharya et al. 2009], [Nigam and Caswell 2003] and [Yongchareon et al. 2012], petri nets [Qi and Huifang 2011] and workflows [Li 2009].

A service can be understood as a set of tasks that perform operations on some artifact(s), where these operations should reflect steps of progress towards the business goal. A service should have exclusive control over the involved artifacts when making these changes. Services can maintain artifacts, or change the artifact state obeying the appropriate conditions. In a correlation with the traditional activity-centric modeling paradigm, processes are composed by pieces of lifecycles from several artifacts. Two artifacts are connected if their lifecycles share at least one task. For instance, one artifact can create a new instance of another artifact as part of a task (the purchase artifact can create a payment artifact in a sales process).

3.3 Artifact-Centric Modeling Advantages

Artifact-centric modeling paradigm has several advantages to service identification. Firstly, artifact-centric technique merges process view with data view. In traditional activity-centric modeling approaches, data models and relationships between tasks and data are not clearly defined in the process model (information is treated purely as activities' inputs or outputs). The data model and task model are defined independently and their relation may not be coherently captured in the activity-centric model. Coherency between task and data models is important in order to identify services with quality, since the data handled by a service can affect its cohesion and entity convergence. Eliciting data dependencies in the identification phase is also important to have a complete vision of the service capability, thus producing better inputs to the service specification phase.

Another advantage of this paradigm is the possibility to slice business scope (focusing in a few number of entities) and obtain services with less modeling effort than in the traditional activity-centric modeling style. An intuitive explanation can be derived from the Pareto principle which states that, for many events, 80% of the effects come from 20% of the causes. When applied to business process analysis, we observe that a few information entities serve as key drivers of the flow of most activities [Kumaran et al. 2008].

Artifact-centric approaches naturally lend themselves well to both object-orientation and service-orientation design principles, because they focus on the design of both business artifacts involved in a process and services performing operations on such artifacts. This nature leads artifact-centric models to provide high levels of flexibility, extensibility, and reusability [Yongchareon et al. 2012]. The focus on business artifacts and services also favors the segregation of the business domain by types of services accordingly to the nature of the capability owned by them. Capabilities related to the artifact maintenance can be gathered into Data services and capabilities related to the artifact behavior into Task or Process services. This segregation into service layers contributes to more cohesive services, since their responsibilities are well defined and to a configurable method that can identify services prioritizing layers of service.

3.4 Chapter Summary

This chapter provided an overview of artifact-centric modeling paradigm, explained the elements that comprise such modeling technique. The advantages of this technique to service identification were discussed, especially the service-oriented nature that leads to business models that contributes to more cohesive services.

4 Related Work

This chapter describes the methodology used to conduct the literature review, presenting a summary of the systematic survey and of the snowballing search. Both techniques were applied in order to ensure high levels of recall for the literature review. The research gaps in service identification methods are also discussed in the chapter and we point out which of them will be addressed by the proposed solution and how they are accomplished.

4.1 Literature Review Methodology

Service-Oriented Architecture needs a well-defined process to be implanted efficiently and reduce its associated risks [Erl 2007]. MDCSIM can use best practices of the existing approaches, but must also promote improvements. In order to provide an in depth analysis of the current SIMs and identify further opportunities for improvements in this field a systematic survey was undertaken based on the guidelines of systematic reviews suggested in [Kitchenham 2009]. Those guidelines were adopted because they provide a way to identify related work in a thorough and unbiased manner. However, our survey cannot be classified as a secondary study as systematic reviews, because it does not aim to provide statistical analysis of the SIMs.

The complete survey is presented in Appendix A. The results of this survey can be used by practitioners or researchers that want to adopt or investigate service-oriented methods. The survey uses a classification scheme based on OASIS' reference architecture framework for SOA [OASIS, 2012] in order to compare SIMs. We adopted a reference architecture as the basis to create our classification scheme, because these architectures describe the various characteristics of a reference SOA environment pointed out as relevant to the industry to assist SOA understanding and adoption. Besides this, the characteristics within the reference architecture are not disconnected of the criteria proposed by other SIMs surveys. The technique used by the SIM to identify candidate services was also included in our classification scheme. The classification scheme provided a complete overview of the

SIMs, nevertheless, in this chapter, only the identified gaps and drawbacks in service identification field are discussed.

In addition to the systematic survey, we conducted a snowballing review. Snowballing search technique was used in conjunction with the systematic survey to ensure high levels of recall for the literature review. In the snowballing technique, a group of relevant references is chosen as the starting point. References to these starting points and publications of the authors are identified. Then, the results are analyzed in order to identify SIMs not retrieved previously within the survey. As the starting point for the snowballing three SIMs were chosen [Baghdadi 2006], [Strosnider et al. 2008] and [Yun et al. 2009]. These SIMs were chosen because they have similarities with MDCSIM. [Baghdadi 2006] and [Yun et al. 2009] are data-focused SIMs and [Strosnider et al. 2008] applies artifact-centric modeling technique. We used the same search period of the survey (from 2002 to June 2013) to perform the snowballing and we identified one more SIM [Al Belushi and Baghdadi, 2007]. This SIM uses legacy applications as input to identify services.

4.2 Search Results and SIMs Drawbacks

The survey returned 105 technical papers that reported service identification: 7 surveys [Birkmeier et al. 2009, Boerner and Goeken 2009, Kohlborn et al. 2009, Gu and Lago 2010, Cai et al. 2011, Zadeh et al. 2012, Vale et al. 2012] and 98 methods. Surveys [Birkmeier et al. 2009, Boerner and Goeken 2009, Kohlborn et al. 2009, Cai et al. 2011] concluded there is a lack of systematic methods that comprises the identification and analysis of services on both the business and the technical level. [Boerner and Goeken 2009], [Cai et al. 2011], [Zadeh et al. 2012] proposed that future methods have to be configurable depending on the utilization constraints within the organizations (e.g. unavailability of an input, the need to apply the method to small domains or small enterprises). [Kohlborn et al. 2009], [Gu and Lago 2010] suggested that SIMs should analyze non-functional requirements. [Birkmeier et al. 2009] remarked that methods do not assess service candidate quality, nor provide means to guarantee it. Finally, [Boerner and Goeken 2009] suggested that economic aspects should be considered in the identification phase.

Analyzing the aforementioned conclusions, the results of our survey and the results of the snowballing search, the following research drawbacks in SIMs were identified: (i) analysis of both business and technical perspectives, (ii) identification of both business and

IT services, (iii) elicitation of non-functional requirements, (iv) analysis of economical aspects, (v) method configurability and (vi) service candidate quality assessment. These drawbacks are described as follows:

- Analysis of both business and technical perspectives: Service identification should be wide-ranged and consider multiples perspectives comprising business and technological issues [Bell 2010]. Business perspective is related to business goals and requirements and is generally structured into processes or business models that express rules, constraints and dependencies [Erl 2008]. IT perspective concerns the automation of the business perspective organized into various technology solutions. The analysis of business perspective is important to identify services that deliver direct value to business and promote business agility which is one of the SOA goals [Erl 2007]. On the other hand, the analysis of IT perspective promotes an alignment with the existing IT assets and helps to identify the resources (data, application functions and existing services) necessary to realize service capabilities, thus providing a better input to the specification phase. These two perspectives are complementary. The analysis of only one perspective can compromise the achievement of SOA goals, or lead to services that are not suitable to the organizations reality. For example, analyzing only the business perspective can lead to the identification of services that fulfill business requirements, but are very expensive to be developed and integrated with the existing software assets or IT architecture. Another example is the identification of services by functional analysis of existing software assets, that might lead to services with a limited range of reuse, compromising the SOA goals of promoting business agility and increasing the ROI. Few SIMs analyze both perspectives (12%). The majority of the SIMs (57%) analyzes only the business perspective and 31% analyzes only IT perspective.
- Identification of both business and IT services: This drawback is related to the type of service candidate identified by the SIM. There are several different classifications proposed to define service types from various viewpoints. The

service classification is often defined based on the value delivered by a service from business and IT perspective or alternatively based on composition layers. The difference in definitions of service types are based on the scope of each method. Some approaches only provide guidelines to derive services in general, others distinguish between basic service types and a few provide a classification scheme with descriptions of services goals. [Cai et al. 2011], [Wang, Xu and Zhan 2005] separate services into business services and software services (IT services). A business service is an abstraction of one or more business functions or business goals. Software services expose part of an application, perform CRUD operations on databases, or perform functions not related to the business, but required to support business services. [Souza et al. 2009], [Weigand et al. 2009] segregate services that perform CRUD operations in databases from the IT service type and name them Data service (or Informational service). Finally, [Alahmari, De Roure and Zaluska 2010], [Dwivedi and Kulkarni 2008], [Erradi, Anand and Kulkarni 2006], [Erradi, Kulkarni and Maheshwari 2007], [Fareghzadeh 2008], [Huayou et al. 2009] [Klose, Knackstedt and Beverungen 2007], [Kohlborn et al. 2009], [Kohlmann and Alt 2007], [Liu et al. 2011], [Rosen et al. 2008], [Shirazi, Fareghzadeh and Seyyedi 2009], [Strosnider et al. 2008] use classifications with 3 layers or more, comprising layers of composite types of services. Composite services are responsible for orchestrate services from other layers in order to fulfill a requirement. Regardless of this classification diversity, we concluded that the classification based on the value delivered by a service is more comprehensive (having correspondence with all SIMs) and is the most suitable to analyze whether the SIM identifies services that contribute to promote business and IT alignment, regardless of the service responsibility. In this scenario, we identified that 47% of the SIMs identify only business services, 27% identify both types of services and 26% identify only IT services.

- Elicitation of non-functional requirements: Business requirements are not the only requirements that originate candidate services or affect candidate services' capabilities. Non-functional requirements (or technical requirements) might also reveal constraints, conditions of use of a service, or even additional candidate services that support the accomplishment of non-functional requirements. For instance, non-functional requirements of security can originate services to authenticate users, to control the access to specific functionalities, or to limit access to some services depending on the user's profile. Besides this, conflicting non-functional requirements might cause service redesign [Gu and Lago 2010] impacting the whole process of service-oriented modelling and design. SIMs elicit non-functional requirements by using service oriented design aspect technique [Bao et al. 2010], [Mosser et al. 2011], [Souza et al. 2011]. This technique identifies services based on the decomposition of interactions, concerns and features into aspects and composing them according to requirements. [Dinh and Nguyen-Ngoc 2010] elicite constraints and legal issues of the information exchange between organizations. [Samavi, Yu and Topaloglou 2008] Identifies non-functional requirements as services soft goals. Finally, [Andersson, Johannesson and Zdravkovic 2009], [Asadi et al. 2009], [Asadi et al. 2011], [Canora et al. 2008], [Erradi,Anand and Kulkarni 2006], [Ilayperuma and Zdravkovic 2010], [Kazemi et al. 2011a], [Lee, Muthig and Naab 2010], [Kaabi, Souveyet and Rolland 2004], [Kang, Song and Baik 2008], [Kim and Doh 2007], [Medeiros, Almeida and de Lemos Meira 2010], [Schmidt 2011], [Weigand 2011] have an activity to elicit non-functional requirements or receive the requirements as inputs.

- Analysis of economical aspects: Technically-driven implementations often fail to be profitable [Boerner and Goeken 2009]. Deployment of certain services can promote business advantages such as reducing the time-to-market of a new product, decreasing maintenance and operation costs by reducing IT complexity and decreasing vendor dependency. These analyses are important to economically advantageous implementation of business services. The customer has to be willing to pay for the result of a process, i.e. services should always increase the value of a product. The degree of value creation depends on an effective and efficient combination and coordination of resources [Boerner and Goeken 2009].

- Method configurability: SOA implementation is a costly process in terms of time, financial and resources [Zadeh et al. 2012]. Many organizations do not involve themselves in implementing SOA, because of the costs associated with the service-oriented modeling process, especially in the service identification phase. Depending on the SIM adopted, several efforts of business process documentation or application reengineering must be taken before adopting it. In order to facilitate SOA adoption, identification methods should be reconfigurable to be compatible to specific situations. For instance, methods should preview the unavailability of an input and suggest alternatives inputs. Methods should enable customization to provide a 'lightweight' version for small domains or cases when fast results are required.
- Service quality attributes assessment: Most researchers agree on the importance of metrics to improve the quality of the identified services and of the SIM itself. Quality is dependent of the stakeholders’ requirements, but some general service quality attributes can be identified in a SOA context. Erl [Erl 2007] emphasizes that the basic software quality design principles of low coupling and high cohesion should be observed during all service creation cycle. Service granularity is also pointed out as a quality attribute, because the granularity level of a service can affect its capabilities, performance, reusability and coupling. SIMs deal with service candidate quality by using metrics of coupling [Kazemi et al. 2011a], [Medeiros, Almeida and de Lemos Meira 2010], [Bianchini et al. 2013], [Bianchini et al. 2009], cohesion [Kazemi et al. 2011a], [Medeiros, Almeida and de Lemos Meira 2010], [Bianchini et al. 2013], [Abdelkader, Malki and Benslimane 2013], [Bianchini et al. 2009], granularity [Kazemi et al. 2011a], [Kim and Doh 2012], modularity [Kazemi et al. 2011b], [Li and Tahvildari 2006], reusability using the semantic distance between features [Kang, Song and Baik 2008] and QOS by estimating a weight to execute activities [Menascé, Casalicchio and Dubey 2008]. SIMs usually do

not assess service quality, nor do any effort to improve identified candidates. Regardless of the quality attribute adopted, SIMs should provide means to assess service candidate quality. Services with low quality can affect the reuse compromising the achievement of the SOA goals of promoting business agility and improving the ROI.

Table 1 summarizes the percentage of the identified SIMs that addressed each aforementioned drawback.

Table 1: Service identification drawback summary

SIM drawbacks	**SIMs that addressed the drawback (%)**
Analysis of both business and technical perspectives	12
Identification of both business and IT services	27
Elicitation of non-functional requirements	19
Analysis of economical aspects	0
Method configurability	0
Service quality attributes assessment	10

Besides the analysis of each drawback fulfillment individually, we also identified that none of the aforementioned SIMs addressed both the drawbacks *Analysis of both business and technical perspectives*, *Identification of both business and IT services, Elicitation of non-functional requirements* and *Service quality attributes assessment.*

4.3 Drawbacks addressed by MDCSIM

MDCSIM addresses the drawbacks *Analysis of both business and technical perspectives*, *Identification of both business and IT services* and *Service quality attributes assessment*. The drawback *Analysis of both business and technical perspectives* is accomplished by using Master data and their lifecycle to describe the business perspective. Master data lifecycle depicts tasks and business rules involved in master data transformation. Each master data transformation can be correlated with several business process activities in the traditional activity-centric modeling paradigm, thus describing the business domain. Master data is composed by a set of attributes that describe it. For example, the attributes Value, Payment date and Currency are part of the master data

Payment. Master data attributes and the tables where they are stored can be identified from logical data models, complementing the business perspective identified previously with the IT perspective. Figure 4 depicts the inputs used by MDCSIM and the perspectives described by them.

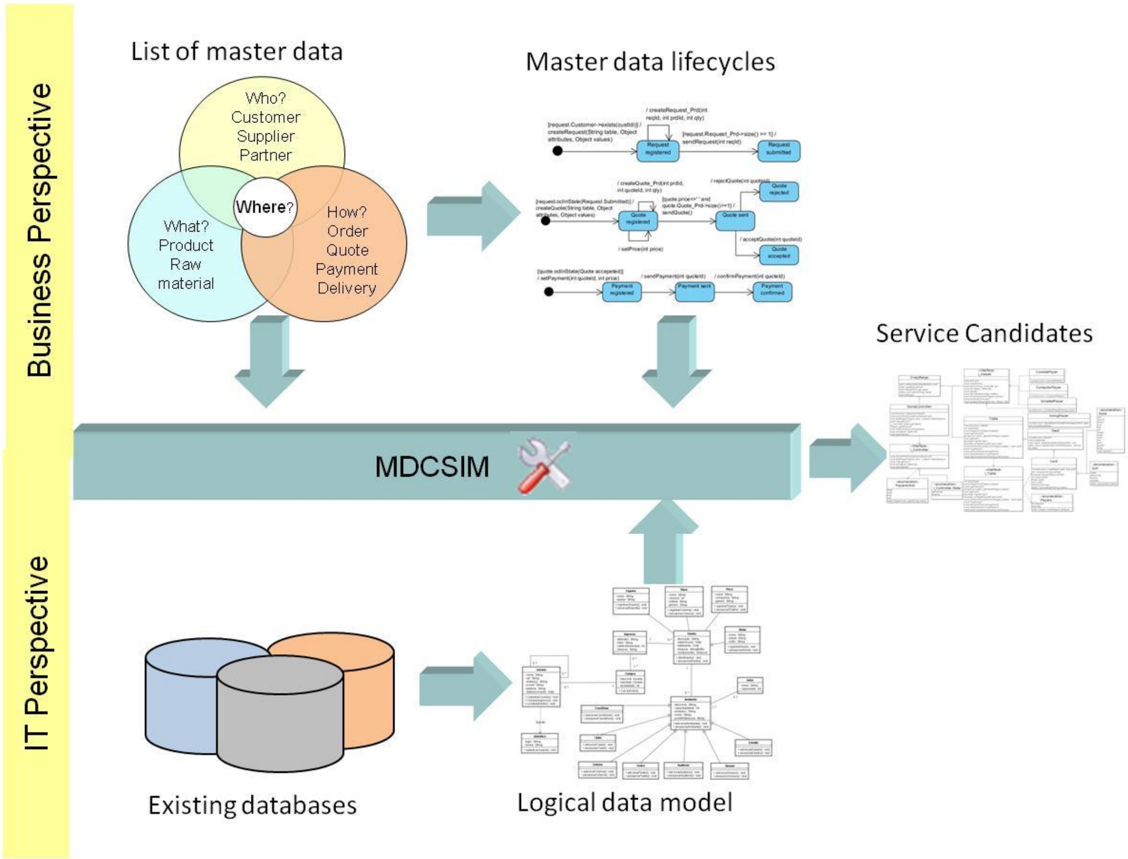

Figure 4: Inputs and perspectives elicited by MDCSIM.

The analysis of both perspectives is also related with the identification of business services and IT services. Business services are identified from tasks and business rules within master data lifecycles and IT services from the logical data model obtained from the existing databases. MDCSIM uses a service layer model in order to aid service identification. These layers can be organized accordingly to the value delivered to business as described in section 5.1 Service Layers.

The drawback *Service quality attributes assessment* is addressed by the evaluation of the granularity, coupling, cohesion and entity convergence quality attributes of the identified service portfolio. These quality attributes were chosen because they are related with SOA goals of promoting business agility, increasing the ROI, and promoting business alignment as mentioned in section 2.3 Service-Oriented Architecture Goals. The evaluation of the aforementioned quality attributes is accomplished by using metrics proposed in [Ma et al. 2009]. These metrics are detailed in Chapter 7 Proof of Concept.

The drawback *Method configurability* is addressed partially, because MDCSIM enables a customization of its steps in order to provide a “lightweight” version, but does not enable utilization of alternatives inputs when the list of master data or the master data lifecycle models are not available. The complete version of MDCSIM supports the identification of services for distinct business scopes, which is the horizontal evolution of the service portfolio and the identification of services of different layers, which corresponds to the vertical evolution. The “lightweight” MDCSIM version enables short iterations in order to identify first a set of services that perform CRUD operations in databases (Entity services) for the prioritized business scopes (horizontal evolution). The “lightweight” version can be used by organizations with low maturity in SOA or to quickly deliver results.

Figure 5 shows the scope of service identification for the “lightweight” MDCSIM and the ways a service portfolio can evolve.

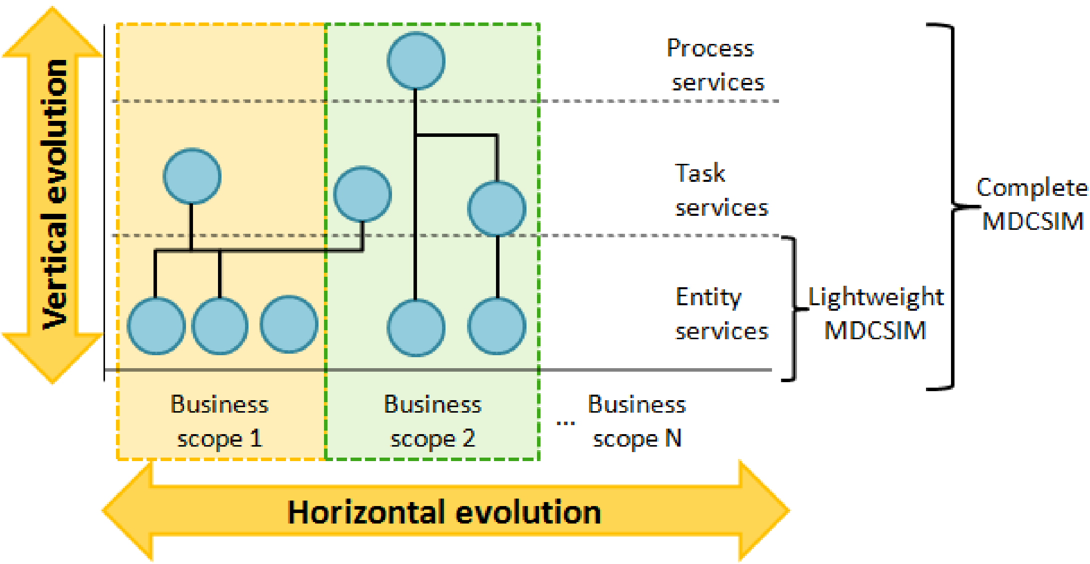

Figure 5: Service identification in lightwieight MDCSIM and portfolio evolution.

Finally the drawbacks *Elicitation of non-functional requirements* and *Analysis of economical aspects* are not addressed in MDCSIM due to time constraints related to a master degree achievement. The elicitation of non-functional requirements can be done in the beginning of the specification phase. The analysis of economical aspects is very complex to accomplish in the identification phase when service project is not ready. Table 2 summarizes the drawbacks addressed by MDCSIM.

Table 2: Drawbacks addressed by MDCSIM.

SIM drawbacks	**Drawback addressed by MDCSIM**
Analysis of both business and technical perspectives	Yes
Identification of both business and IT services	Yes
Elicitation of non-functional requirements	No
Analysis of economical aspects	No
Method configurability	Partial
Service quality attributes assessment	Yes

4.4 Chapter Summary

This chapter described the methodology used to conduct the literature review. A systematic survey and a snowballing search were conducted and 106 technical papers (105 in the systematic survey and 1 in the snowballing search) that reported service identification were identified. Among the aforementioned technical papers, 7 are surveys and 99 are SIMs. Six drawbacks were identified from an analysis of the aforementioned surveys' conclusions and of the identified SIMs. The drawbacks addressed by MDCSIM were pointed out and an overview of how each drawback is achieved in MDCSIM was provided.

5 MDCSIM: Master Data-Centric Service Identification Method

This chapter introduces MDCSIM and depicts its steps, artifacts, roles and a service layer model used to aid service identification. MDCSIM uses as inputs a list of master data, a logical data model and artifact-centered modeling. The method can be classified as a Meet in the middle approach, because it elicits both business and IT perspectives. The business perspective is elicited from the master data and the artifact-centric models and the IT perspective is obtained from the extraction of logical data models from existing databases.

5.1 Service Layers

In order to make services identification easier, services can be categorized into abstraction layers [Alahmari et al. 2010]. Layers are related to the service reuse potential and to the logic encapsulated by them. The organization in layers is often used to guarantee the definition of services with "right" granularity, cohesion and it is also a natural composition hierarchy. This work uses the service layer model proposed by [Erl 2007], [Kohlborn et al. 2009] to guide service identification. The model defines four layers of services: Utility services, Entity services, Task services and Process services. Some layers' definitions have a great correlation with business processes and activities, referring to the traditional activity-centric process modeling paradigm. Thus, these definitions were extended in this work to comprise also services identified from the artifact-centric process modeling paradigm using state transitions, as follows.

Utility service is not related to business logic. It provides shared functions, such as authentication, encryption, logging and event handling, for other services [Erl 2007], [Kohlborn et al. 2009]. Entity service is a business service whose functional scope is related to the functional context of one business entity [Krafzig, Banke and Slama, 2004], [Erl 2007], [Kohlborn et al. 2009]. In MDCSIM, an Entity service manipulates one master data and ensures data completeness. Task service is modeled for specific processes to meet immediate requirements of the organization and therefore contains specific business logic [Erl 2007], [Kohlborn et al. 2009]. It represents the behavior of a business entity and implements the transitions in a master data lifecycle [Strosnider et al. 2008], [Ponnalagu and Narendra, 2008]. Finally, Process service introduces a level of abstraction that alleviates the need of Entity, Task, and Utility services to manage interaction details required to ensure that service operations are executed in a specific sequence. It represents a workflow of set of states of one entity or the coordination of a process resultant from the state transitions of several entities. The four layers, their granularity and potential for reuse are presented in Figure 6.

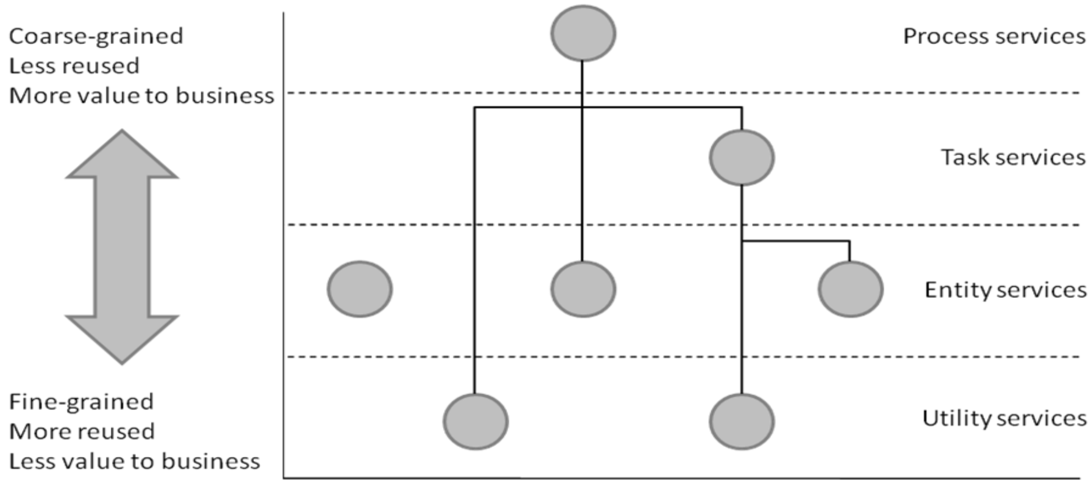

Figure 6: Service layers

5.2 Service Identification Steps

MDCSIM intends to support the identification of Entity services, Task services and Process services. Nevertheless, Utility services can also be identified by analyzing cross-cutting concerns not related to business in Entity services, Task services and Process services and exposing them as Utility services. The identification of Entity services' CRUD operations uses a subset of the operations patterns proposed in [Baghdadi 2006] as described in Section 5.2.4. MDCSIM also uses data models as [Baghdadi 2006]. Nevertheless, these models are correlated with master data to aid the identification of business relevant concepts. Some guidelines to aid master data identification were also provided in the step 5.2.2 Master Data Identification as part of our proposal. The UML activity diagram of Figure 7 presents the steps of the proposed method, which are detailed in the next sections:

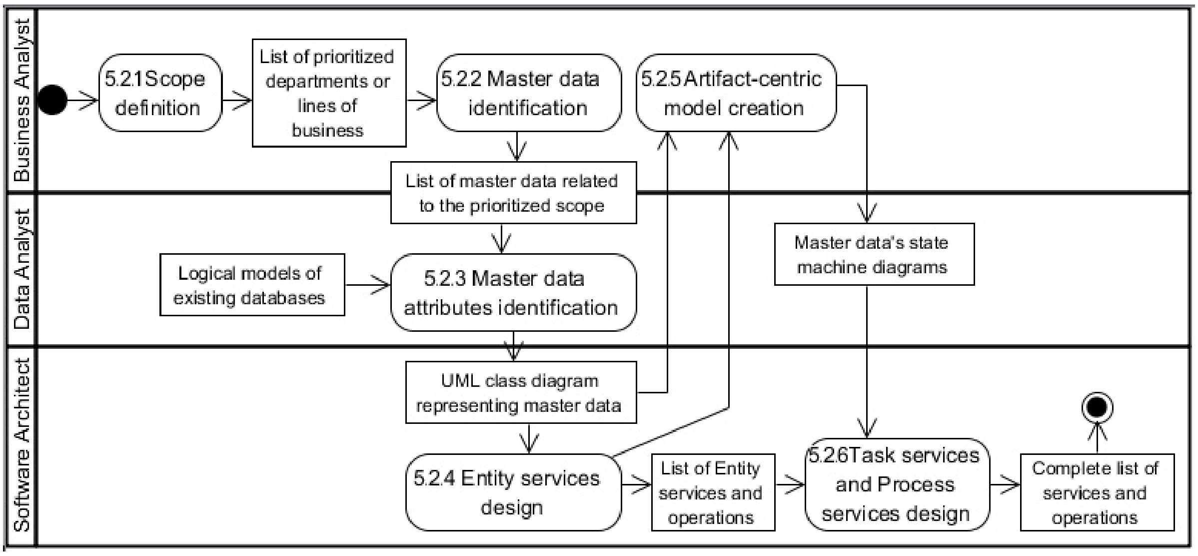

Figure 7: Service identification method

5.2.1 Scope Definition

Service identification cannot be applied in all organizational units at the same time. Instead, it makes sense to prioritize the analysis at core lines of business or departments, in which a number of visible benefits and the potential of services reuse can be estimated. Thus, the first step of the proposed method is the definition of the analysis scope. Scope definition step is conducted by the Business Analyst and it produces as output a prioritized list of lines of business or departments.

The scope can be delimited by choosing a line of business or a department. Within large organizations, lines of business frequently act as sub-organizations. Therefore, it could still be necessary to restrict the scope to a department. Subsequent iterations of the method should be done in lines of business or departments that interact with the ones previously prioritized, because several master data will be shared between them. Therefore, Entity services already identified will be potentially reused.

It is important to mention that MDCSIM supports the evolution of the service portfolio both horizontally and vertically. Horizontal evolution is related to the identification of services in the scope of distinct lines of business or departments. Vertical evolution concerns the identification of services of different layers, from the fine-grained to the coarse-grained, accordingly to the aforementioned layers. This is an important feature since the method can be customized to support service identification in organizations with different levels of SOA maturity. Organizations can start with the identification of Entity services of several lines of business or departments executing the steps 5.2.1 to 5.2.4. Later, as the maturity in SOA increases, the identification of Task and Process services can be accomplished by executing steps 5.2.5 and 5.2.6.

5.2.2 Master Data Identification

The next step is the elicitation of a set of master data that are relevant to the scope prioritized in the previous step. Common organization master data can be elicited from the following domains [Dreibelbis et al. 2008]:

- Parties: Roles played by persons or organizations, such as patients, suppliers and employees.
- Things: Products, services or other items used in the production lifecycle (from development, through manufacturing, sale and delivery).
- Account: Addresses how a Party is related to a Thing. Cost centers and contracts are examples of master data in the account domain.
- Location: Usually is associated with the aforementioned domains. Geographic position is an example of location master data.

Master data can be identified by answering the following questions: “Who?”, “What?”, “How?” and “Where?”. Question “Who?” addresses the Party domain. The question “What?” addresses the Things domain. The question “How?” addresses the Account domain and “Where?” the Location domain.

The step of master data identification is conducted by the Business Analyst with participation of the Data Analyst. The output is a list of master data related to a business scope.

5.2.3 Master Data attributes identification

This step aims to identify master data attributes and relationships by correlating the master data identified in the previous step with the logical data model that describes them. This correlation promotes a systematic way to identify which attributes should compose a master data and can also promote the identification of master data that were not identified previously. The logical data models can be gathered by using relational database reverse engineering [Chiang et al. 1994], Latent Semantic Indexing (LSI) technique [Bradford 2005] domain models [Guizzardi 2005] or master data catalogs within the organization. Logical models are presented as UML class diagrams. The logical models must be manually analyzed by the Business Analyst and the Data Analyst in order to identify which classes describe the master data identified in the step 5.2.2. Each master data should be described by one class. Classes that describe master data are marked with the stereotype “MasterData” by the Data Analyst. The logical model must be normalized to derive classes whose attributes were aggregated using the functional dependency constraints. Therefore, the master data are comprised by a cohesive set of attributes. The master data cohesion affects the resulting service cohesion, because a service will act in a set of attributes defined by each master data.

This step is conducted by the Data Analyst with participation of the Business Analyst. The output is a stereotyped UML class diagram representing the master data, their attributes and the classes that have a relationship with them.

5.2.4 Entity Services Design

After identifying master data and their attributes, a set of Entity services and basic operations can be designed. Basic operations are CRUD operations that act in master data attributes.

Each class marked with the stereotype “MasterData” in the UML class diagram produced in the previous step originates one Entity service with the same name. This rule ensures cohesion of the Entity services because they deal with a set of attributes grouped by functional dependency constraints. Entity service operations should follow the format described in Table 3. This format is a subset of the CRUD operations defined in [Baghdadi 2006]. This subset was chosen based on the needs of composition to create the coarse-grained Task services and Process services as described in the step 5.2.6.

Table 3: CRUD operations format

Operation	Signature	Explanation
Create	Boolean Insert_MasterDataName (String tn, Object la, Object lv) [a]	When inserting a row that contains values for foreign keys, check if it exists as value of a primary key.

Update	Boolean Update_MasterDataName (String tn, Object la, Object lv, String wc)	When updating a list of attributes that contains values for foreign keys, check if it exists as value of a primary key.
Delete	Boolean Delete_MasterDataName(String tn, String wc)	When deleting a row check if it is referenced as foreign key by other tables.
Read	Object Read_MasterDataName (Object lt, Object la, String wc)	Retrieves specific attributes from one or more tables.

[a] tn - table name, lt - list of tables, la - list of attributes, lv - list of values, wc - where clause.

This step is conducted by the Software Architect with participation of the Data Analyst. The output is a list containing all Entity services and methods elicited in this step.

5.2.5 Artifact-Centric Model Creation

Artifact-centric process modeling technique defines business processes in terms of interacting business artifacts. Each business artifact is characterized by an information model (set of attributes) and a lifecycle model describing tasks that can be invoked on these artifacts [Cohn and Hull 2009]. A service comprises one or more tasks that perform operations on some artifact(s), where these operations should reflect steps of progress towards the business goal.

In MDCSIM, the master data play the role of artifact whose information model is expressed by the UML class diagram defined in the step 5.2.3. The lifecycle model is expressed by UML state machine diagrams created in this step. Existing state machine diagrams can be used, only if they describe state transitions of the master data identified in the step 5.2.2. State machine diagrams are used to identify Task and Process services that implement business rules, interactions among master data or other activities that use Utility or Entity services. A state machine diagram must be constructed for Master data that have transitions different of the CRUD operations or have transitions that participate in orchestrations. Transitions and business rules should be modeled using the Events, Conditions and Actions format and OCL (Object Constraint Language) [OMG 2012]. The usage of such patterns allows the explicitation of the inputs, outputs and internal actions of the services, creating a more detailed input for the specification phase in a service-oriented modeling and architectural design process. Each transition is an operation in a Task service or an Entity service. A set of transitions that are executed in a predefined order can be modeled as a Process service. Rules to transform state machine transitions into services are detailed in the step 5.2.6.

This step is conducted by the Business Analyst and the Software Architect. The output is a set of UML state machine diagrams.

5.2.6 Task Services and Process Services Design

This step aims to update Entity services' operations and to design Task services and Process services. Each state machine diagram constructed in the previous step has its transitions transformed into service operations. Each transition is mapped to one operation. This rule states that an operation comprises the events, conditions and actions modeled into one transition. Basic CRUD operations can be referenced within events, conditions and actions. In this case, the identified operation will reference an Entity service. The master data attributes manipulated by the transition must also be included as operation parameters.

Operations should be grouped into services in accordance with the pattern presented in Table 4. This pattern analyzes the source and target states of the transition that originated the operation and defines the destination service where the operation will be included. This pattern aims to segregate transitions related to the master data behavior from transitions related to master data maintenance. The application of this pattern also ensures that services are identified according to the layers model (Section 5.1).

Table 4: Operations grouping pattern

Condition	Destination service
The transition source state is the same of target state.	This operation should be included in the *Entity service* that represents the master data that owns the state machine diagram where the transition was modeled.
The transition source state is different from the target state.	This transition is related with control flow and should be included in a *Task service*. It should be created only one *Task service* per state machine diagram to gather operations derived from this kind of transitions.

After identifying Task services and updating Entity services, the Software Architect can analyze the need to represent sequences of state transitions as Process services. The transitions that participate in orchestrations must be marked with the stereotype "Orchestration". This step is conducted by the Software Architect. The output is the service list comprising the updated Entity services and the Task services and Process services indentified in this step.

5.3 Chapter Summary

This chapter introduced MDCSIM and depicted its steps, roles, inputs and outputs. A service layer model used to aid service identification was also presented.

6 Implementation

This chapter provides an overview of the model-driven approach justifying its utilization to automate MDCSIM. The tool developed to support MDCSIM (MDCSIM plug-in) is presented and its operation, inputs and outputs are depicted. Finally the technologies used within the implementation are described.

6.1 Model-driven approach overview

According to the Object Management group (OMG) the model driven architecture (MDA) is an approach to use models in software development. MDA separates the specification of a system from the details about the way that system uses the capabilities of its computational platform [OMG 2003]. MDA is designed according to three levels of abstraction: the Computation Independent Model (CIM), the Platform Independent Model (PIM) and the Platform Specific Model (PSM). CIM models focus on requirements and business rules of the systems. PIM models describe the operation of a system independently of a target platform. PSM complements information of the PIM models with an additional focus on a specific platform to be used by the system.

Models are interrelated. One model can be converted to another model of the same system in a process named Transformation. Transformations can be Model-To-Model or Model-To-Code. The process of transformation is supported by standards such as Meta-Object Facility (MOF), Unified Modeling Language (UML) and XML Metadata Interchange (XMI) in order to ensure high level of completeness and consistency of the models. MOF is a standard used in the specification and development of meta-models. A meta-model precisely describes the properties and constructs of every model. UML, as a graphical modeling language, provides the basic constructs to define and visualize meta-models. XMI defines rules for interchanging models.

In service-oriented analysis and design the high-level business view can be represented with CIM models, while the information system view can be represented first by PIMs, and then specified in PSMs [Cai et al. 2011]. For the service identification phase, CIMs are used to describe business domain and PIMs are built to identify candidate services, while

PSM can be used in the service specification phase. Since the models defined in MDA can be correlated to the process of service identification and MDCSIM uses UMLs diagrams to describe the business domain, MDA is a natural choice as the underpinning technology for the implementation of a tool to support MDCSIM (MDCSIM plug-in). Such tool aims to facilitate MDCSIM utilization, contributing for its scalability.

6.2 MDCSIM plug-in

MDCSIM plug-in is a tool that reads the stereotyped UML class diagram (master data attributes) created in the step 5.2.3 Master Data attributes identification and the UML state machine diagrams (master data lifecycles) created in the step 5.2.5 Artifact-Centric Model Creation and applies the rules defined in the steps 5.2.4 Entity Services Design and 5.2.6 Task Services and Process Services Design in order to generate a UML class diagram of identified candidate services. MDCSIM plug-in builds on MDA abstractions. The UML class diagram and UML state machine diagrams correspond to the CIM model and the UML class diagram of identified services corresponds to the PIM model. As explained in the aforementioned section, the PSM abstraction is not used in the identification phase. Nevertheless, MDCSIM plug-in can be extended to generate PSM models supporting the specification and implementation phases in the service design process.

Figure 8 shows MDCSIM plug-in inputs, outputs and the correlation of each element with MDA abstractions.

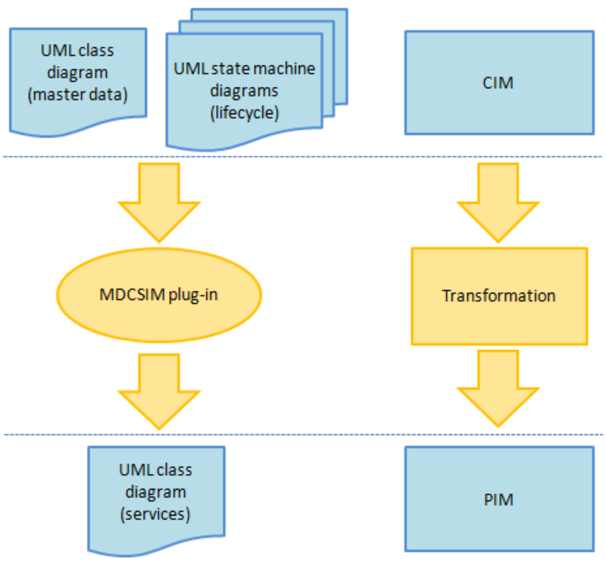

Figure 8: MDCSIM plug-in correlation with MDA abstractions.

MDCSIM plug-in was built using ATLAS transformation language (ATL) [Jouault and Kurtev 2005]. ATL provides a modeling transformation platform to transform a set of source models into a set of target models. An ATL transformation is composed of rules that define how source model elements are navigated to create the elements of the target models. The source and target models conform to meta-models or standardized meta-meta-models such as MOF and Ecore. ATL is developed on top of the Eclipse environment as an Integrated Development Environment (IDE) and an ATL transformation engine is used to compile and execute ATL programs. ATL was chosen because:

- It provides a complete transformation model and supports complex transformations, and
- ATL has gained extensive support for development from the user community (various examples and case studies are available).

In MDCSIM plug-in the transformation is named *IdentifyServices.atl*. This transformation implements the rules that retrieve each element of the UML class diagram of master data and of the UML state machine diagrams and generate the UML class diagram of identified candidate services. The complete code of the *IdentifyServices.atl* transformation is presented in the Appendix B.

The source and target models were constructed and visualized using Papyrus [Lanusse et al. 2009]. Papyrus is graphical editing tool for UML2 as defined by OMG. As ATL, Papyrus is developed on top of the Eclipse environment as an IDE. Papyrus has also the advantage to offer advanced support of UML profiles. The source and target models comply to a profile meta-model named *Stereotype.profile.di* which is based on Ecore meta-meta-model. The profile defines two stereotypes: *MasterData* and *Orchestration* as presented in Figure 9.

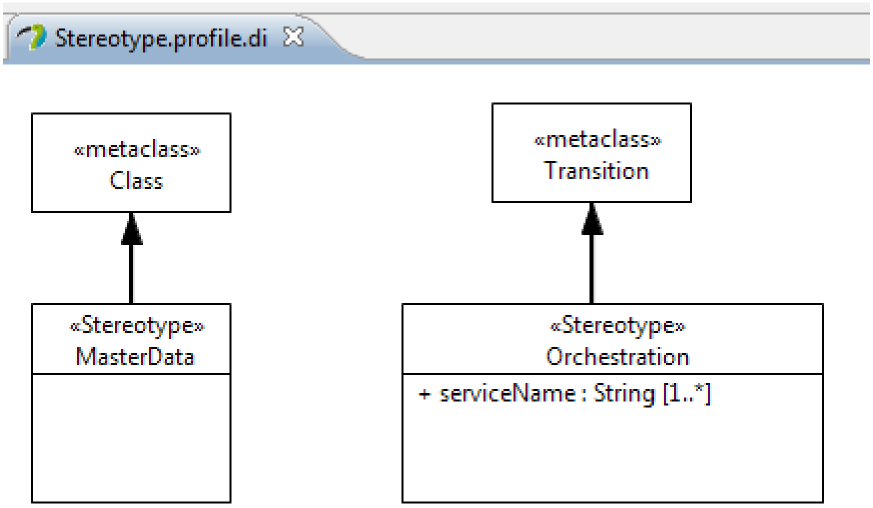

Figure 9: Profile and stereotypes defined in MDCSIM plug-in.

Figure 9 shows that *MasterData* stereotype is a Class stereotype. It is used to identify which classes correspond to a master data in the class diagram used as input for MDCSIM plug-in. *Orchestration* is a Transition stereotype. It is used in the state machine diagrams in order to identify which transitions are parts of orchestrations. *Orchestration* stereotype has an attribute named *serviceName* that defines the name of the service and of the operation that orchestrates the transition(s). For instance, when two transitions have the stereotype *Orchestration* and the same name in the attribute *serviceName* they are part of the same orchestration. One transition can participate of more than one orchestration. Orchestrations originate Process services.

The utilization of MDCSIM plug-in, as well as the construction of the aforementioned source models are demonstrated in the section 7.3 Service Identification Using MDCSIM.

6.3 Chapter Summary

This chapter provided an overview of the model-driven approach, discussing MDA utilization in the service identification phase and justifying its adoption to automate MDCSIM. MDCSIM plug-in was presented and its operation, inputs and outputs were depicted. A correlation of MDCSIM plug-in with MDA abstractions was provided. Finally the technologies used within the implementation were described.

7 Proof of Concept

This chapter depicts a proof of concept of the MDCSIM using a real business scenario in a reinsurance company. The proof of concept demonstrates each step defined by MDCSIM and the outputs generated. Then, other two data-focused SIM are tested in the same business scenario in order to promote a qualitative comparison with MDCSIM. A quantitative comparison is also accomplished by measuring the granularity, coupling, cohesion and entity convergence quality attributes of the service portfolio identified by each SIM. Finally, some threads to validity are discussed.

7.1 Evaluation Steps

In order to demonstrate the execution of MDCSIM steps and evidence if MDCSIM is suitable to support service identification phase in a service-oriented modeling process, accomplishing the drawbacks mentioned in Chapter 4, the following steps are planned:

- Conduction of a proof of concept using a real business scenario: Two iterations of MDCSIM will be performed in correlated business areas, demonstrating service reuse between iterations and the practical usage of method and of the MDCSIM plug-in.
- Qualitative comparison with two other data-focused SIMs: Application of the two other SIMs in the same business scenario and comparison of the types of services identified and their ability to cover the business domain.
- Quantitative comparison with two other data-focused SIMs: Use of metrics to compare the granularity, coupling, cohesion and entity convergence quality attributes of the service portfolio identified by each SIM. The reuse accomplished by each method was also assessed.

7.2 Business Scenario

The business scenario used to test MDCSIM is of a reinsurance company. The core areas of the reinsurance company are Sales and Claim processing. The Sales area comprises the activities of risk analysis, underwriting and premium processing. The Claim processing area comprises the activities of claim notification receipt, claim analysis (regulation) and claim recovery. The Claim area was prioritized because it deals with several legal constraints (e. g. time to answer claim notifications or pay indemnities). Due to confidentiality issues the business scenario used in this proof of concepts presents only the subset of activities for the Guarantee reinsurance line.

The first process of the Claim area is Claim Notification Reception. The Claim reception department receives claim notifications and analyzes it in order to identify if the loss is covered by the contracted covers in the reinsurance, if the reinsurance is up to date or to calculate the applicable penalties. The notification can be reproved or forwarded to the claim recovery department. If it is forwarded, a provision to liquidate the claim must be created, the contractor must be blocked and the foreign participants and managers must be notified depending on the loss estimated value. Figure 10 presents the activity diagram of the Claim Notification Reception process. The process documentation is not used as an input of MDCSIM. The process is presented here in order to describe the business scenario and enable the application of the other SIMs in the section 7.4 Service Identification Using Other Methods.

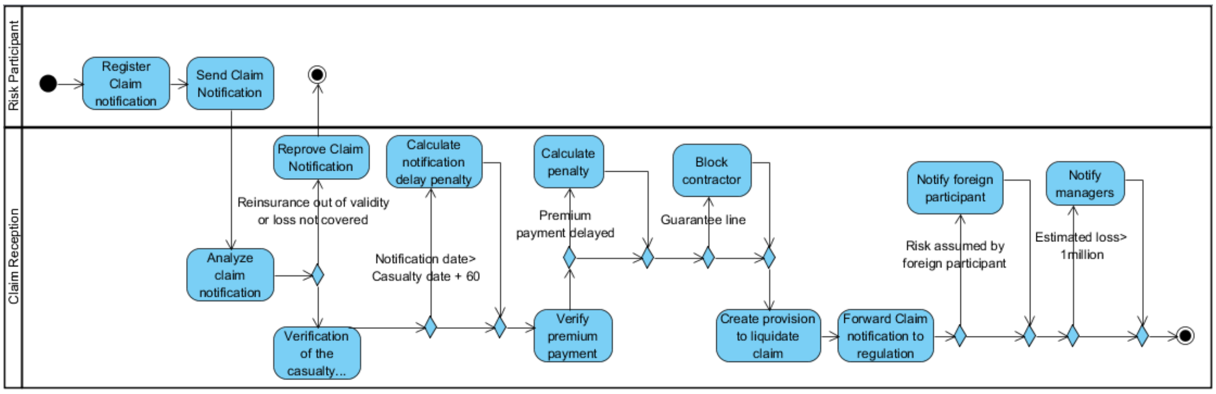

Figure 10: Claim Notification Reception process.

The second process of the Claim area is Claim Recovery process. The Claim can be regulated by the own reinsurer or delegated to the risk participant depending on the percentage of reinsurance assignment and on the estimated loss value. When the claim is regulated by the risk participant, the Claim recovery department is responsible to monitor the process and answer to the participant requests. Otherwise, the Claim recovery department names a regulator to deal with the claim. The regulator performs inspections in order to calculate the real loss and confirm the accident cover. The regulator can ask the Claim recovery department to adjust the claim provision, to refuse the claim or to authorize the claim recovery. Accordingly to the regulator request, the Claim recovery department updates claim entries (provisions, expenses or indemnities paid for the claim), registers payments, authorizes claim recovery or refuses claim recovery and notifies the risk participant. The risk participant can send recovery requests (requests for indemnity anticipation) which are also analyzed by the claim recovery department. Figure 11 shows the activity diagram of the Claim Recovery process.

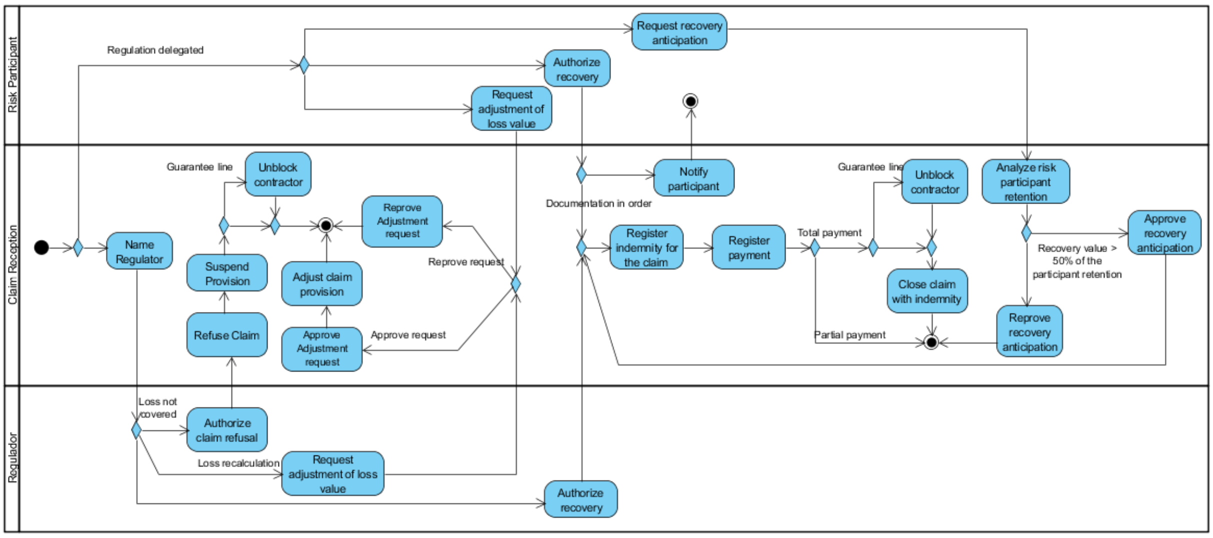

Figure 11: Claim Recovery process

7.3 Service Identification Using MDCSIM

7.3.1 First Iteration – Claim Notification Reception Process

Service identification using MDCSIM was accomplished in two iterations within the Claim area. For the first iteration the prioritized process was the Claim Notification Reception process. Although the business process documentation is not used in MDCSIM, the iterations were sliced according to the processes executed by the Claim processing area. Such division aims to facilitate the comparison with the other SIMs ([Baghdadi 2006] and [Yun et al. 2009]) and also to assess service reuse within iterations. After defining the scope of analysis as stated in the step 5.2.1, the related master data should be identified. Master data identification is supported by the questions described in step 5.2.2. Table 5 presents the master data identified for the Claim Notification Reception process:

Table 5: List of master data identified for the Claim Notification Reception process.

Master Data	Question	Meaning
Claim entry	How?	Records of provisions, expenses or reimbursements paid for a claim.
Claim notification	How?	Notification of a loss resulting from an accident in order to start the process of indemnity and reimbursement.
Contract	How?	Formal instrument to effectuate the risk distribution to foreign participants.
Contractor	Who?	Part contracted for execution of works or supply of goods or services. This term is only used in reinsurances of the guarantee line. The contractor pays the premium.
Cover	What?	The scope of protection provided under reinsurance.
Line	How?	Category of reinsurance, such as the liability line or the guarantee line.
Modality	How?	Form of reinsurance hiring. For example, facultative reinsurance modality is used for the hiring of individual risks. Each line has its modalities defined by law.
Premium	How?	The price of reinsurance protection for a specified risk for a specified period of time.
Reinsured	Who?	Part who receives the indemnity in case of a loss under protection of reinsurance.
Reinsurance	How?	Agreement in which one or more Risk participants indemnify an insurer for all or part of the risk of a policy originally issued and assumed by the insurer.
Risk participant	Who?	Organization that assumes all or part of a specified risk for a premium. It is an insurer or a reinsurer.

The mater data Contractor, Reinsured and Risk participant corresponds to the Party domain (identified by the “Who?” question). The master data Cover corresponds to the Things domain (identified by the “What?” question). Finally, Claim entry, Claim notification, Contract, Line, Modality, Premium and Reinsurance address how the Reinsured is related to the Reinsurer or the Risk Participant is related to the Reinsurer (identified by the “How?” question). In order to identify master data attributes as stated in step 5.2.3, the UML class diagram presented in Figure 12 was constructed. The master data class diagram was constructed by consolidating existing conceptual data models, master data definitions provided by business specialists and the logical data models obtained from reverse engineering of the databases used by the Integrated Business System, by the Claim List System and by the Guarantee of Contractual Obligations System. It is important to mention that, although there were conceptual data models in the organization, we had to convert them into class diagrams, because it is the input format supported by MDCSIM plug-in.

In the consolidation process, the attributes that describe each master data were manually identified from the attributes scattered within the tables of the three databases and then gathered into one class per master data. Redundant attributes were removed and definitions were harmonized with the aid of the master data catalog and of the business knowledge of the analyst running MDCSIM. Classes that are not master data, but have relationships with the master data were also identified and then added to the master data class diagram. Finally the master data class diagram was normalized.

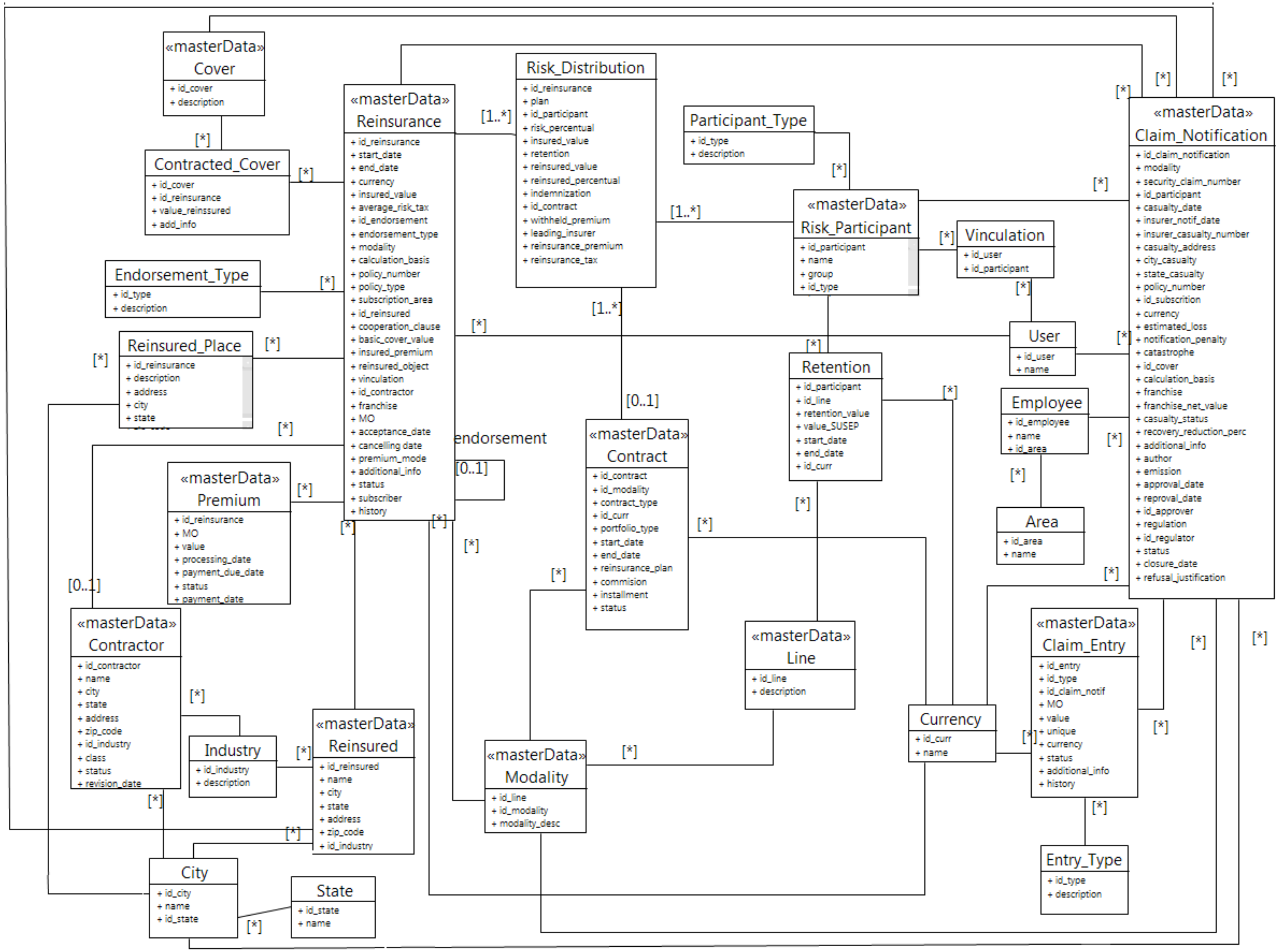

Figure 12: UML Class Diagram representing the logical data model.

The diagram was modeled with the aid of MDCSIM plug-in described in Chapter 6. The classes that represent the aforementioned master data are marked with the stereotype *masterData* and named with the corresponding master data name. Figure 12 presents only the main attributes of the business domain in order to be legible. Nevertheless all attributes that compose a master data are taken into consideration by the CRUD operation patterns presented in Table 3.

In the next step of MDCSIM (5.2.4 Entity Services Design) each master data of the UML class diagram presented in Figure 12 originates one Entity service that comprises the CRUD operations presented in Table 3. This task was accomplished by running the transformation *IdentifyServices* from the MDCSIM plug-in using as input the master data class diagram as presented in Figure 13. The class diagram of the identified Entity candidate services and their CRUD operations is presented in Figure 14.

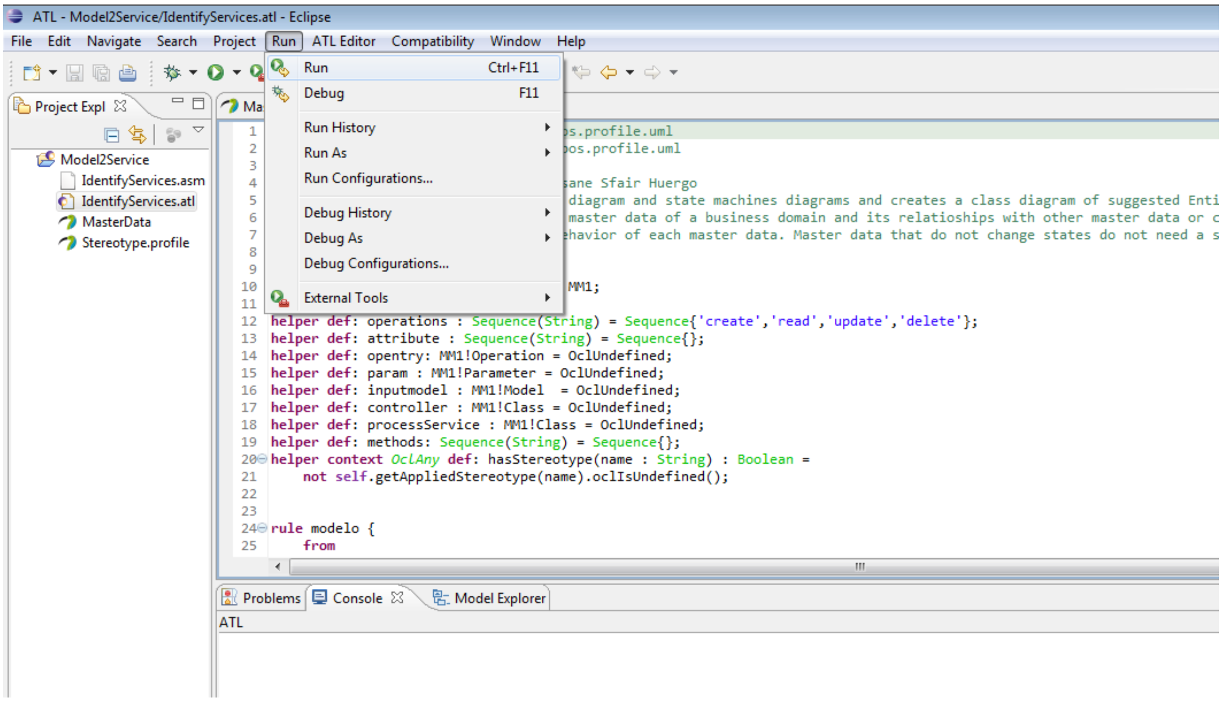

Figure 13: Entity service identification in MDCSIM plug-in.

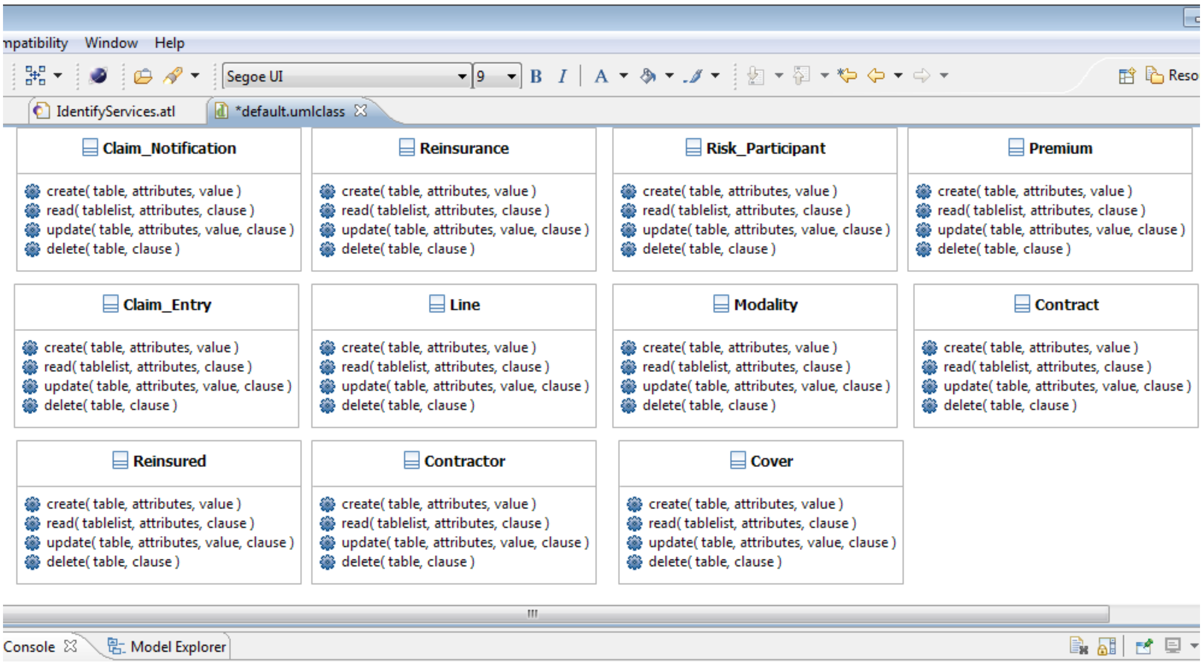

Figure 14: Entity services identified in the first iteration of MDCSIM.

The next step (5.2.5) is the Artifact-centric model creation, when state machine diagrams must be designed to master data identified in the previous steps. Master data that do not have transitions different from the CRUD operations do not need a state machine diagram (in this scenario, this was the case for Cover, Line, Modality and Reinsured). The state machine diagrams were modeled with the aid of MDCSIM plug-in as presented in Figure 15, **Erro! Fonte de referência não encontrada.**, Figure 17, Figure 18, Figure 19, Figure 20 and Figure 21. Each transition was modeled using OCL and ECA pattern as presented in the aforementioned figures. Some basic CRUD operations (for example read and delete operations) were not included in the state machine diagrams, in order to simplify the visualization. Nevertheless, this kind of operation was already identified in the Entity Services Design step. Transitions that must be orchestrated in order to compose Process services were marked with the stereotype orchestration and the name of the Process service was defined as a property of the stereotype.

Although we had the effort to create the state machine diagrams in this step, these diagrams will be completely reused in the subsequent iterations where the identified master data take part. Even when it is necessary to create the aforementioned models from scratch, MDCSIM demands less modeling effort, than methods that use process as inputs, because the majority of the business processes in an organization are described by a few number of master data.

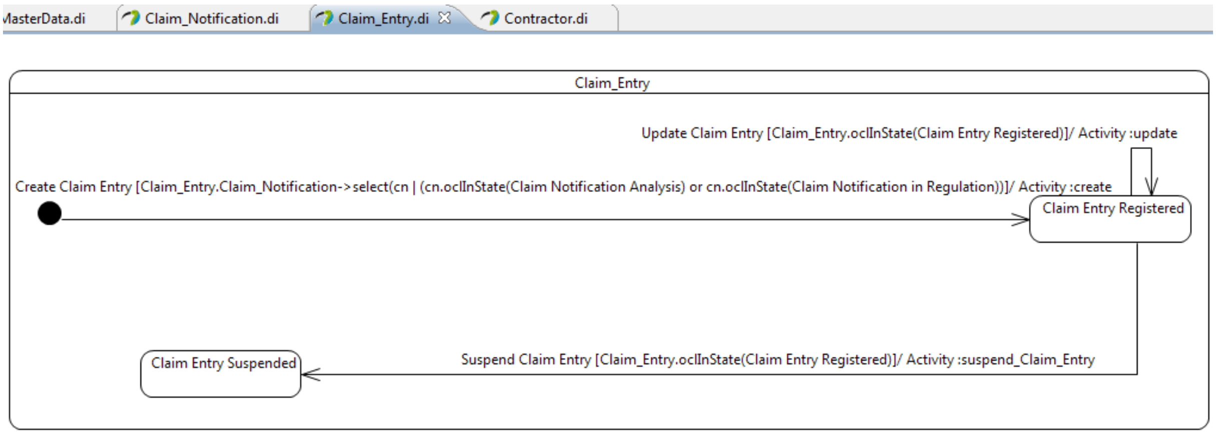

Figure 15: State machine diagram of the Claim Entry master data lifecycle.

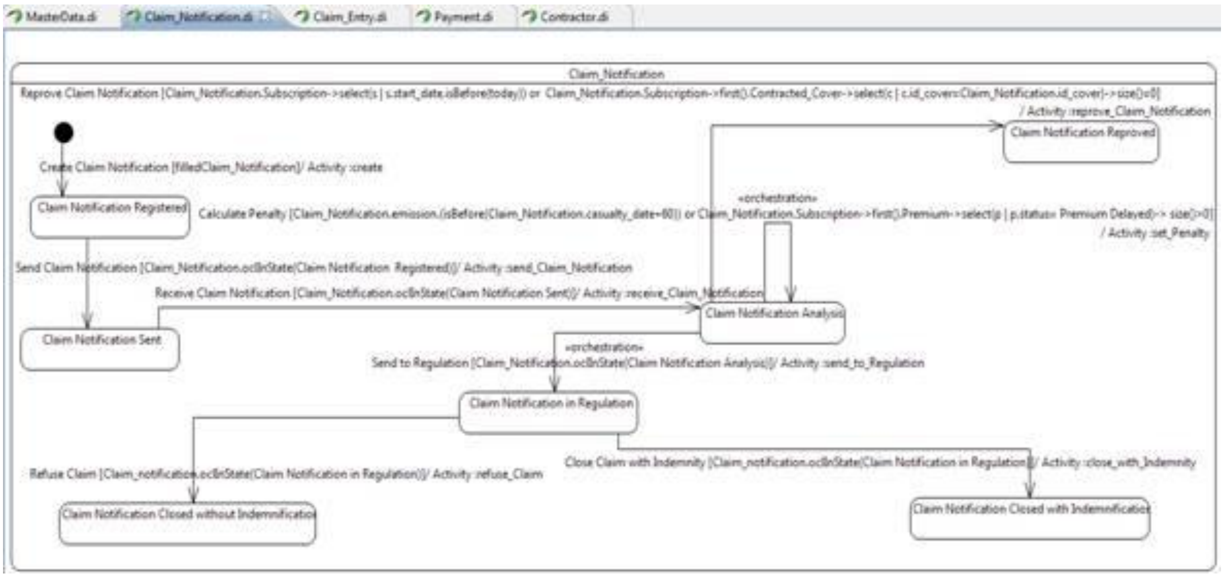

Figure 16: State machine diagram of the Claim Notification master data lifecycle.

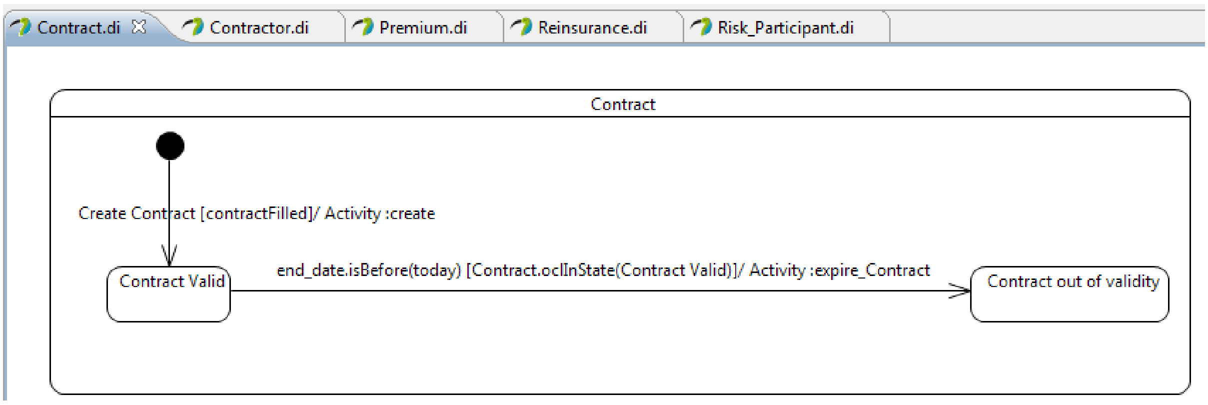

Figure 17: State machine diagram of the Contract master data lifecycle.

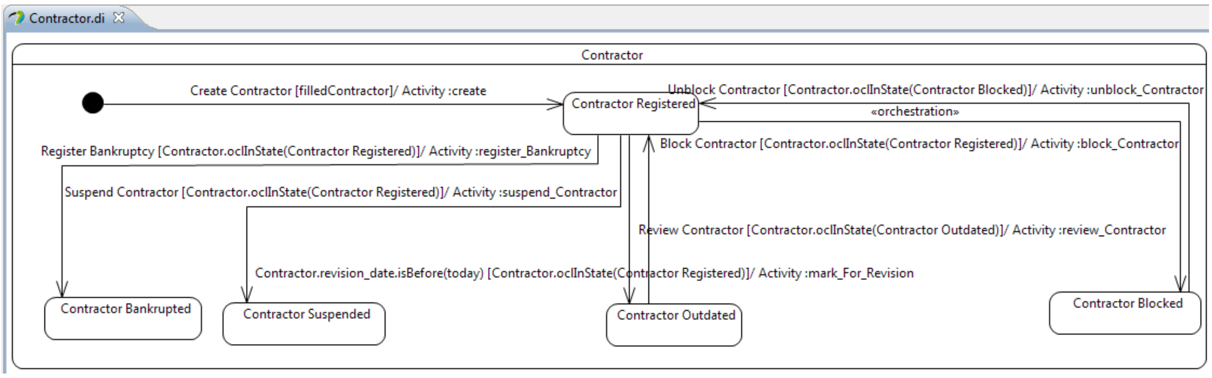

Figure 18: State machine diagram of the Contractor master data lifecycle.

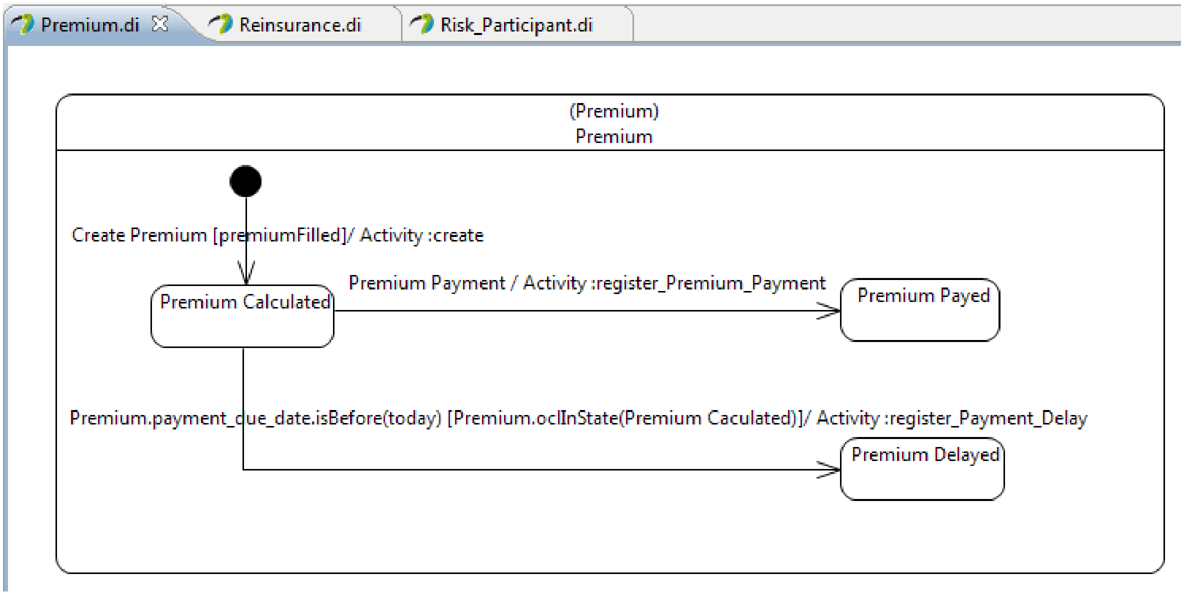

Figure 19: State machine diagram of the Premium master data lifecycle.

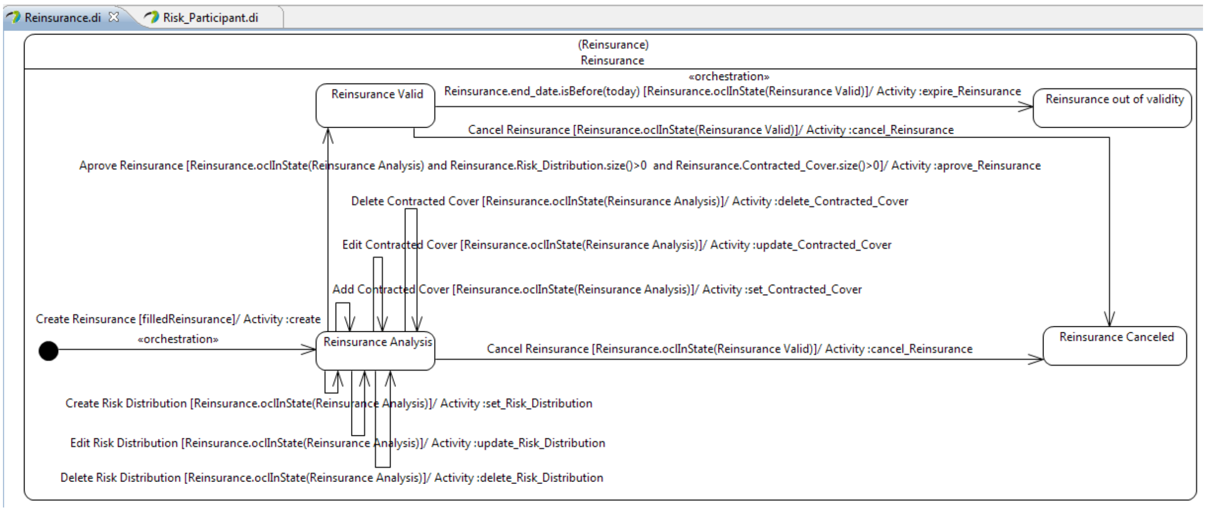

Figure 20: State machine diagram of the Reinsurance master data lifecycle.

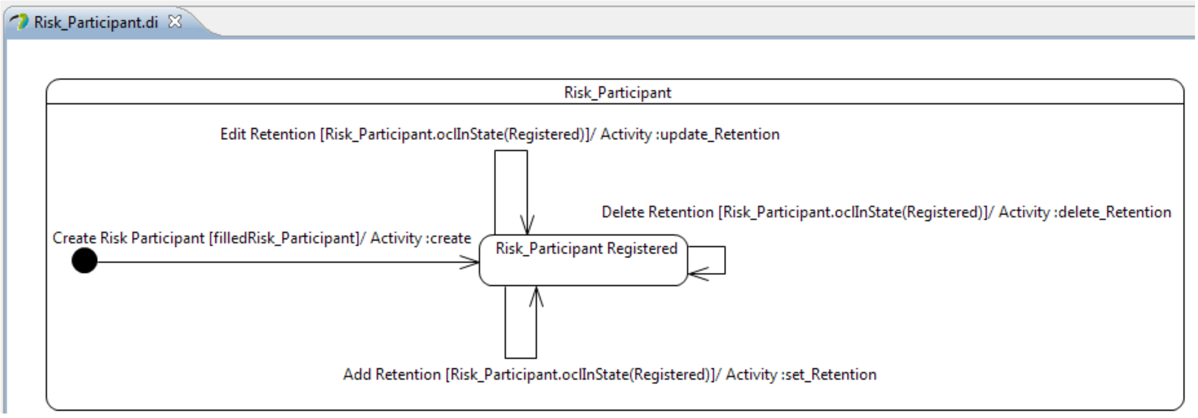

Figure 21: State machine diagram of the Risk participant master data lifecycle.

In the last step of MDCSIM (5.2.6 Task Services and Process Services Design), each transition that does not correspond to a basic CRUD operation identified in the Entity services design step was transformed into a service operation. Services operations were identified by running MDCSIM plug-in using the state machine diagrams designed in the previous steps as inputs. Operations were included in Entity services identified previously or in Task services created in this step according to the pattern described in Table 4. Each created Task service was named adopting the convention <Master data name _Controller>. For example, the transition */setPenalty (int id_claim_notification)* does not change Claim notification's state, so it becomes an operation with the same name as the transition in the *Claim_Notification* service. On the other hand, the transition */reprove_Claim_Notification (int id_claim_notification, date reproval_date)* was added to the *Claim_Notification_Controller* service, because it changes the *Claim_Notification* state to "Claim Notification Reproved".

MDCSIM plug-in also created process services by identifying transitions marked with the stereotype *Orchestration* and gathering them into the Process service whose name was defined as the stereotype property. The final list of Entity, Task and Process candidate services identified for the first iteration of MDCSIM is presented in Figure 22.

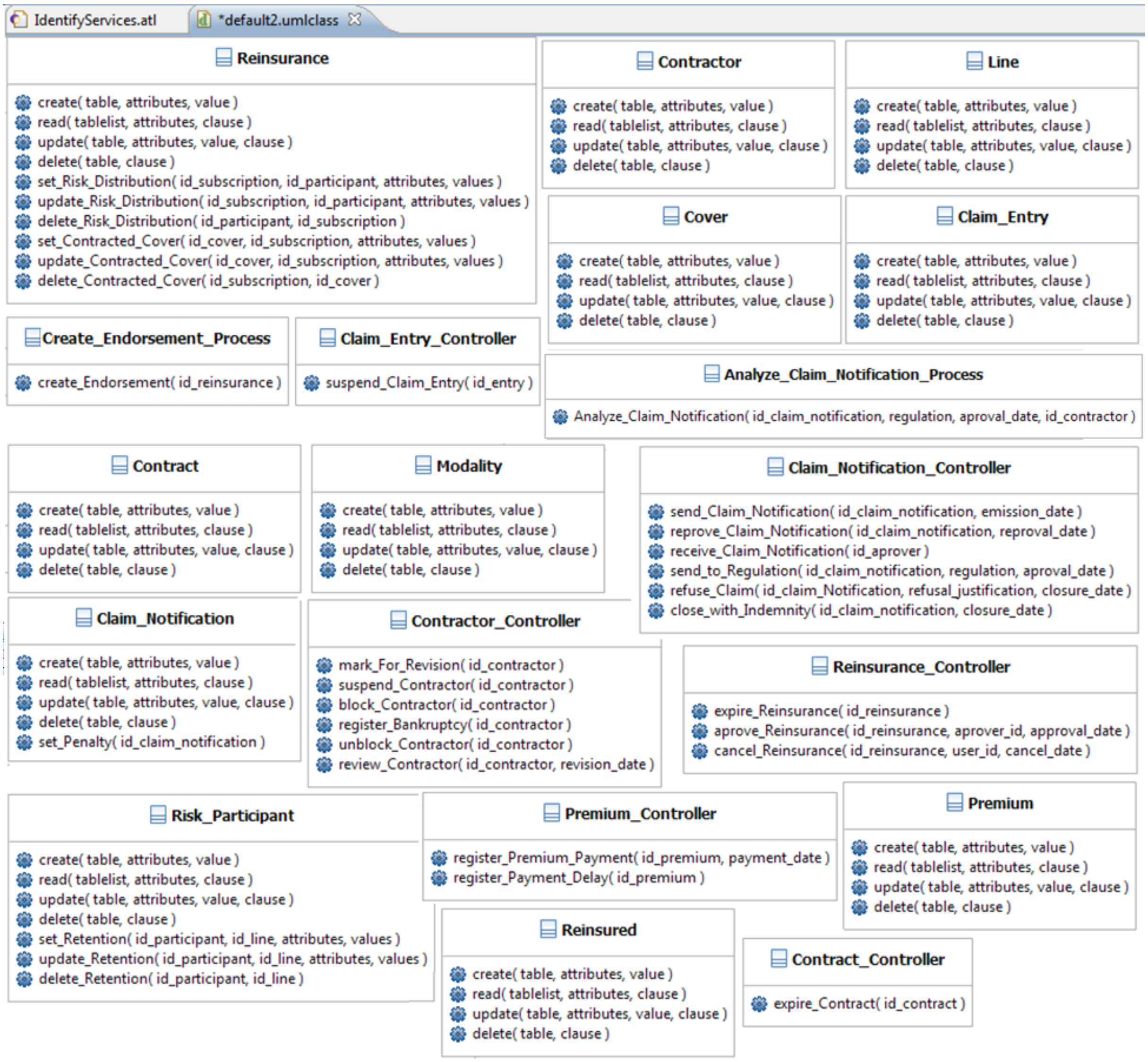

Figure 22: Class diagram of the identified Entity, Task and Process candidate services.

In total 19 services were identified: 11 Entity services (*Claim_Entry, Claim_Notification, Contractor, Contract, Cover, Line, Modality, Premium, Reinsurance, Reinsured* and *Risk_Participant*), 6 Task services (*Claim_Entry_Controller, Claim_Notification_Controller, Contract_Controller, Contractor_Controller, Premium_Controller* and *Reinsurance_Controller*) and 2 Process services (*Analyze_Claim_Notification_Process* and *Create_Endorsement_Process*). *Analyze_Claim_Notification_Process* service orchestrates the services *Claim_Notification, Claim_Notification_Controller* and *Contractor_Controller* in order to calculate the applicable penalties, block the contractor and then send the claim notification to the regulation department. *Create_Endorsement_Process* service orchestrates the *Reinsurance* service and the *Reinsurance_Controller* service in order to close the previous reinsurance when an endorsement is created.

7.3.2 Second Iteration – Claim Recovery Process

The prioritized process for the second iteration was the Claim recovery process. Table 6 presents the master data identified for this process:

Table 6: List of master data identified for the Claim recovery process.

Master Data	**Question**	**Meaning**
Adjustment request	How?	Request for adjustment of the claim provision (claim entry). It is filled by the responsible for regulating the claim.
Claim entry	How?	Records of provisions, expenses or reimbursements paid for a claim.
Claim notification	How?	Notification of a loss resulting from an accident in order to start the process of indemnity and reimbursement.
Contractor	Who?	Part contracted for execution of works or supply of goods or services. This term is only used in reinsurances of the guarantee line. The contractor pays the premium.
Payment	How?	Record of an amount paid or programmed to be paid as reimbursement for the indemnity payment.
Recovery Request	How?	Request for reimbursement anticipation. It is filled by the risk participant who paid the indemnity for the insured.
Reinsured	Who?	Part who receives the indemnity in case of a loss under protection of reinsurance.
Risk participant	Who?	Organization that assumes all or part of a specified risk for a premium. It is an insurer or a reinsurer.

The master data Adjustment request, Payment and Recovery request address how the Risk Participant is related to the Reinsurer (identified by the “How?” question). Claim entry, Claim notification, Contractor, Reinsured and Risk participant were already mapped in the previous iteration. In order to identify master data attributes, the UML class diagram constructed in the previous iteration was updated with the master data and the classes that have relationship with them identified in this iteration as presented in Figure 23.

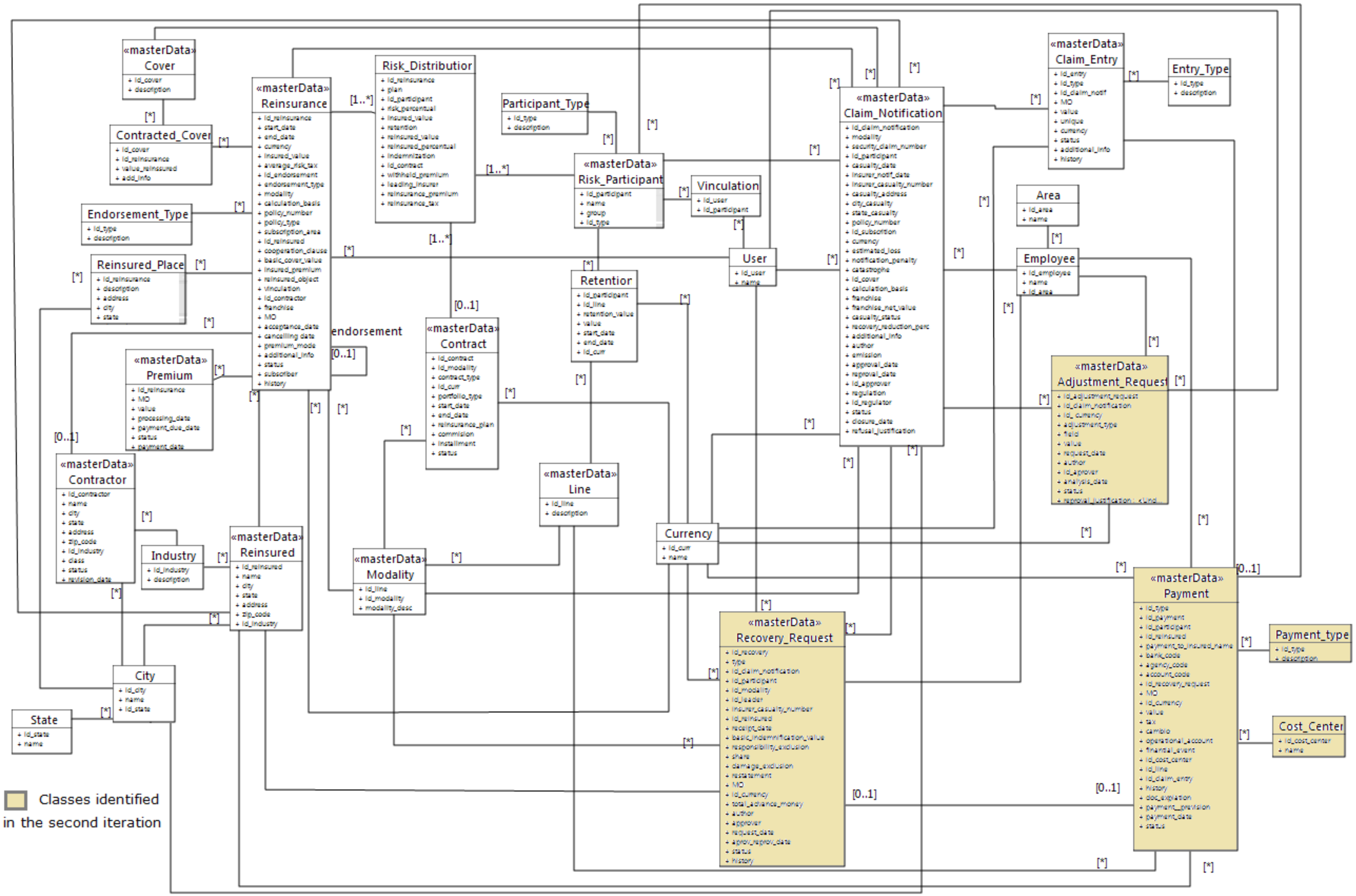

Figure 23: UML Class Diagram representing the updated logical data model.

For the next step (5.2.4 Entity Services Design) the transformation *IdentifyServices* from the MDCSIM plug-in was executed using as input the updated master data class diagram presented in Figure 23. The Entity services from the master data mapped in this iteration are identified. The Entity services from the master data mapped in iteration 1 are repeated and they might be reused by the services identified in this iteration. The new Entity candidate services and their CRUD operations are presented in Figure 24.

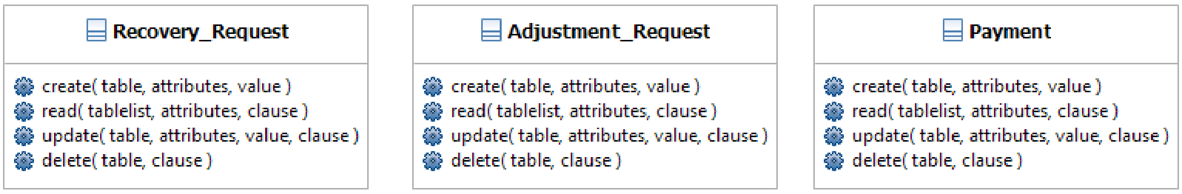

Figure 24: Entity services from the master data identified in MDCSIM second iteration.

In the next step (5.2.5 Artifact-centric model creation), state machine diagrams were designed for the master data identified in this iteration. The new state machine diagrams are presented in Figure 25, Figure 26 and Figure 27.

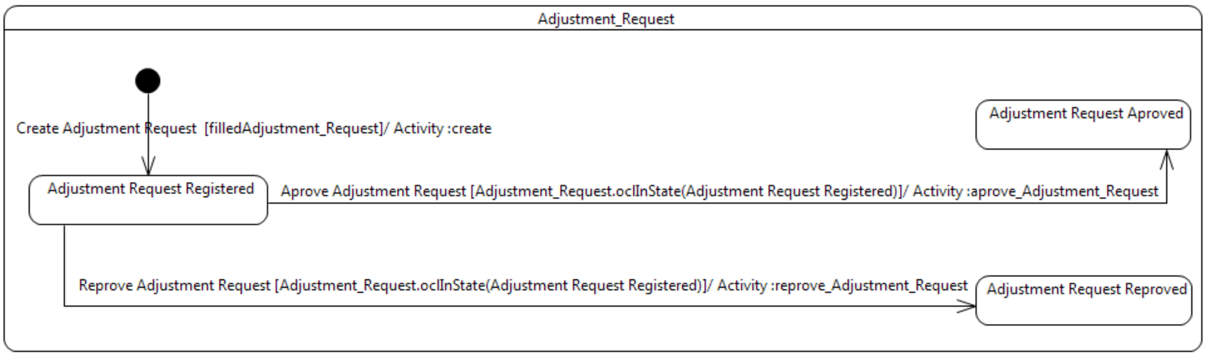

Figure 25: State machine diagram of the Adjustment request master data lifecycle.

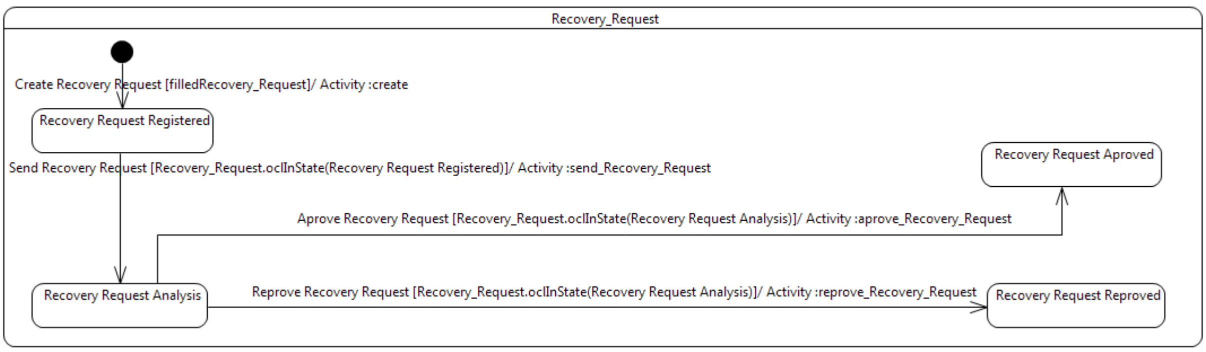

Figure 26: State machine diagram of the Recovery request master data lifecycle.

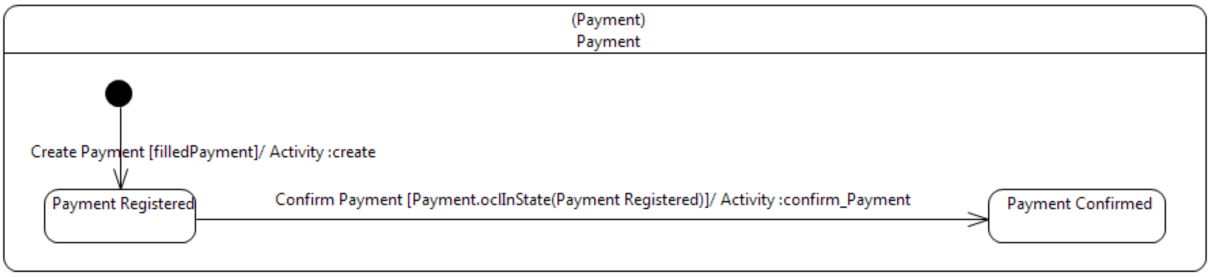

Figure 27: State machine diagram of the Payment master data lifecycle.

The state machine diagrams modeled in the previous iteration were also revisited, in this step, in order to identify new transitions, or transitions that must be orchestrated to fulfill requirements of the Claim recovery process. No new orchestrations were identified, but a transition was included in the Claim Notification diagram as shown in Figure 28.

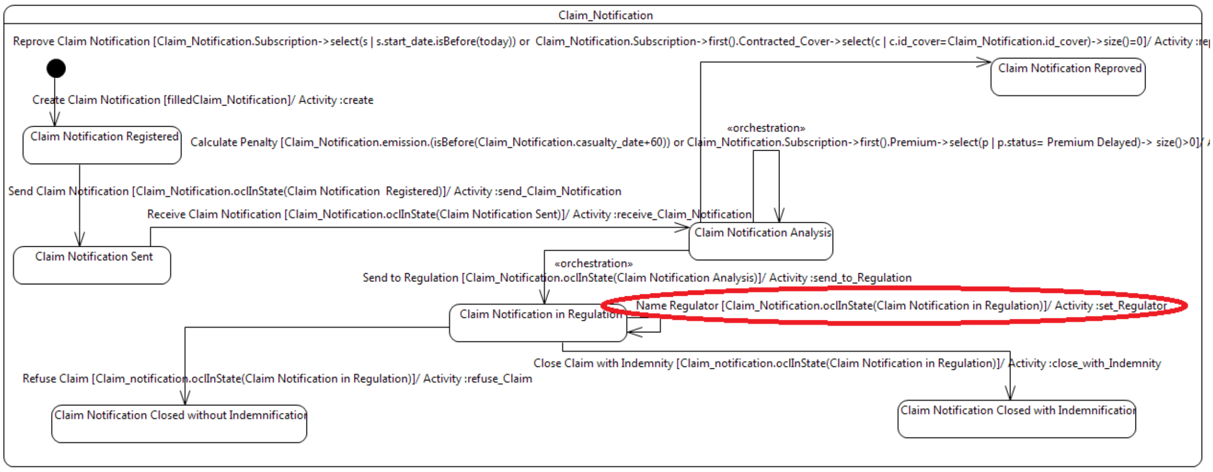

Figure 28: Revisited Claim Notification state machine diagram.

In the last step of MDCSIM (5.2.6 Task Services and Process Services Design), the new and the revisited state machine diagrams were processed by the MDCSIM plug-in in order to identify Task and Process services. A total of 6 new services were identified: 3 Entity services (*Adjustment_Request*, *Payment* and *Recovery_Request*) and 3 Task services (*Adjustment_Request_Controller*, *Payment_Controller* and *Recovery_Request_Controller*). The final list of Entity, Task and Process candidate services identified for both iterations is presented in Figure 29.

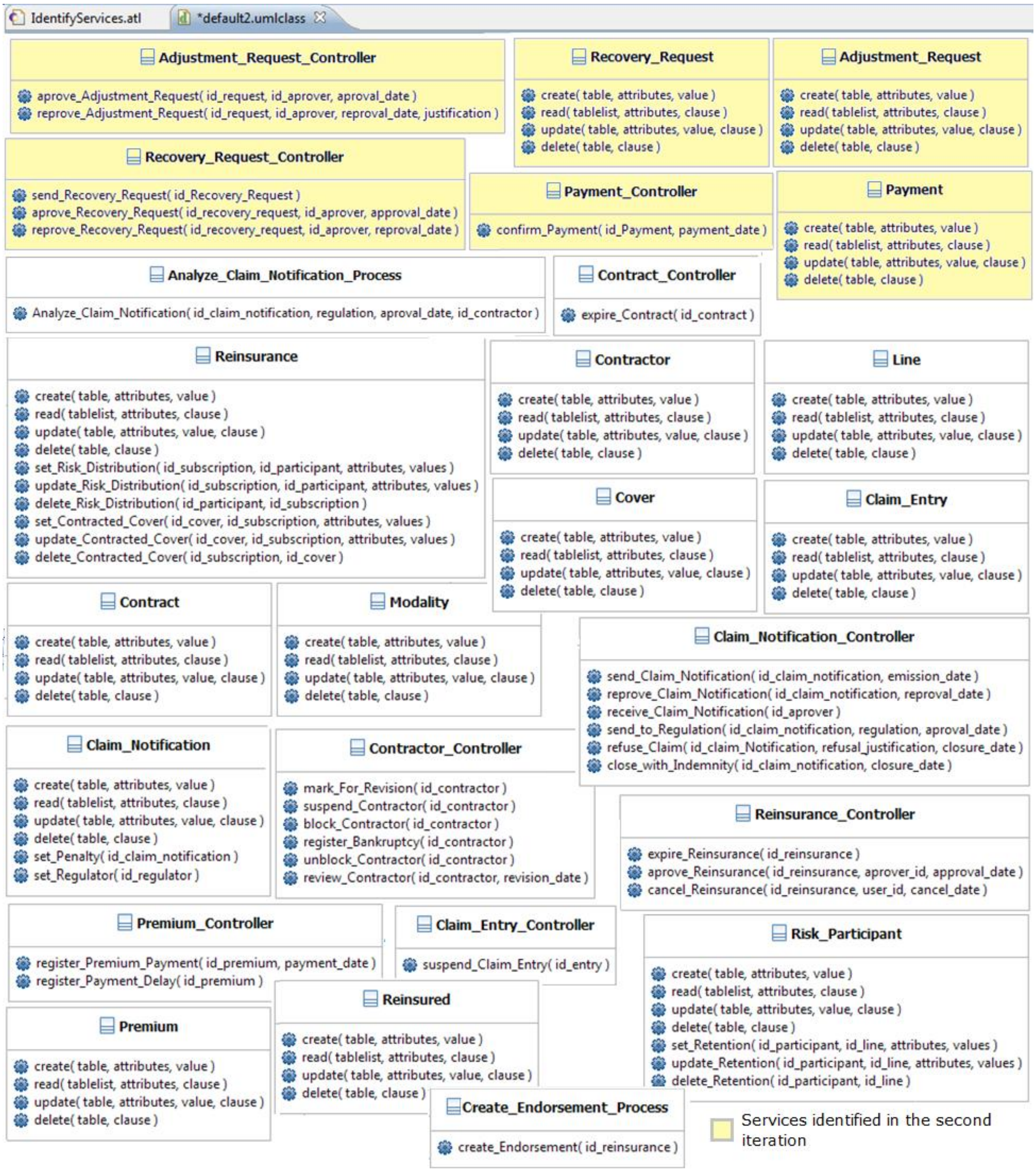

Figure 29: Class diagram of the complete list of identified services.

It is important to mention that some operations of the Task services identified in the first iteration are not executed in the scope of the Claim Reception process, but they were identified during the elicitation of the master data lifecycle. These operations were modeled in order to provide wider service interfaces, taking into account the scope of the whole organization. Among these operations, the operations *suspend_Claim_Entry* of the *Claim_Entry_Controller* service, *close_with_Indemnity* and *refuse_claim* of the

Claim_Notification_Controller service and *unblock_contractor* of the *Contractor_Controller* service were used in the second iteration for the Claim recovery process, without the need of changing these services contract.

7.4 Service Identification Using Other Methods

In order to provide an initial assessment of MDCSIM, two SIMs [Yun et al. 2009] and [Baghdadi 2006] were chosen to identify services using the business scenario aforementioned. These SIMs were chosen for comparison, because they are data-focused similarly to MDCSIM. Besides, such methods provide all the details necessary to their execution.

7.4.1 Service Identification Using Database Reverse Engineering [Baghdadi 2006]

The first method [Baghdadi 2006] uses database reverse engineering to identify basic data access services. The practitioner chooses the tables considered relevant to the domain. Each table originates one data service. Then the practitioner chooses the CRUD operations comprised by this service within a set of CRUD operations signatures provided by the method. This method does not provide guidelines to aid the choice of tables used to derive services. Besides, this method is classified as a Bottom-up approach, because it elicits only the IT perspective.

For the first iteration, we used the same UML class diagram (Figure 12) used for MDCSIM first iteration. The classes marked with the stereotype *MasterData* will be chosen to derive the candidate services. In order to get services more close to those identified by MDCSIM, all the CRUD operations will be added. Table 7 shows the identified services and their operations.

Table 7: Identified candidate services for the first iteration of [Baghdadi 2006].

Service	Operation
Claim_entry	public boolean createClaim_entry(String table, Object attributes, Object values);
	public boolean deleteClaim_entry(String table, String condition);
	public boolean updateClaim_entry(String table, Object attributes, Object values, String condition);
	public Object readClaim_entry(Object tables, Object attributes, String condition);
Claim_notification	public boolean createClaim_notification(String table, Object attributes, Object values);
	public boolean deleteClaim_notification(String table, String condition);

	public boolean updateClaim_notification(String table, Object attributes, Object values, String condition);
	public Object readClaim_notification(Object tables, Object attributes, String condition);
Contract	public boolean createContract(String table, Object attributes, Object values);
	public boolean deleteContract(String table, String condition);
	public boolean updateContract(String table, Object attributes, Object values, String condition);
	public Object readContract(Object tables, Object attributes, String condition);
Contractor	public boolean createContractor(String table, Object attributes, Object values);
	public boolean deleteContractor(String table, String condition);
	public boolean updateContractor(String table, Object attributes, Object values, String condition);
	public Object readContractor(Object tables, Object attributes, String condition);
Cover	public boolean createCover(String table, Object attributes, Object values);
	public boolean deleteCover(String table, String condition);
	public boolean updateCover(String table, Object attributes, Object values, String condition);
	public Object readCover(Object tables, Object attributes, String condition);
Line	public boolean createLine(String table, Object attributes, Object values);
	public boolean deleteLine(String table, String condition);
	public boolean updateLine(String table, Object attributes, Object values, String condition);
	public Object readLine(Object tables, Object attributes, String condition);
Modality	public boolean createModality(String table, Object attributes, Object values);
	public boolean deleteModality(String table, String condition);
	public boolean updateModality(String table, Object attributes, Object values, String condition);
	public Object readModality(Object tables, Object attributes, String condition);
Premium	public boolean createPremium(String table, Object attributes, Object values);
	public boolean deletePremium(String table, String condition);
	public boolean updatePremium(String table, Object attributes, Object values, String condition);
	public Object readPremium(Object tables, Object attributes, String condition);
Reinsured	public boolean createReinsured(String table, Object attributes, Object values);
	public boolean deleteReinsured(String table, String condition);
	public boolean updateReinsured(String table, Object attributes, Object values, String condition);
	public Object readReinsured(Object tables, Object attributes, String condition);
Reinsurance	public boolean createReinsurance(String table, Object attributes, Object values);
	public boolean deleteReinsurance(String table, String condition);
	public boolean updateReinsurance(String table, Object attributes, Object values, String condition);
	public Object readReinsurance(Object tables, Object attributes, String condition);

Risk_participant	public boolean createRisk_participant(String table, Object attributes, Object values);
	public boolean deleteRisk_participant(String table, String condition);
	public boolean updateRisk_participant(String table, Object attributes, Object values, String condition);
	public Object readRisk_participant(Object tables, Object attributes, String condition);

For the second iteration, we used the UML class diagram presented in Figure 23. Table 8 show the list of candidate services identified in the second iteration.

Table 8: Identified candidate services for the second iteration of [Baghdadi 2006].

Service	Operation
Adjustment_recovery	public boolean createAdjustment_recovery(String table, Object attributes, Object values);
	public boolean deleteAdjustment_recovery(String table, String condition);
	public boolean updateAdjustment_recovery(String table, Object attributes, Object values, String condition);
	public Object readAdjustment_recovery(Object tables, Object attributes, String condition);
Payment	public boolean createPayment(String table, Object attributes, Object values);
	public boolean delete Payment(String table, String condition);
	public boolean updatePayment(String table, Object attributes, Object values, String condition);
	public Object readPayment(Object tables, Object attributes, String condition);
Recovery_Request	public boolean createRecovery_Request(String table, Object attributes, Object values);
	public boolean deleteRecovery_Request(String table, String condition);
	public boolean updateRecovery_Request(String table, Object attributes, Object values, String condition);
	public Object readRecovery_Request(Object tables, Object attributes, String condition);

This method is only able to identify Entity services. The services identified have the same CRUD operations as the Entity services identified by MDCSIM. Nevertheless, Entity services identified by MDCSIM also have operations that maintain attributes of classes related to the master data in the logical model (*set_Risk_Distribution*, *update_Risk_Distribution*, *delete_Risk_Distribution*, *set_Contracted_Cover*, *update_Contracted_Cover*, *delete_Contracted_Cover* from *Reinsurance* service and *set_Retention, update_Retention, delete_Retention* from *Risk_Participant* service).

7.4.2 Service Identification Using DFDs (Data flow diagrams) [Yun et al. 2009]

The second method [Yun et al. 2009] identifies services using Data flow diagrams. Every process in a DFD can be identified as a service. A coarse-grained process corresponds to composite services, while a fine-grained process, which cannot be decomposed, corresponds to an atomic service. This method is classified as a Top-down approach covering only the business perspective.

For the first iteration, the DFDs presented on Figure 30 and Figure 31 were constructed. Figure 30 shows the first level of decomposition and Figure 31 shows the second level of decomposition where the process *Analyze Claim Notification* was decomposed into *Set Penalty* and *Send to Regulation*.

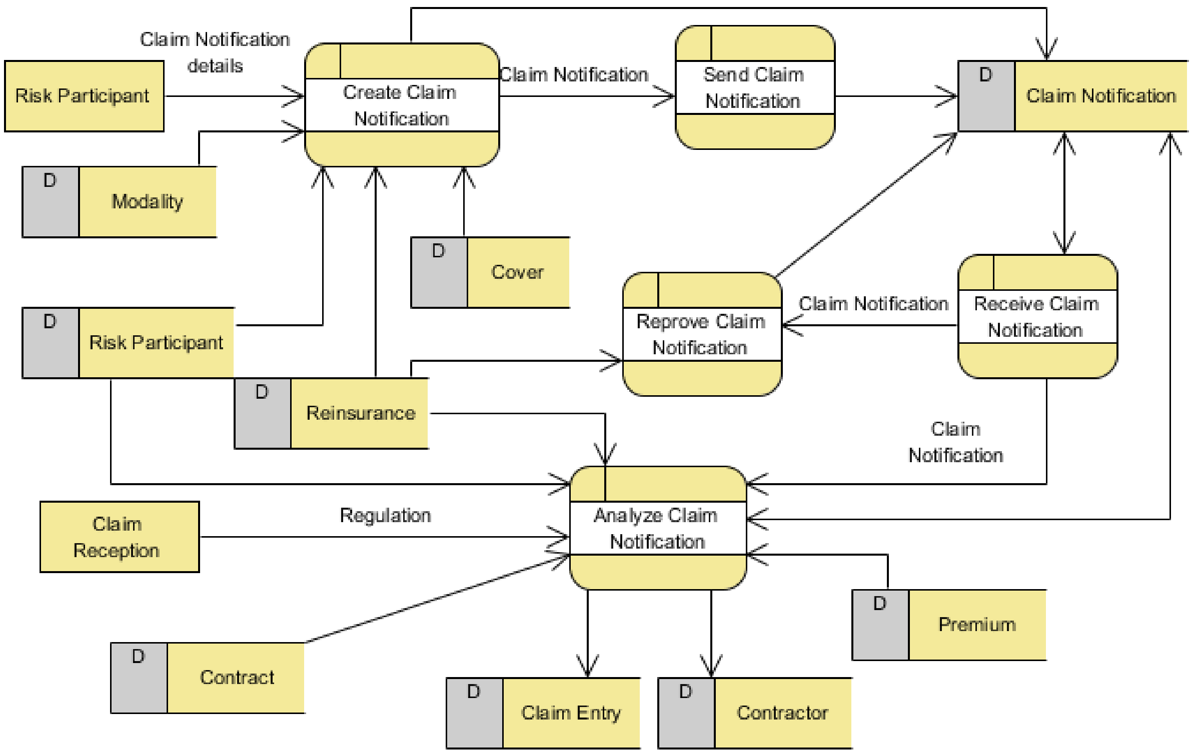

Figure 30: DFD – first level of the Claim Notification Reception process.

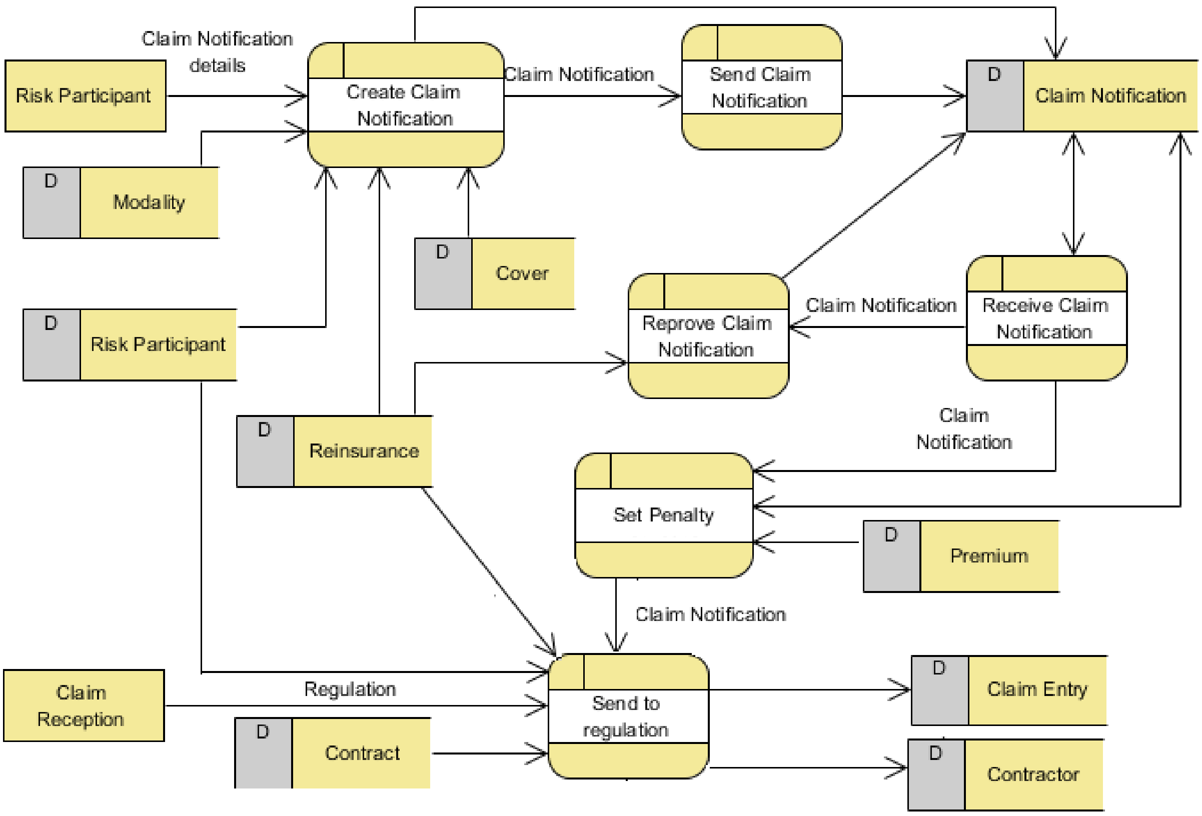

Figure 31: DFD – second level of the Claim Notification Reception process.

Each atomic process in the DFD - second level (Figure 31) derived a service (Task or Entity) and each data flow to a data store derived an Entity service that performs *create*, *update* or *read* operations. The process *Analyze Claim Notification* in Figure 30 also derived a Process service. Presents the complete list of services identified in this iteration.

Table 9: Identified candidate services for the first iteration of [Yun et al. 2009].

Service	**Type**	**Operation**
Create Claim_entry	Entity	public void createClaim_entry(Object class);
CreateClaim_notification	Entity	public void createClaim_notification(Object class);
UpdateClaim_notification	Entity	public void updateClaim_notification(int id, String values);
ReadClaim_notification	Entity	public Object readClaim_notification(int id);
ReadCover	Entity	public Object readCover(int id);
ReadModality	Entity	public Object readModality(int id);
ReadReinsurance	Entity	public Object readReinsurance(int id);
ReadPremium	Entity	public Object readPremium(int id);
ReadRisk_Participant	Entity	public Object readRisk_participant(int id);
ReadContract	Entity	public Object readContract(int id);
UpdateContractor	Entity	public void updateContractor(int id, String values);
SetPenality	Entity	public void set_Penalty (int id)
SendClaim_Notification	Task	public void sendClaim_Notification (int id, date emission_date)
ReceiveClaim_Notification	Task	public void receiveClaim_Notification (int id)

ReproveClaim_Notification	Task	public void reproveClaim_Notification (int id, date reproval_date)
SendClaim_Notification to regulation	Task	public void send_to_Regulation(int id, String regulation, date approval_date)
Analyze_Claim_Notification	Process	public void analyze_Claim_Notification(int id, String regulation)

For the second iteration, the DFD presented on Figure 32 was constructed. In this iteration, there was no need to decompose any process, thus Process services were not identified.

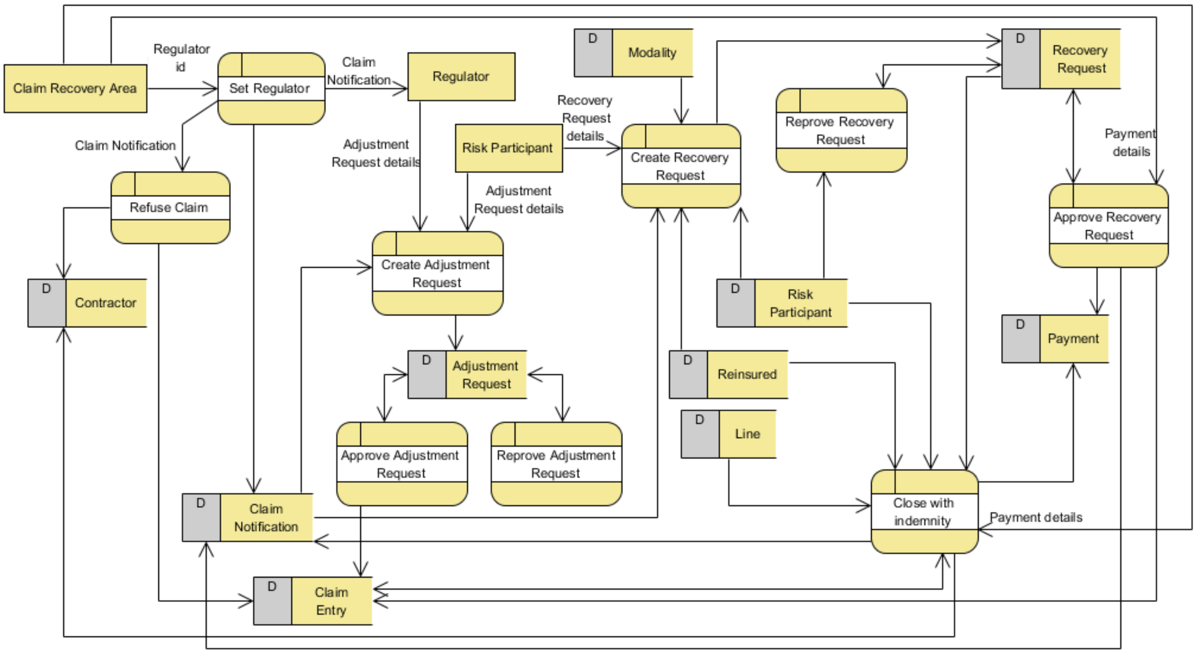

Figure 32: DFD of the Claim Recovery process.

Each process presented in Figure 32 was transformed into a service (Task or Entity). Data flows to data stores were also transformed into Entity services. Table 10 presents the list of services identified in this iteration.

Table 10: Identified candidate services for the second iteration of [Yun et al. 2009].

Service	**Type**	**Operation**
UpdateClaim_entry	Entity	public void updateClaim_entry(int id, String values);
CreateAdjustment_Request	Entity	public void createAdjustment_Request(Object class);
UpdateAdjustment_Request	Entity	public void updateAdjustment_Request(int id, String values);
ReadAdjustment_Request	Entity	public void readAdjustment_Request(int id);
CreateRecovery_Request	Entity	public void createRecovery_Request(Object class);
UpdateRecovery_Request	Entity	public void updateRecovery_Request(int id, String values);
ReadRecovery_Request	Entity	public void readRecovery_Request(int id);
ReadReinsured	Entity	public Object readReinsured(int id);

CreatePayment	Entity	public void createPayment(Object class);
ReadClaim_Entry	Entity	public Object readClaim_Entry(int id);
ReadLine	Entity	public Object readLine(int id);
Set_Regulator	Entity	public void set_Regulator(int id, int regulator);
ApproveAdjustment_Request	Task	public void approveAdjustment_Request(int id, int id_approver, date approval_date);
ReproveAdjustment_Request	Task	public void reproveAdjustment_Request(int id, int id_reprover, date reproval_date);
ApproveRecovery_Request	Task	public void approveRecovery_Request(int id, int id_approver, date approval_date);
ReproveRecovery_Request	Task	public void reproveRecovery_Request(int id, int id_reprover, date reproval_date);
Close_with_indemnity	Task	public void close_with_indemnity(int id, date closure_date);
Refuse_Claim	Task	public void refuse_Claim(int id, String justification);

This method was able to identify Entity services, Task services and Process services. However, since [Yun et al. 2009] does not adopt a service layer approach, some Entity services can deal with both behavioral and attribute maintenance (CRUD) concerns. Another disadvantage of [Yun et al. 2009] is that business rules cannot be made explicit in DFDs what is possible when using state diagrams.

7.5 Methods Comparison

In order to provide an initial assessment of MDCSIM, qualitative and quantitative comparison with [Baghdadi 2006] and [Yun et al. 2009] were accomplished. The qualitative comparison aims to discuss the types of services identified and their ability to cover the business domain. The quantitative comparison aims to assess the quality of the service portfolio identified by each SIM by measuring the granularity, coupling, cohesion and entity convergence quality attributes and also the reuse between iterations accomplished by each method.

7.5.1 Qualitative Comparison

MDCSIM and [Yun et al. 2009] were able to identify Entity services, Task services and Process services, supporting all the activities performed by the prioritized business areas. Nevertheless, [Baghdadi 2006] was able to identify only Entity services, which do not cover all the activities executed within the business domain. Table 11 shows the correspondence of the services identified by the three methods.

Table 11: Services identified by the three SIMs.

	MDCSIM		[Yun et al. 2009]	[Baghdadi 2006]	
Type	**Service**	**Operation**	**Service\Operation**	**Service**	**Operation**
Entity	Cover	Create		Cover	Create
		Update			Update
		Read	ReadCover		Read
		Delete			Delete
Entity	Premium	Create		Premium	Create
		Update			Update
		Read	ReadPremium		Read
		Delete			Delete
Task	Premium_ Controller	register_Premium_ Payment			
		register_Payment_ Delay			
Entity	Contractor	Create		Contractor	Create
		Update	UpdateContractor		Update
		Read			Read
		Delete			Delete
Task	Contractor_ Controller	mark_For_revision			
		suspend_Contractor			
		block_Contractor			
		register_Bankruptcy			
		unblock_Contractor			
		review_Contractor			
Entity	Reinsured	Create		Reinsured	Create
		Update			Update
		Read	ReadReinsured		Read
		Delete			Delete
Entity	Modality	Create		Modality	Create
		Update			Update
		Read	ReadModality		Read
		Delete			Delete
Entity	Line	Create		Line	Create
		Update			Update
		Read	ReadLine		Read
		Delete			Delete
Entity	Contract	Create		Contract	Create
		Update			Update
		Read	ReadContract		Read
		Delete			Delete
Task	Contract_ Controller	expire_Contract			
Entity	Reinsurance	Create			
		Update			
		Read	ReadReinsurance		
		Delete			
		set_Risk_Distribution			

		update_Risk_ Distribution			
		delete_Risk_ Distribution			
		set_Contracted_Cover			
		update_Contracted_ Cover			
		delete_Contracted_ Cover			
Task	Reinsurance_ Controller	expire_Reinsurance			
		aprove_Reinsurance			
		cancel_Reinsurance			
Process	create_ Endorsement _ Process	create_Endorsement			
Entity	Risk_ Participant	Create			
		Update			
		Read	ReadRisk_Participant		
		Delete			
		set_Retention			
		update_Retention			
		delete_Retention			
Entity	Recovery_ Request	Create	CreateRecovery_ Request	Recovery_ Request	Create
		Update	UpdateRecovery_ Request		Update
		Read	ReadRecovery_ Request		Read
		Delete			Delete
Task	Recovery_ Request_ Controller	send_Recovery_ Request			
		approve_Recovery_ Request	ApproveRecovery_ Request		
		reprove_Recovery_ Request	ReproveRecovery_ Request		
Entity	Claim_ Notification	Create	CreateClaim_ Notification	Claim_ Notification	Create
		Update	UpdateClaim_ Notification		Update
		Read	ReadClaim_ Notification		Read
		Delete			Delete
		set_Penalty	Set_Penality		
		set_Regulator	Set_Regulator		
Task	Claim_ Notification_ Controller	send_Claim_Notification	SendClaim_ Notification		
		reprove_Claim_ Notification	ReproveClaim_ Notification		
		receive_Claim_ Notification	ReceiveClaim_ Notification		

		send_to_Regulation	Send_to_Regulation		
		refuse_Claim	RefuseClaim		
		close_with_indemnity	close_with_ indemnity		
Process	Analyse_ Claim_ Notification_ Process	Analyse_ Claim_ Notification	Analyse_ Claim_ Notification		
Entity	Claim_Entry	Create	CreateClaim_Entry	Claim_Entry	Create
		Update	UpdateClaim_Entry		Update
		Read	ReadClaim_Entry		Read
		Delete			Delete
Task	Claim_ Entry_ Controller	suspend_Claim_Entry			
Entity	Adjustment_ Request	Create	CreateAdjustment_ Request	Adjustment _ Request	Create
		Update	UpdateAdjustment_ Request		Update
		Read	readAdjustment_ Request		Read
		Delete			Delete
Task	Adjustment_ Request_ Controller	approve_Adjustment_ Request	ApproveAdjustment_ Request		
		reprove_Adjustment_ Request	ReproveAdjustment_ Request		
Entity	Payment	Create	CreatePayment	Payment	Create
		Update			Update
		Read			Read
		Delete			Delete
Task	Payment_ Controller	confirm_Payment			

Table 11 shows that MDCSIM offered the best coverage to the analyzed business domain than the other 2 SIMs for the business scenario presented in Section 7.2. The services identified by MDCSIM have the same CRUD operations as the Entity services identified by [Baghdadi 2006] and also have operations that maintain attributes of classes related to the master data in the logical model (*set_Risk_Distribution*, *update_Risk_Distribution*, *delete_Risk_Distribution*, *set_Contracted_Cover*, *update_Contracted_Cover*, *delete_Contracted_Cover* from *Reinsurance* service and *set_Retention, update_Retention, delete_Retention* from *Risk_Participant* service). The services identified by MDCSIM also own operations that are not executed within the scope of the prioritized area, therefore are not explicit in the DFDs used in method [Yun et al. 2009], but are part of the master data lifecycle (*register_Premium_Payment,* *register_Payment_Delay* from *Premium_Cotroller* service; *mark_For_revision, suspend_Contractor, register_Bankruptcy, review_Contractor* from *Contractor_Controller* service; *expire_Contract* from *Contract_Controller* service; expire*_Reinsurance, aprove_Reinsurance, cancel_Reinsurance* from *Reinsurance_Controller* sevice; *create_Endorsement* from *Create_Endorsement_Process* service and *confirm_Payment* from *Payment_Controller* service).

The advantage of identifying the aforementioned operations is to provide service interfaces with wider scope, taking into account the ambit of the whole organization. In future iterations of the MDCSIM, the services that own these operations can be reused without the need to change their contracts. This situation was demonstrated in the second iteration of MDCSIM, when the operations *suspend_Claim_Entry* of the *Claim_Entry_Controller* service, *close_with_Indemnity* and *refuse_claim* of the *Claim_Notification_Controller* service and *unblock_contractor* of the *Contractor_Controller* service, identified in the first iteration, were used for the Claim recovery process.

Another disadvantage of [Yun et al. 2009] in relation to MDCSIM is that, since [Yun et al. 2009] does not adopt a service layer approach, some Entity services can deal with both behavioral and attribute maintenance concerns. Such situation makes those services less stable as business process changes frequently.

7.5.2 Quantitative Comparison

The service portfolios identified by each SIM were also compared by evaluating the reuse among iterations and the quality attributes of granularity, coupling, cohesion and entity convergence. These quality attributes were chosen because they are related with SOA goals of promoting business agility, increasing the ROI, and promoting business alignment as detailed in section 2.3 Service-Oriented Architecture Goals. Business agility is related to the flexibility to respond to business changes. Coupling covers business agility goal. A service with low coupling is more easily composed to support new requirements. Cohesion and entity convergence cover the ROI increasing, because services that encapsulate all actions of a business entity or are cohesive are more reusable, avoiding redundancy. Granularity covers the business alignment goal, because coarse-grained services offer rich functionality and have larger contribution to business processes. The evaluation of the aforementioned quality attributes was accomplished by using metrics proposed in [Ma et al. 2009]. The metrics are explained as follows.

7.5.2.1 Granularity

Service granularity refers to the scope of functionality implemented by a service. Therefore, service granularity is defined as the average number of operations that the identified services have. The formula of this metric is as follows [Ma et al. 2009]:

$$\vartheta granu = \frac{a}{s}$$

where

a total number of service operations in the service portfolio

s total number of services

Services should contain the highest number of operations as possible, defining a coarser interface that hides interaction details, avoiding multiple fine-grained interations. However, with the increase of granularity, the reusability of a service decreases.

7.5.2.2 Cohesion

Service cohesion refers to the degree of relation between the operations carried out by a service. Therefore, the degree of cohesion depends on the number of operations inside a service and on the dataflow between these operations as defined in the following formula [Ma et al. 2009]:

$$\vartheta cohes = \frac{\sum_{k=1}^{s}\left(\frac{1}{a_k} + \lambda^{-\frac{\overline{d_k}}{d_k}}\left(1 - \frac{1}{a_k}\right)\right)}{s}$$

where

d_i total complexity of data flows within the service

$k\left(d_k \geq 0,\ if\ d_k = 0, then\ \frac{d_k}{d_k} = \infty\right)$;

$\overline{d_k}$ expected value of the data flow complexity in the service

$k(\overline{d_k} \geq 0)$;

λ experiential variable to control the impact of data flows on the service cohesion ($\lambda > 1$).

The formulas for d_k and $\bar{d}_k$ are:

$$d_k = \sum_{i=1}^{a} \sum_{\substack{j=1 \\ i \neq j}}^{a} \left(x_{ik} x_{jk} \sum_{u=1}^{e_{ij}} c_{iju} \right)$$

$$\bar{d}_k = \sum_{i=1}^{a} \sum_{\substack{j=1 \\ i \neq j}}^{a} \left(x_{ik} x_{jk} \sum_{u=1}^{e_{ij}} \bar{c} \right)$$

x_i 1 if the operation i is partitioned into the service k;

0 if the operation i is not partitioned into the service k;

e_{ij} total number of information entities in the data flow from the operation i to j $(e_{ij} \geq 0)$;

c_{iju} relative complexity of the information entity u in the data flow from the operation i to j $(c_{iju} > 0)$;

$\bar{c}$ expected complexity of a data flow parameter, averaged over all data flow parameters in the considered operations ($\bar{c} > 0)$.

Cohesion should be the highest possible, because it increases the clarity and comprehension of identified services, therefore simplifying maintenance and future enhancements.

7.5.2.3 Coupling

Service coupling indicates the degree of interdependence between two services. An input of a service "A" can be obtained from an output of a service "B", thus resulting in the coupling of the service "A" with the service "B". As the requests to services are implemented through messages, the coupling degree is dependent of the number and the complexity of informational entities in the messages. The formula for coupling metric is as follows [Ma et al. 2009]:

$$\vartheta coupl = \sum_{l=1}^{s} \sum_{\substack{k=1 \\ k \neq l}}^{s} \sum_{t=1}^{m_{kl}} \sum_{u=1}^{e_{klt}} c_{kltu}$$

where

m total number of messages sent from the service k to l $(m_{kl} \geq 0)$;

e_{klt} total number of information entities in the message t sent from the service k to l $(e_{klt} \geq 0)$;

c_{kltu} relative complexity of the information entity u in the message t sent from the service k to l $(c_{kltu} > 0)$.

Coupling should be preferably low in order to minimize the interdependence between services. With the increase of coupling, it is more difficult to reuse a service.

7.5.2.4Entity Convergence

Business entity convergence is the extent to which a service focuses on processing operations of a specific business entity. Business entities provide a natural way to do a partitioning of business activities into services. The formula for business entity convergence is as follows [Ma et al. 2009]:

$$\vartheta conve = \frac{1}{s} \sum_{k=1}^{s} N\left[\bigcup_{i=1}^{a} B_{ik}\right] + \frac{1}{b} \sum_{l=1}^{b} N\left[\bigcup_{j=1}^{a} S_{jl}\right]$$

where

b total number of business entities (b > 0);

B the set of business entities processed by the operation i included in the service k;

S_{jl} the set of services which involve the operation j to process the business entity l;

$N[.]$ Count the number of elements in a set.

A service encapsulating all the actions on a same business entity favors reuse because of its clear “business” functionality, thus entity convergence should be low.

To calculate the metrics, we assume that the relative complexities of the information entities are equal to the number of relationships their corresponding class have in the logical model. Complexity of Premium entity is 1, Cover entity complexity is 2, complexities of entities Contract, Contractor and Line are 3, complexities of entities Claim Entry and Adjustment request are 4, complexity of Modality entity is 5, complexities of entities Risk participant and Reinsured are 6, Recovery request entity complexity is 7, Payment entity complexity is 9 and complexities of entities Claim notification and Reinsurance are 12. For calculating service cohesion, we assume that the impact of data flows on the service cohesion (λ) is equal to 2 as in [Ma et al. 2009]. We also assume that the total number of activities is the total number of operations identified by each approach, as we are not working with process decomposition. Utility services were also not evaluated by the metrics, because the analyzed methods do not support their identification. The results of the metrics for each set of services identified by the three approaches are summarized in Table 12.

Table 12: Metrics value.

	Granularity	**Coupling**	**Cohesion**	**Entity convergence**
MDCSIM	3,76	307,00	0,76	1,56
[Yun et al. 2009]	1,00	478,00	1,00	2,20
[Baghdadi 2006]	4,00	134,00	0,57	2,00

The best values of granularity and coupling were obtained by [Baghdadi 2006]. Nevertheless, this method does not identify the Task services necessary to accomplish the Claim processing area activities, neither operations that maintain attributes of classes related to the master data in the logical model. The absence of Task services in this portfolio is also a reason for the low value of coupling, because this kind of service usually depends of the Entity services. Services identified in [Yun et al. 2009] had the maximum value of cohesion, because they have only one operation, thus granularity was low. The proliferation of fine-grained services is undesirable, since it increases service governance efforts. Coupling is very high because Task services depend on many Entity services as CRUD operations are scattered into several services.

The service portfolio identified by MDCSIM has the best quality since it achieves a balance among the different metrics. Our service portfolio had the best value of entity convergence and the second best values of granularity, coupling and cohesion. The best value of entity convergence was expected, because MDCSIM gathers operations accordingly to the master data manipulated by them. The value of cohesion metric shows that gathering operations by master data and by type of the operation (maintenance operation or behavioral operations) contributed to provide cohesive services. Good results in entity convergence and in granularity metrics suggest that MDCSIM is able to provide coarse-grained services that tend to be more reused since they (i) provide functionality of a business entity (providing a larger contribution to business processes) and (ii) avoid the multiple fine-grained interactions (several data flows are inside of services).

Analyzing service reuse between iterations, MDCSIM also had the best performance: 9 services identified in the first iteration were reused in the second iteration (*Claim_Entry, Claim_Notification, Claim_Notification_Controller, Contractor_Controller, Line, Modality, Reinsurance, Reinsured and Risk_Participant*). Among the services identified by [Baghdadi 2006] only 6 services were reused in the second iteration (*Claim_Entry, Claim_Notification, Line, Modality, Reinsured* and *Risk_Participant*). Finally, in [Yun et al. 2009] 6 services identified in the first iteration were reused in the second iteration (CreateClaim_Entry, *UpdateContractor, UpdateClaim_Notification, ReadClaim_Notification, ReadRisk_Participant* and *ReadModality)*. Providing reusable services is an important feature to SIMs in order to accomplish SOA goals of increase organizational agility and increase ROI.

7.6 Threats to the Validity of the Proof of Concept

The aforementioned comparison is a simplified analysis that aims to provide an initial assessment on the capacities of MDCSIM to support SOA goals by promoting reuse among services and to identify services that deal with business and IT perspectives. It does not intend to be an experiment. We understand that this analysis has some weakness as follows:

- Service identification was done only in one business scenario. Comparison should be repeated in other business scenarios for more evidences of the method validity.
- All steps were executed by the same person, thus eliminating communication issues among the several roles involved in the proposed method. The execution of MDCSIM should be repeated using different people for each role.

- It was not possible to compare the time and efforts to execute the SIMs, neither to assess the perception of MDCSIM utilization and learning. This kind of comparison is also a gap in SIM field. A controlled experiment should be done with groups of specialists and non-specialists in order to compare their performance in identifying services using MDCSIM and using other SIMs.

7.7 Chapter Summary

This chapter presented the steps used to provide an initial assessment of MDCSIM. Initially a proof of concept of MDCSIM in a real business scenario was depicted, showing the inputs and the outputs generated for each MDCSIM step using MDCSIM plug-in. Two iterations were run in order to demonstrate service reuse. Next, other two data-focused SIM were tested in the same business scenario and a qualitative comparison of the SIMs was provided. The granularity, coupling, cohesion and entity convergence quality attributes of the service portfolios identified by each SIM and the reuse between iterations were measured and compared. Finally, some threads to MDCSIM validity were discussed.

Conclusion

This dissertation presented a critical analysis of several service identification methods published from 2002 until June 2013, identifying the drawbacks in SIM field. In order to address some of the identified drawbacks, a method (MDCSIM) and a tool (MDCSIM plug-in) were proposed. MDCSIM is a Meet in the middle method that identifies services using as inputs master data and logical data models. Artifact-centric modeling technique is used within MDCSIM in order to identify master data lifecycle. MDCSIM plug-in is a MDA tool that automates the steps 5.2.4 Entity Services Design, 5.2.5 Artifact-Centric Model Creation and 5.2.6 Task Services and Process Services Design of MDCSIM. MDCSIM plug-in reads a UML class diagram (that represents master data attributes and relationships) and the UML state machine diagrams (that represent master data lifecycle) and identifies a list of Entity, Task and Process candidate services.

A proof of concept with MDCSIM was accomplished using a real business scenario and some qualitative and quantitative comparisons with other two data-focused SIMs were performed. The proof of concept demonstrated the usage of master data and of the artifact-centric modeling technique to describe Claim processing business area and the business services (Task and Process services) identified from this business description. Such demonstration suggests that master data can be used as an alternative input to business process to elicit business perspective and identify business services as stated in the first hypothesis presented in Chapter 1.

During the proof of concept, master data attributes were identified from databases of the Integrated Business System, of the Claim List System and of the Guarantee of Contractual Obligations System. A layer of services (Entity services) that abstracts data storage and ensures data integrity was also identified from the logical data models. Such identification indicates that IT perspective can be elicited by correlating master data with the logical data models obtained from organization databases and the software services identified, as stated in the second hypothesis depicted in Chapter 1.

The method configurability stated in the third hypothesis was demonstrated in the proof of concept by slicing candidate services' identification into 2 steps. First, only Entity services were identified for the business domain prioritized as presented in Figure 14. Next, Task and Process services were identified in order to compose the complete set of services for the domain as presented in Figure 22. If an organization wants a “lightweight” version of MDCSIM, it can perform iterations in order to identify only Entity services (steps 5.2.1 to 5.2.4), and in the future execute steps 5.2.5 and 5.2.6 to identify Process and Task services. This feature is important for organizations with low maturity in SOA or organizations that need to quickly deliver results.

Service quality attributes assessment, stated in the fourth hypothesis, was addressed by using the metrics proposed by [Ma et al. 2009] to assess granularity, coupling, cohesion, and entity convergence service quality attributes as presented in section 7.5.2 Quantitative Comparison. This comparison suggests that MDCSIM identified services with more quality than [Baghdadi 2006] and [Yun et al. 2009], since it achieved a balance among the different metrics. MDCSIM portfolio had the best value of entity convergence and the second best values of granularity, coupling and cohesion. Good results in such metrics suggest that services identified by MDCSIM contribute to achieve SOA goals of promoting business agility, increasing the return of investment (ROI), and promoting business alignment, since they are coarse-grained services that provide larger contribution to business processes, avoid multiple fine-grained interactions and have clear responsibility.

The comparison with [Baghdadi 2006] and [Yun et al. 2009] also revealed that MDCSIM offered the best coverage to the analyzed business domain. MDCSIM was able to identify the operations necessary to fulfill business tasks and also identify operations that are not executed within the scope of the prioritized business area, but are part of the master data lifecycle. This characteristic affords services with wider interfaces than those identified by the other 2 SIMs, taking into account the scope of the whole organization. In subsequent iterations of the MDCSIM, the services that own these operations can be reused without the need to change their contracts, what is not accomplished by [Baghdadi 2006] and [Yun et al. 2009]. Finally, MDCSIM has had more services identified in the first iteration reused in the second iteration than the other two SIMs.

The aforementioned facts, suggests that MDCSIM can support the service identification phase in a service-oriented modelling and architectural design process.

Nevertheless future work can be done in order to extend MDCSIM improving the following aspects:

- Support for the specification and realization phases: MDCSIM includes steps and a tool to aid service identification. Both the tool and the method can be extended to support the specification and realization phases covering the complete lifecycle of the service design;
- Identification of Utility services: The artifacts used and produced by MDCSIM can also support Utility services identification, but some steps, such as non-functional requirements elicitation or analysis of cross-cutting concerns not related to business in the identified services, need to be added to MDCSIM in order to enable Utility services identification;
- Extraction of business rules from stored procedures: Stored procedures comprise business rules that can be exposed as services, or to be part master data lifecycle. Some research can be done in order to promote the analysis of stored procedures in MDCSIM;
- Non-functional requirements elicitation: Non-functional requirements can originate Utility services or be fulfilled by Entity, Task and Process services. Steps to elicit non-functional requirements and guidelines to define if they should be exposed as an Utility service or to be part of Entity, Task and Process services can be included in MDCSIM;
- Economical aspects evaluation: Aspects related to service creation and maintenance can be used to decide which services can bring more benefits to the business.

References

ABDELAZIZ, E. F., MUSTAPHA, A., MOHAMMED, S. A service-oriented approach for information systems development, Multimedia Computing and Systems, ICMCS (2011) 1-6.

ABDELKADER, M. MALKI, M., BENSLIMANE, S. M. A heuristic approach to locate candidate web service in legacy software, International Journal of Computer Applications in Technology 47 (2013) 152-161.

ABU-MATAR, M., GOMAA, H., Variability Modeling for Service Oriented Product Line Architectures, Software Product Line Conference (SPLC), 2011 15th International, (2011) 110-119.

ACM Digital Library. http://dl.acm.org. Last access: December 2013.

ADAMOPOULOS, D. X., PAVLOU, G., PAPANDREOU, C. A. Advanced service creation using distributed object technology, IEEE Communications Magazine 40, 3 (2002) 146-154.

AIER, S. How clustering enterprise architectures help to design service oriented architectures, Proceedings - 2006 IEEE International Conference on Services Computing, SCC (2006) 269-272.

AL BELUSHI, W., BAGHDADI, Y. An Approach to Wrap Legacy Applications into Web Services, Service Systems and Service Management, 2007 International Conference on (2007) pp.1,6, 9-11.

ALAHMARI, S., DE ROURE, D., ZALUSKA, E. A Model-Driven Architecture Approach to the Efficient Identification of Services on Service-oriented Enterprise Architecture, Enterprise Distributed Object Computing Conference Workshops (EDOCW) (2010) 165-172.

ALIZADEH, K., SEYYEDI, M. A., MOHSENZADEH, M. Mapping service concept and enterprise ontology in service identification, Networked Computing (INC) (2011) 22-27.

ALONSO, G. et al. Functionalities and Limitations of Current Workflow Management Systems, IEEE Expert 12, 5 (1997).

ANDERSSON, B., JOHANNESSON, P., ZDRAVKOVIC, J. Aligning goals and services through goal and business modeling, Information Systems and E-Business Management 7 (2) (2009) 143-169.

ARSANJANI, A. Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA. 2005 In: http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/ Last access: December 2013.

ARSANJANI, A. et al. SOMA: A method for developing service-oriented solutions, IBM Systems Journal 47 (3) (2008) 377-396.

ASADI, M. et al. Developing families of method-oriented architecture, IFIP Advances in Information and Communication Technology, 351 AICT (2011) 168-183.

ASADI, M. et al. Model-driven development of families of service-oriented architectures, ACM International Conference Proceeding Series (2009) 95-102.

AVERSANO, L., CERULO, L., PALUMBO, C. Mining candidate web services from legacy code, Web Site Evolution WSE (2008) 37-40.

BAGHDADI, Y. Reverse engineering relational databases to identify and specify basic Web services with respect to service oriented computing, Information Systems Frontiers 8 (5) (2006) 395-410.

BAO, L. et al. Extracting reusable services from legacy object-oriented systems, Software Maintenance (ICSM), IEEE International Conference, (2010) 1-5.

BELL, M. SOA modeling patterns for service-oriented discovery and analysis, first ed., John Wiley & Sons Inc., 2010.

BHATTACHARYA, K. et al. A Data-centric Design Methodology for Business Processes, Handbook of Research on Business Process Management, Information Science Publishing, Hershey, PA, 2009.

BIANCHINI, D. P2S: A methodology to enable inter-organizational process design through web services, International Conference on Advanced Information Systems (2009) 334-348.

BIANCHINI, D. et al. Service Identification in Inter-Organizational Process Design, IEEE Transactions on Services Computing (2013).

BIRKMEIER D., KLÖCKNER S., OVERHAGE S. A Survey of Service Identification Approaches - Classification Framework, State of the Art, and Comparison. In: Enterprise Modeling and Information Systems Architectures vol 4, S.20-36, 2009.

BOERNER R., GOEKEN M. Service identification in SOA governance literature review and implications for a new method. In: Int. Conference on Digital Ecosystems and Technologies, pp. 588–593, 2009.

BRADFORD, R. Efficient Discovery of New Information in Large Text Databases. Proceedings, IEEE ISI, Vol. 3495, pp. 374–380, 2005.

BRZOSTOWSKI, K. et al. Service discovery in the SOA system, Proceedings of the Second international conference on Intelligent information and database systems: Part II (ACIIDS'10) (2010) 29-38.

CAETANO, A., SILVA, A. R., TRIBOLET, J. Identification of Services through Functional Decomposition of Business Processes, Business Information Systems Lecture Notes in Business Information Processing 47 (5) (2010) 144-157.

CAI S., LIU Y., WANG X. A Survey of Service Identification Strategies. In: International Conference on Asia-Pacific Services Computing Conference, pp.464-470, 2011.

CANFORA, G. et al. A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures, Journal of Systems and Software 81 (4) (2008) 463-480.

CHAARI, S. et al. Towards a service-oriented enterprise based on business components identification, Enterprise Interoperability II (2007) Part V, 495-506.

CHANG, S. H. A Systematic Analysis and Design Approach to Develop Adaptable Services in Service Oriented Computing, In: Services, 2007 IEEE Congress on, 2007, 375-378.

CHEN, F. el al. Feature analysis for service-oriented reengineering, Software Engineering Conference, APSEC '05. 12th Asia-Pacific (2005) 8 pp.

CHEN, F. et al. Service Identification via Ontology Mapping, Computer Software and Applications Conference, COMPSAC '09 33rd Annual IEEE International (2009) 486-491.

CHIANG, R.H.L. et al. Reverse engineering of relational databases: Extraction of an EER model from a relational database, Data & Knowledge Engineering 12 (2), 1994, pp. 107-142

CHO, M. et al. Service identification and modeling for service oriented architecture applications, Proceedings of the 7th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (2008) 193-199.

CHUNG, S. et al. Service-oriented reverse reengineering: 5W1H model-driven re-documentation and candidate services identification, Service-Oriented Computing and Applications, SOCA (2009) 1-6.

COHN, D., HULL, R. Business artifacts: A data-centric approach to modeling business operations and processes. IEEE Data Eng. Bull. 32(3), pp.3-9, 2009.

DBLP Computer Science Bibliography. http://www.dblp.org/search/index.php. Last access: December 2013.

DE BRUIN, J. et al. On the Design of Knowledge Discovery Services Design Patterns and Their Application in a Use Case Implementation, Leveraging Applications of Formal Methods, Verification and Validation, Communications in Computer and Information Science 17 (2009) p. 649.

DEMIRKAN H. et al. Service-oriented technology and management: Perspectives on research and practice for the coming decade. Electronic Commerce Research and Applications vol. 7, issue 4 pp. 356-376, 2008.

DINH, T. L., NGUYEN-NGOC, A. V. A conceptual framework for designing service-oriented inter-organizational information systems, Proceedings of the 2010 Symposium on Information and Communication Technology, ACM (2010) 147-154.

DOMINIK, Q. et al. Alignment of Business and IT Architectures in the German Federal Government: A Systematic Method to Identify Services from Business Processes, Proceedings of the 2013 46th Hawaii International Conference on System Sciences (2013) 3848-3857.

DREIBELBIS A. et al. Enterprise master data management: An SOA approach to managing core information. Upper Saddle River, NJ: IBM Press, 2008.

DWIVEDI, V., KULKARNI, N. A Model Driven Service Identification Approach for Process Centric Systems, Congress on Services Part II (2008) 65-72.

ENDREI, M. et al. Patterns: Service-Oriented Architectures and Web Services. IBM Redbooks, 2004. 348p.

ERL T. SOA Principles of Service Design. Prentice Hall, 2007.

ERRADI, A., ANAND, S., KULKARNI, N. SOAF: An Architectural Framework for Service Definition and Realization, Services Computing, SCC '06 IEEE International Conference (2006) 151-158.

ERRADI, A., KULKARNI, N., MAHESHWARI, P. Service Design Process for Reusable Services: Financial Services Case Study, Proceedings of the 5th international conference on Service-Oriented Computing (2007) 606-617.

FAREGHZADEH, N. Service Identification Approach to SOA Development, World Academy of Science, Engineering and Technology 45 (2008).

FLAXER, D., NIGAM, A. Realizing Business Components, Business Operations and Business Services, Proceedings of the E-Commerce Technology for Dynamic E-Business, IEEE International Conference (2004) 328-332.

FUHR, A. et al. Model-driven software migration into service-oriented architectures, Computer. Science, 28 (2013) 65-84.

GACITUA-DECAR, V., LERO, C. P. Automatic Business Process Pattern Matching for Enterprise Services Design, Services - II, SERVICES-2 '09 World Conference (2009) 111-118.

GORDIJN, G. J., YU, E., VAN DER RAADT, B. E-Service Design Using i* and e3 value Modeling, IEEE Software 23 (2006) 26-33.

GU Q., LAGO P. Service Identification Methods: A Systematic Literature Review. In: Service Wave, pp. 37-50, 2010.

GUAN, Q., FENG, S., MA, Y. A Network Topology Clustering Algorithm for Service Identification, Proceedings of the 2012 International Conference on Computer Science and Service System (2012) 1583-1586.

GUIZZARDI, G. Ontological foundations for structural conceptual models. PhD thesis, University of Twente, Enschede, The Netherlands. CTIT Ph.D.-thesis series No. 05-74 ISBN 90-75176-81-3 (2005).

HAN F., MOLLER E., BERRE A. J. Organizational Interoperability Supported through Goal Alignment with BMM and Service Collaboration with SoaML. In: Interoperability for Enterprise Software and Applications China, International Conference on, pp. 268-274, 2009.

HUANG, C. LEE, G. M., CRESPI, N. A semantic enhanced service exposure model for a converged service environment, Communications Magazine, IEEE, 50 (3) (2012) 32-40.

HUAYOU, S. et al. A Service-Oriented Analysis and Modeling Using Use Case Approach, Computational Intelligence and Software Engineering, CiSE (2009) 1-6.

IEEE Xplore. http://ieeexplore.ieee.org. Last access: December 2013.

ILAYPERUMA, T., ZDRAVKOVIC, J. Exploring business value models from the inter-organizational collaboration perspective, Proceedings of the 2010 ACM Symposium on Applied Computing, SAC'10 (2010) 99-105.

INAGANTI, S., BEHARA, G. K. Service Identification: BPM and SOA Handshake (2007) In: http://w.bptrends.com/publicationfiles/THREE%2003-07-ART-BPMandSOAHandshake-InagantiBehara-Final.pdf Last accessed: December 2013

JABREF In: http://jabref.sourceforge.net/ Last access: December 2013.

JAIN, H., ZHAO, H., CHINTA, N. R. A Spanning Tree Based Approach to Identifying Web Services, International Journal of Web Services Research 1 (2004) 1-20.

JAMSHIDI, P. et al. Toward automatic transformation of enterprise business model to service model, Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, IEEE Computer Society (2009) 70-74.

JAMSHIDI, P., SHARIFI, M., MANSOUR, S. To Establish Enterprise Service Model from Enterprise Business Model, Services Computing, SCC '08. IEEE International Conference (2008) 93-100.

JIN, Z., ZHU, H. A framework for agent-based service-oriented modeling, Proceedings of the 4th IEEE International Symposium on Service-Oriented System Engineering, SOSE (2008) 160-165.

JOUAULT, F., KURTEV, I. Transforming models with ATL, Proceedings of the 2005 international conference on Satellite Events at the MoDELS, MoDELS'05 (2005), 128-138.

KAABI, R. S., SOUVEYET, C., ROLLAND, C. Eliciting service composition in a goal driven manner, Proceedings of the 2nd international conference on Service oriented computing, ACM (2004) 308-315.

KANG, D., SONG, C., BAIK, D. A Method of Service Identification for Product Line, Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology 2 (2008) 1040-1045.

KANNAN, K., SRIVASTAVA, B. Promoting Reuse via Extraction of Domain Concepts and Service Abstractions from Design Diagrams, Services Computing SCC '08. IEEE International Conference (2008) 265-272.

KAZEMI, A. et al. A Genetic Algorithm Based Approach to Service Identification, Proceedings of the 2011 IEEE World Congress on Services (2011a) 339 -346.

KAZEMI, A. et al. A metric suite for measuring service modularity, Computer Science and Software Engineering (CSSE) (2011b) 95-102.

KENZI, A. et al. Multi-functional service oriented system development for user-driven adaptability, Information and Communication Technologies: From Theory to Applications, ICTTA 2008 3rd International Conference, (2008) 1-7.

KIM, G., SUHH, Y. Ontology-based semantic matching for business process management, ACM SIGMIS Database 41 (4) (2010) 98-118.

KIM, S., KIM, M., PARK, S. Service Identification Using Goal and Scenario in Service Oriented Architecture, Software Engineering Conference, APSEC '08. 15th Asia-Pacific (2008) 419-426.

KIM, Y., DOH, K. Formal Identification of Right-Grained Services for Service-Oriented Modeling, Proceedings of the 10th International Conference on Web Information Systems Engineering (2009) 261-273.

KIM, Y., DOH, K. The service modeling process based on use case refactoring, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2007) 108-120.

KIM,Y., DOH, K. Pragmatic granularity decision for right-grained services in service-oriented modeling, International Journal Web Grid Services 8, 2 (2012) 111-133.

KIM, Y. DOH, K. Use-case driven service modeling with XML-based tailoring for SOA, International Journal of Web and Grid Services 9, 1 (2013) 35-53.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering – A systematic literature review, Information and Software Technology 51 (1) (2009) 7-15.

KLOSE, K., KNACKSTEDT, R., BEVERUNGEN, D. Identification of Services - A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning, Proceedings of the 15th European Conference on Information Systems, ECIS (2007) paper 116.

KOHLBORN T. et al. Service analysis - A critical assessment of the state of the art. In: Proceedings of ECIS'2009, pp.1583-1594, 2009.

KOHLMANN, F., ALT, R. Business-Driven Service Modeling - A Methodological Approach from the Finance Industry, SABRE 2007 (2007) pp. 14.

KOMONDOOR, R. et al. Identifying services from legacy batch applications, Proceedings of the 5th India Software Engineering Conference (2012) 13-22.

KRAFZIG, D., BANKE, K., SLAMA, D. Enterprise SOA. Service-Oriented Architecture Best Practices. Fifth ed. Prentice Hall, 2004.

KUMARAN, S. et al. On the Duality of Information-Centric and Activity-Centric Models of Business Processes, Proceedings of the 20th international conference on Advanced Information Systems Engineering, CAiSE '08 (2008), 32-47.

LANUSSE, A. et al. Papyrus UML: an open source toolset for MDA, Model driven architecture - foundations and applications: 5th European conference, ECMDA-FA (2009), 23-26.

LEE, J., MUTHIG, D., NAAB, M. A feature-oriented approach for developing reusable product line assets of service-based systems, Journal of Systems and Software 83 (2010) 1123-1136.

LEE, J. et al. An Approach for Service Identification Using Value Co-creation and IT Convergence, Computers Networks Systems and Industrial Engineering, CNSI (2011) 441-446.

LEE, T. B., HENDLER, J., ORA, L. The Semantic Web, The Scientific American, May 2001.

LI, S., TAHVILDARI, L. A service-oriented componentization framework for Java software systems, Proceedings - Working Conference on Reverse Engineering, WCRE (2006) 115-124.

LI, S., TAHVILDARI, L. E-BUS: a toolkit for extracting business services from java software systems, Companion of the 30th international conference on Software engineering, ACM (2008) 961-962.

LI, D. Conceptual Workflow Verification and Optimization for Artifact-Centric Business Process, Proceedings of the 2009 International Forum on Information Technology and Applications, IFITA '09 (2009), Vol. 3, 142-146.

LIU, W. et al. Use Case-Based Service-Oriented Analysis and Modeling, Internet Computing Information Services ICICIS (2011) 94-96.

LO, A., YU, E. From business models to service-oriented design: a reference catalog approach, Proceedings of the 26th international conference on Conceptual modeling ER'07 (2007) 87-101.

LOSHIN D. Master data management. Burlington, MA: Morgan Kaufmann 2008.

MA, Q. et al. Evaluating Service Identification with Design Metrics on Business Process Decomposition, Proceedings of the 2009 IEEE International Conference on Services Computing (SCC '09), pp. 160-167,

MANI, S. et al. Using User Interface Design to Enhance Service Identification, Web Services ICWS '08, IEEE International Conference (2008) 78-87.

MEDEIROS, F. M., DE ALMEIDA, E.S., de LEMOS MEIRA, S. R., Designing a set of service-oriented systems as a Software Product Line, Proceedings - 4th Brazilian Symposium on Software Components, Architectures and Reuse, SBCARS (2010) 70-79.

MEERTENS, L. O., IACOB, M. E., NIEUWENHUIS, L. J. M. Goal and model driven design of an architecture for a care service platform, Proceedings of the 2010 ACM Symposium on Applied Computing, ACM (2010) 158-164.

MENASCÉ, D. A., CASALICCHIO, E., DUBEY, V. A heuristic approach to optimal service selection in service oriented architectures, Proceedings of the 7th international workshop on Software and performance, ACM (2008) 13-24.

MOSSER, B. et al. From aspect-oriented requirements models to aspect-oriented business process design models: An iterative and concern-driven approach for software engineering, Proceedings of the 10th International Conference on Aspect-Oriented Software Development, AOSD.11 (2011) 31-42.

NAKAMURA, M. et al. Identifying Services in Procedural Programs for Migrating Legacy System to Service Oriented Architecture, International Journal of Information Systems in the Service Sector, Vol. 3 (2011) 54-72.

NGUYEN, D. K. et al. Gap analysis methodology for business service engineering, 2009 IEEE Conference on Commerce and Enterprise Computing, CEC (2009) 215-220.

NIGAM, A., CASWELL, N.S. Business artifacts: An approach to operational specification, IBM Systems Journal 42, pp. 428–445, 2003.

OASIS Reference Architecture Foundation for Service Oriented Architecture. (2012) http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-v1.0.pdf. Last access: December 2013.

OMG Object Management Group, MDA Guide V1.0.1, http://www.omg.org/cgi-bin/doc?omg/03-06-01, (2003). Last accessed: December 2013.

OMG Object Management Group; Object Constraint Language Specification, Version 2.3.1, January 2012. http://www.omg.org/spec/OCL/2.3.1/PDF, Last accessed: January 2014.

PAPAZOGLOU, M. P. Service-Oriented Computing: Concepts, Characteristics and Directions. In Proceedings of the International Conference on Web Information Systems Engineering (WISE),4, 2003, pp.3-12.

PARK, J. et al. An approach to enhancing reusabilities in service development, Proceedings of the 2009 International Conference on Hybrid Information Technology (2009) 143-150.

PATIG, S., WESENBERG, H. Role of process modeling in software service design, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2009) 420-428.

PONNALAGU K., NARENDRA N. C. Deriving service variants from business process specifications. In: Proceedings of the 1st Bangalore Annual Compute Conference (COMPUTE '08). Article 4 , 9 pages, 2008.

QI, H., HUIFANG, L. Optimizing for ArtiFlow and Transform ArtiFlow to ServiceFlow, Proceedings of the 2011 Seventh International Conference on Computational Intelligence and Security, CIS '11 (2011), 578-582.

REN, M., WANG, Y. Rule based business service identification using UML analysis, Information Management and Engineering (ICIME) (2010) 199-204.

RICCA, F., MARCHETTO, A. A "quick and dirty" meet-in-the-middle approach for migrating to SOA, Proceedings of the joint international and annual ERCIM workshops on Principles of software evolution (IWPSE) and software evolution (Evol) workshops, ACM (2009) 73-78.

ROSEN, M. et al. Applied SOA: Service-Oriented Architecture and Design Strategies. New Work, Wiley, 2008.

ROSTAMPOUR A. et al. A metric suite for measuring service alignment with business agility. In: Computer Science and Software Engineering (CSSE), pp. 121 – 128, 2011.

RUIZ, M., PELECHANO, V., Pastor, O. Designing web services for supporting user tasks: A model driven approach, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4231 LNCS (2006) 193-202.

SAMAVI, R., YU, E., TOPALOGLOU, T. Applying strategic business modeling to understand disruptive innovation, Proceedings of the 10th international conference on Electronic commerce, ACM, (2008) 15:1-15:10.

SCHMIDT, R. A Service-System Based Identification of Meta-services for Service-Oriented Enterprise Architecture, Enterprise Distributed Object Computing Conference Workshops, EDOCW (2011) 293-300.

SCOPUS http://www.scopus.com. Last access: December 2013.

SEWING, J., ROSEMANN, M., DUMAS, M. Process-Oriented Assessment of Web Services, International Journal of E-Business Research, 2, 1 (2006) 19-44.

SHAN, T. C., HUA, W. W. “Service-Oriented solution framework for internet banking”, IJWSR 3.1 pp.29-48, 2006.

SHIANG, W., Rau, H., Lin, Y. Service identification of a collaborative design workflow in a dynamically changing environment, Networking, Sensing and Control, ICNSC '09 (2009) 685-690.

SHIM, B. et al. A Design Quality Model for Service-Oriented Architecture. In Proceedings of the 2008 15th Asia-Pacific Software Engineering Conference (APSEC '08). IEEE Computer Society, Washington, DC, USA, 403-410.

SHIRAZI, H. M., FAREGHZADEH, N., SEYYEDI, A. A combinational approach to service identification in SOA, Journal of Applied Sciences Research (2009) 1390-1397.

SOUZA, A. et al. Service identification in aspect-oriented business process models, Service Oriented System Engineering (SOSE) IEEE 6th International Symposium (2011) 164-174.

SOUZA, A. et al. A method for service identification from business process models in a SOA approach, Lecture Notes in Business Information Processing (2009) 99-112.

STEWART, G., CHAKRABORTY, A. Service identification through value chain analysis and prioritization, Proceedings of the 16th Americas Conference on Information Systems (2010).

STROSNIDER, J. et al. A. Model-driven synthesis of SOA solutions. IBM Systems Journal 47 (3) pp. 415-432, 2008.

UPADHYAYA, B., KHOMH, F., ZOU, Y. Extracting RESTful services from Web applications, Proceedings of the 2012 5th IEEE International Conference on Service-Oriented Computing and Applications – SOCA (2012) 1-4.

VALE, T. et al. A study on service identification methods for software product lines, Proceedings of the 16th International Software Product Line Conference (SPLC '12), Vol. 2 (2012) 156-163.

VEMULAPALLI, A., SUBRAMANIAN, N. Transforming Functional Requirements from UML into BPEL to Efficiently Develop SOA-Based Systems, Proceedings of the Confederated International Workshops and Posters on On the Move to Meaningful Internet Systems (2009) 337-349.

WANG, Z., Xu, X., Zhan, D. Normal Forms and Normalized Design Method for Business Service, Proceedings of the IEEE International Conference on e-Business Engineering, ICEBE '05 (2005) 79-86.

WEIGAND, H. et al. Value-Based Service Modeling and Design: Toward a Unified View of Services, CAiSE '09 Proceedings of the 21st International Conference on Advanced Information Systems Engineering (2009) 410-424.

WEIGAND, H. The pragmatics of event-driven business processes, Proceedings of the 7th International Conference on Semantic Systems, ACM (2011) 211-218.

WEN, B. et al. Process semantic-enabled customization for active service provisioning, Proceedings - 2010 13th IEEE International Conference on Computational Science and Engineering, CSE (2010) 375-381.

YONGCHAREON, S. et al. A framework for behavior-consistent specialization of artifact-centric business processes. Proceedings of the 10th international conference on Business Process Management (BPM'12), Alistair Barros, Avigdor Gal, and Ekkart Kindler (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 285-301, 2012.

YOUSEF, R. et al. BPAOntoSOA: A generic framework to derive software service oriented models from business process architectures, Applications of Digital Information and Web Technologies ICADIWT'09 (2009) 50-55.

YUN, Z. et al. A Service-Oriented Analysis and Design Approach Based On Data Flow Diagram, Computational Intelligence and Software Engineering, CiSE (2009) 1-5.

ZADEH, A.T. et al. A Systematic Input Selection for Service Identification in SMEs, Journal of Applied Sciences 12 (2012) 1232-1244.

ZHANG, Z., LIU, R., YANG, H. Service Identification and Packaging in Service Oriented Reengineering, In Proceedings of the 7th International Conference on Software Engineering and Knowledge Engineering (SEKE) (2005) 241-249.

ZHANG, Z., YANG, H. Incubating services in legacy systems for architectural migration, Software Engineering Conference, 11th Asia-Pacific (2004) 196-203.

ZHANG, Z. et al. A service composition approach based on sequence mining for migrating e-learning legacy system to SOA, International Journal of Automation and Computing 7 (4) (2010) 584-595.

ZIMMERMANN, O. et. al. Elements of service oriented analysis and design. An interdisciplinary modeling approach for SOA projects. IBM (2004).

Appendix

APPENDIX A – A SYSTEMATIC SURVEY OF SERVICE IDENTIFICATION METHODS

Abstract: One of the major challenges for the service- oriented architecture (SOA) adoption is the service identification phase that aims to determine which services are appropriate to be implemented. In the last decade, several service identification methods (SIMs) were proposed. However, the service identification phase still remains as a challenge to the organizations because of the lack of systematic methods and comprehensive approaches that support the examination of the business from multiple perspectives and also that consider service quality attributes. This work aims to provide an overview of existing SIMs by detailing which of the service's perspectives, stated as relevant by the industry, are addressed by the SIMs, and also by synthesizing the identification techniques used by them. We carried out a systematic survey over publications about SIMs from 2002 to June 2013, and 105 studies were selected. A detailed investigation of the proposed SIMs revealed that the identification techniques applied by them have a correlation on how they address many of the service's perspectives and the evaluated SIMs are actually supporting SOA adoption by handling many perspectives of OASIS' reference architecture for Service-Oriented Architectures (SOA). However, most of them does not explicitly address service quality attributes and few studies support the evaluation of both business and technical perspectives. Therefore, future research should follow the direction of hybrid methods with mechanisms to elicit business and service's quality attributes.

Keywords: Service-Oriented Architecture (SOA), Service identification method, systematic survey.

1. Introduction

Organizations are in a competitive environment. Today's dynamic markets, the pressure to improve quality and productivity and to maintain competitive advantages made the adaptability to new business requirements critical issue to the survival of organizations. The Information Technology, as an important tool for organizations, also followed this trend. Therefore, the construction of interoperable services, which could be organized in a flexible way to quickly meet business needs as described in the Service-Oriented Architecture (SOA), became a promising alternative to be considered.

The process of service-oriented modeling and architectural design consists of three general steps, namely identification, specification, and realization of services, components and workflows [Arsanjani 2004]. The identification step (the main subject of this study) aims to determine which services are appropriate to be implemented in a service-oriented architecture. Erl [Erl 2007] defines three possible strategies for service identification. One is the top-down strategy, which advocates the completion of an inventory analysis (definition of enterprise business model, technology architecture and service inventory blueprint) prior to the physical design and development of services. The second is the bottom-up strategy that is tactically focused and makes the fulfillment of immediate business requirements a priority and the prime objective of the SOA project. The last strategy is the meet in the middle, which is a combination of both. The top-down strategy is used to promote alignment with business goals or processes and the bottom-up to evaluate the existing assets such as information systems, services repositories, databases and legacy documentation.

The service identification phase is one of the most practical and a real challenge in designing and implementing a service-oriented architecture [Demirkan et al. 2008]. This challenge lays in the fact that besides predicting which services an enterprise will eventually need and defining which functions should be part of each service, service identification should also take into account different levels of service granularity in order to promote reuse and, at the same time, to provide enough flexibility to enable service composition and orchestration without significant performance loss. Moreover, service identification should also produce a catalogue of services that is meaningful to the business. In order to address these challenges, it is essential to have a methodology to support examination of the business from multiple perspectives.

Considering the importance and the complexity of the identification phase in a SOA process, many Service Identification Methods (SIMs) have been proposed in the recent literature. These SIMs offer different techniques to identify SOA services, such as process decomposition, model driven approach, value analysis, source code extraction, and ontology mapping [Cai et al. 2011]. The goal of a SIM is to deal with challenges in service identification phase in order to identify services that have correct functionality, granularity, reuse, and flexibility for service composition and orchestration.

In this context, some surveys [Boerner and Goeken 2009], [Birkmeier et al. 2009], [Kohlborn et al. 2009], [Gu and Lago 2010], [Cai et al. 2011], [Vale et al. 2012], [Zadeh et al. 2012] have been published with the aim of providing an overview of the existing SIMs. However, meet in the middle approaches are generally not addressed in these surveys. Moreover, none of them evaluates several perspectives pointed out as relevant to the industry, including standards such as reference architectures [OASIS]. Furthermore, up to date only one survey on SIMs [Gu and Lago 2010] was conducted by adopting the guidelines of a systematic literature review [Kitchenham 2004], which provides a methodological, fair analysis of a given subject in a comprehensive and non-biased way. Therefore, all the aforementioned points create a gap that we aim to fulfill with this survey.

In such a context, this survey intends to: (i) take into account the different service perspectives stated by the SOA reference architectures presented in [OASIS] and suggest which of them are relevant to the service identification phase; (ii) provide a comprehensive overview of existing SIMs by detailing the techniques used by them to identify candidate services, and; (iii) to shed light on further opportunities for improvements in this field.

The remainder of this paper is structured as follows. Section 2 describes related work. Section 3 defines the research method adopted for the survey. Section 4 presents the comparison criteria and reports the obtained results. Section 5 describes the threats to validity. Finally the conclusion is provided in Section 6.

2. Related work

By following the search procedure detailed in Section 3, seven surveys focused on SIMs were found, as summarized in Table 1.

Table 1 – Comparison among existent SIM surveys.

Survey	**Year of publication**	**Purpose**	**Data sources**	**Years covered**	**Analyzed papers**
[Boerner and Goeken 2009]	2009	Compare existing SIMs and discuss their shortcomings, in	Not mentioned	2005, 2007, 2008	5

		particular economic and governance aspects			
[Birkmeier et al. 2009]	2009	Analyze the current state of the art about service identification and highlight differences between the investigated approaches.	Not mentioned	1984, 2001-2008	15
[Kohlborn et al. 2009]	2009	Provide a general overview of the SIMs and make a comparison covering business and technical view-points	Not mentioned	2004-2008	30
[Gu and Lago 2010]	2010	Aid practitioners on selecting the most appropriate SIM by providing an overview of their basic elements	ACM IEEEXplore Web of Knowledge ScienceDirect SpringerLink	2004-2009	30
[Cai et al. 2011]	2011	Present a comprehensive understanding of service identification and a classification based on the common high-value activities	Not mentioned	2004-2011	41
[Vale et al. 2012]	2012	Compare SIMs to aid stakeholders to choose the most suitable method in Service-Oriented Product Line	Not mentioned	2004-2009	32

		Engineering context			
[Zadeh et al. 2012]	2012	Evaluate the inputs that could be selected in SIMs process and determine their level of applicability for small and medium enterprises	Not mentioned	2002-2010	48
This survey	2014	Provide a classification scheme based on a SOA reference architecture and an comprehensive overview of existing SIMs	ACM DBLP IEEEXplore Scopus	2002-June 2013	105

Boerner and Goeken [Boerner and Goeken 2009] proposed six groups of SIM characteristics, namely basic characteristics, business aspects, technical aspects, economic aspects, components of method engineering, and principles of design science research. Such classification was used to compare five SIMs.

In turn, [Birkmeier et al. 2009] define 13 SIM characteristics, classified into Foundations (conceptual design), Procedure (technique), Model and Supporting Measures. The authors in [Kohlborn et al. 2009] list eight SIM characteristics (type of services, strategies, lifecycle covered, degree of prescription of the methods, validity, adoption of existing notations/processes, adoption of consumers/providers perspectives and use of service classification) and compare 30 identification methods in the context of such characteristics.

Gu and Lago [Gu and Lago 2010] select 30 primary studies from a set of 237 examined studies and compare them regarding types of inputs, outputs and strategies used in service identification. Cai et al. [Cai et al. 2011] assess 41 studies and propose a classification of high-value activities shared by different identification methods. Zadeh et al. [Zadeh et al. 2012] propose a criterion to evaluate SIM inputs regarding their machine readability, the level of interaction details among process, stakeholders, and service choreography that they elicit, their level of abstraction, goals coverage and possibility to be decomposed. Finally, Vale et al [Vale et al. 2012] selected the most significant criteria in previous surveys and compared 30 SIMs by considering the service granularity and type, strategy, inputs, outputs, activities, research method, validation formalism, economic aspects, and the industry sector where the method is applied.

The aforementioned surveys provide a good overview of service identification methods. Basically, they differ in the adopted criteria for selection and comparison. Nevertheless, we can observe that there is a lack of systematic methods for service identification. Moreover, they propose that the method to be developed has to be configurable depending on the utilization constraints within the organizations (e.g., unavailability of an input or the need of applying the method to small domains) and some of them also suggest that economic aspects and non-functional requirements must be considered.

The main differences among these surveys and the survey presented in this paper regard three aspects: (i) the research method employed; (ii) the inclusion of a significant new research papers on SIMs, published between 2010 and 2013, and; (iii) the adopted criteria for comparison and evaluation of existent SIMs.

The first aspect concerns the fact that, as we have already mentioned, only one of the aforementioned SIM surveys [Gu and Lago 2010] has followed the guidelines of a systematic literature review (SLR) [Kitchenham et al. 2009], which provides a methodological, fair analysis of a given subject in a comprehensive and non-biased way. Therefore, our survey was undertaken with the guidelines of a systematic literature review suggested by Kitchenham et al. [Kitchenham et al. 2009].

The second aspect pertain the fact that our survey represents an update of the state of the art regarding SIMs. As also presented in Section 3.2, 36 studies were published from 2010 to June 2013 that represent an expressive amount of recent studies that were not considered in the aforementioned surveys. In addition, some studies (for instance, [Caetano et al. 2010]) are cited by more than 40 authors, thus indicating significant contributions published in the last years that were addressed in our survey.

The third aspect concerns the fact the present survey compares and evaluates SIMs according to the perspectives based on the OASIS' reference architecture for SOA [OASIS]. The OASIS' reference architecture for SOA is a well-known architecture for both academia and industry as it describes several characteristics of a reference SOA environment and assists SOA understanding and adoption. Many of these characteristics are related to service definition and their evaluation may reveal opportunities of improvements in service identification methods.

Furthermore, this survey also presents a greater number of findings when compared to the other surveys as it incorporates researches based on the meet in the middle strategy, which is less analyzed in previous surveys. This is an important aspect of the present survey since approaches based on such a strategy are more comprehensive than those based on the top-down or bottom-up strategies. Meet in the middle strategies evaluate both business and technical perspectives and are more aligned to the enterprise reality considering existing assets and quickly delivering recognizable benefits without neglecting the fact that services are designed for reuse and must comply with the business context.

3. Systematic Literature Review (SLR)

SLRs are means of evaluating and interpreting all available relevant research to particular research question(s), topic area or phenomenon of interest, thus aiming to present a fair evaluation of a research topic by using a rigorous methodology. Such a rigorous methodology can be viewed as the main point that differentiates a SLR from a simple, traditional literature review (as it was performed in the related surveys about SIMs in Section 2) as it seeks to avoid the maximum of bias throughout the process, thus providing scientific value for the obtained findings. SLRs have been recently viewed as an useful way for dealing with research evidences, thus making it possible to systematically identify, select, analyze, and aggregate them for providing knowledge about a given research topic. They have been commonly used for synthesizing existing work from the literature in a comprehensive and non-biased way and for identifying research challenges and opportunities in the state-of-the-art regarding the research subject.

As proposed by Kitchenham et al. [Kitchenham et al. 2004], a SLR is structured over a systematic process typically divided in three main basic steps (Figure 1):

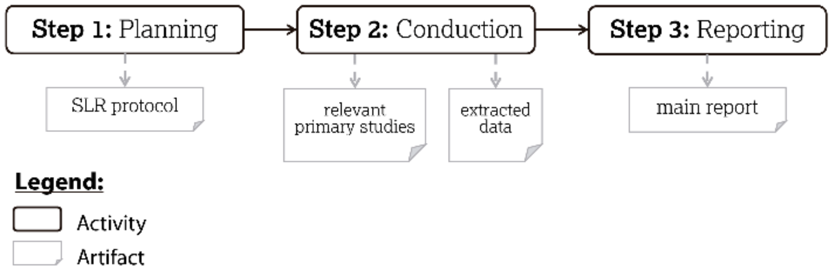

Figure1: Systematic literature review process.

- Step 1: Planning, which defines the research questions, search strategy, selection criteria, data extraction and synthesis methods to be used, and yields a protocol that will guide the conduction of the whole SLR process;
- Step 2: Conduction, in which the primary studies are identified, selected and evaluated according to the previously established protocol, and;
- Step 3: Reporting (or Analysis), which aggregates extracted information from the relevant primary studies considering the research questions and outlines conclusions from them.

Sections 3.1 and 3.2 detail the application of each the Planning and Conduction steps to the systematic survey presented in this paper, while Section 4 presents the results of the performed SLR (Reporting step).

3.1. Systematic review

In this phase, the goals and protocol of the SLR were defined. Such a protocol consists of a predetermined plan that describes the research questions, how the SLR process itself will be conducted (i.e., the search strategy to be adopted), and establishes the selection criteria and the data extraction and synthesis methods. The research questions must have a clear and well-defined focus as they drive the whole SLR, so that the search procedure must identify the studies that help to answer the previously-defined research questions, and the data extraction and analysis processes must produce data and knowledge to answer them.

3.1.1. Research questions

Despite of many published SIMs, there are gaps towards a comprehensive and systematic method for service identification. Identification methods can vary depending on the availability of inputs (business models, documentation, etc.) and the scope of identification (comprehensive and proactive identification or direct answer to a development project). Therefore, aiming at analyzing previous studies and summarize evidences about how existing SIMs work, the following research questions (RQs) were proposed based on these challenges:

RQ1: How do current SIMs address the different service's perspectives stated by the SOA reference architectures presented in [OASIS]?

RQ2: Which techniques are used by existing SIMs to identify candidate services?

3.1.2. Search Strategy

In order to establish the search strategy based on the defined research questions, three main terms were initially identified, namely service-oriented architecture, identification and services. In addition, in order to ensure a greater coverage in the results, we included the design and analysis terms, thus resulting in the following search string:

"service-oriented architecture" AND ("service identification" OR "identify services" OR "service design" OR "service analysis")

in which the main terms were connected by using the AND logical operator and the possible variations by using the OR logical operator.

In order to select the most proper databases for the search process, the following criteria discussed by [Dieste et al. 2009] were considered: (i) content update, i.e., if the publications are regularly updated; (ii) availability, i.e., if the full text of the primary study is available, and; (iii) quality of results, which is related to the accuracy of the results obtained by the search. As shown in Table 2, four electronic databases were selected based on these criteria and because they are the most commonly used databases in systematic reviews in the Software Engineering domain, as pointed out by [Kitchenham and Charters 2009] and [Dybå et al. 2007].

Table 2: Electronic databases selected as sources for the search process in the conducted SLR

ACM Digital Library	http://dl.acm.org
IEEEXplore	http://ieeexplore.ieee.org
Scopus	http://www.scopus.com

DBLP Computer Science Bibliography	http://www.dblp.org/search/

3.1.3. Inclusion and Exclusion Criteria

Some of the found studies might contain the keywords used in the search string, but are irrelevant to the research questions. Therefore, selection criteria are used to evaluate each primary study obtained by the search procedure according to the defined research questions, thus making it possible to include studies that are potentially relevant to answer the research questions and exclude studies that do not contribute to answer them.

The considered inclusion criteria (IC) were:

IC1: The study focuses on service identification.

IC2: The study should address different SIMs. If two different studies address improvements on the same SIM, only the most recent is considered.

The established exclusion criteria (EC) were:

EC1: The study is not publicly available in its complete form (full-access).

EC2: The study is not inserted into the SOA context.

EC3: The study is not written in English, which is the most common language in scientific papers.

3.1.4. Data Extraction and Synthesis Methods

The bibliographic details of each selected primary study were recorded by using Jabref [JABREF]. We also recorded the names of the authors, title of the study, source (journal, proceedings of conferences, etc.) and year of publication in a spreadsheet for quantitative analysis purposes. Furthermore, in order to extract data from these studies, data extraction spreadsheets related to each research question were built in order to synthesize the results and facilitate the drawing of conclusions.

3.2. Conduction

In this phase, the primary studies were searched, selected and evaluated according to the previously established protocol, thus resulting in a set of possibly relevant studies for the SLR. During the search process, the generic search string defined in the Planning phase has undergone minor changes in order to make it compatible with the specificities of each electronic database engine. After that, the automated search of primary studies was then performed over the selected electronic databases (see Table 2) by searching for all primary studies that matched the adapted search string and was limited to title, abstract and keywords fields. The performed search procedure covered publications from 2002 to June 2013; this start year was chosen by considering the oldest SIM reported by the surveys presented in Table 1.

In addition, a cross-reference check was also conducted by reviewing the related works section of the surveys presented in Table 1 to identify additional studies that are relevant but did not appear in the search results. These relevant studies found in such cross-reference check were then included in the result set of selected studies.

As summarized in Table 3, 871 studies were retrieved from the electronic databases.

Table 3: Number of retrieved studies after the search procedure over the electronic databases.

Search mechanism	Results
ACM Digital Library	291
DBLP Computer Science Bibliography	229
IEEEXplore	152
Scopus	199
Total	**871**

As depicted in Figure 2, from this initial set of 871 studies, 93 studies were removed as they were duplicate entries between the databases and 692 studies were removed based on the selection (inclusion/exclusion) criteria. All of the retrieved studies had theirs title, abstract and keywords read for the evaluation against the selection criteria. In case of doubt, the full text was analyzed. Many studies were excluded because they used terms such as “service identification” to refer to the process of discovering services already deployed that could be reused, while other studies used “service design” to address implementation issues for composing services. Additionally, 19 studies were included after the cross-references checking, thus resulting in a set of 105 primary studies (7 surveys and 98 methods) that were finally considered relevant to this SLR and then selected for data extraction. In this SLR, a given primary study is considered relevant if it does not meet any exclusion criterion and meet at least one inclusion criterion.

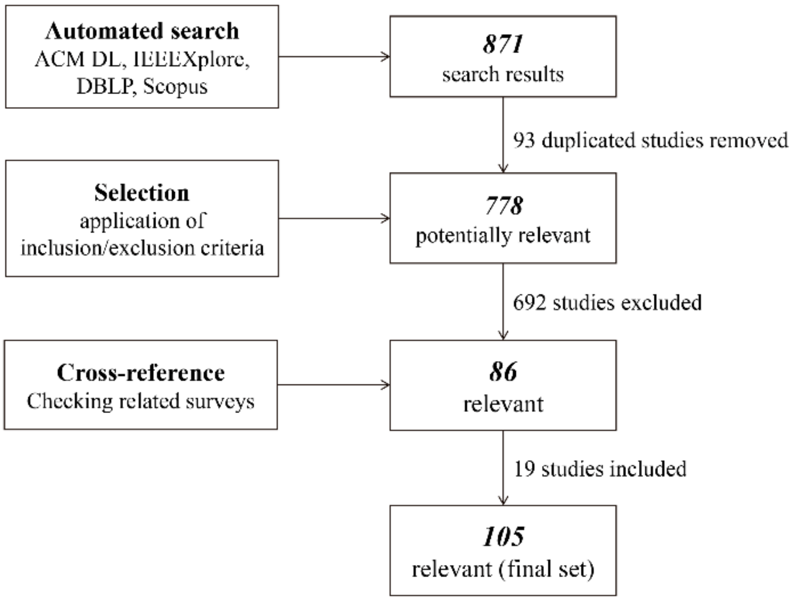

Figure 2: Selection of relevant studies.

Figure 3 shows the number of selected studies classified per year of publication. From 2010 to June 2013 (a range that is not covered by the unique systematic survey [Gu and Lago 2010] about SIMs published in 2010), 36 studies were published in these last four years. Therefore, this expressive number of recent studies (34.28% of the total amount) reveals one of the main contributions of this systematic survey, in terms of updating the state of the art about SIMs.

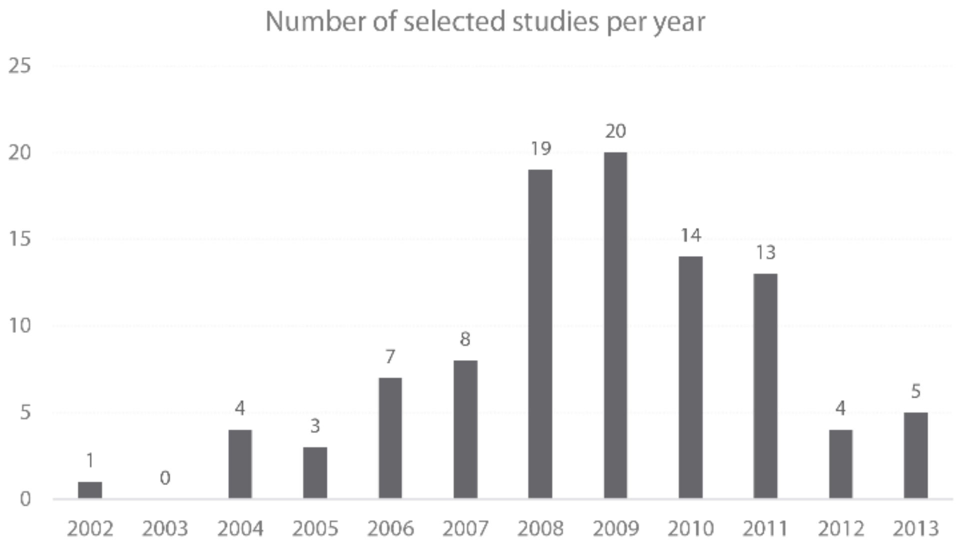

Figure 3: Number of selected studies per year.

4. Analysis of the Selected Studies

This section summarizes the results of the conducted systematic survey.

4.1. Classification scheme

In order to compare the approaches for identification of candidate services, we have developed a new classification scheme based on the OASIS' reference architecture framework for SOA [OASIS]. The proposed classification scheme intends to provide means to analyze how perspectives pointed out as relevant by industry are addressed by existing SIMs. The OASIS' reference architecture is an abstract realization of SOA that focuses on the required elements and their relationships to enable SOA-based systems to be used, realized and governed. It also provides a common language for understanding the important features of SOA, is independent from any technology, and has already been adopted by industry, thus being an important guide for issues that should be considered during the service identification phase.

The OASIS' reference architecture framework is structured by views. Views are representations of the whole system from the perspective of a related set of concerns. Each view is comprised of models, which represent an abstraction or representation of some architectural aspect. Models are mainly described by class diagrams in which each class is an element or a concept involved in a SOA ecosystem. A SOA ecosystem is a network of processes and machines that, along with a community of people, creates, uses, and governs specific services [OASIS].

According to the OASIS' framework, three views are used to describe SOA concerns: (i) *Participation in a SOA ecosystem*; (ii) *Realization of a SOA ecosystem*, and; (iii) *Ownership in a SOA ecosystem*. The *Participation in a SOA ecosystem* view focuses on the constraints and context in which people conduct business using a SOA-based system. The *Realization of a SOA ecosystem* view focuses on elements that are needed to support the discovery of and interaction with services. Finally, the *Ownership in a SOA ecosystem* view focuses on the governance and management of SOA-based systems. Since service identification can be crosscutting within the software development activities of elicitation, analysis and project of services, our classification schema only considers the first two views, which are directly related to such activities. Therefore, the third view is out of the scope of this work.

The first two views and their models were analyzed and the main elements and concepts of each model were identified. Figure 4 shows the classification scheme derived from the OASIS views and models. Table 4 explains each element or concept selected from the reference architecture and the classification perspective that was derived from them.

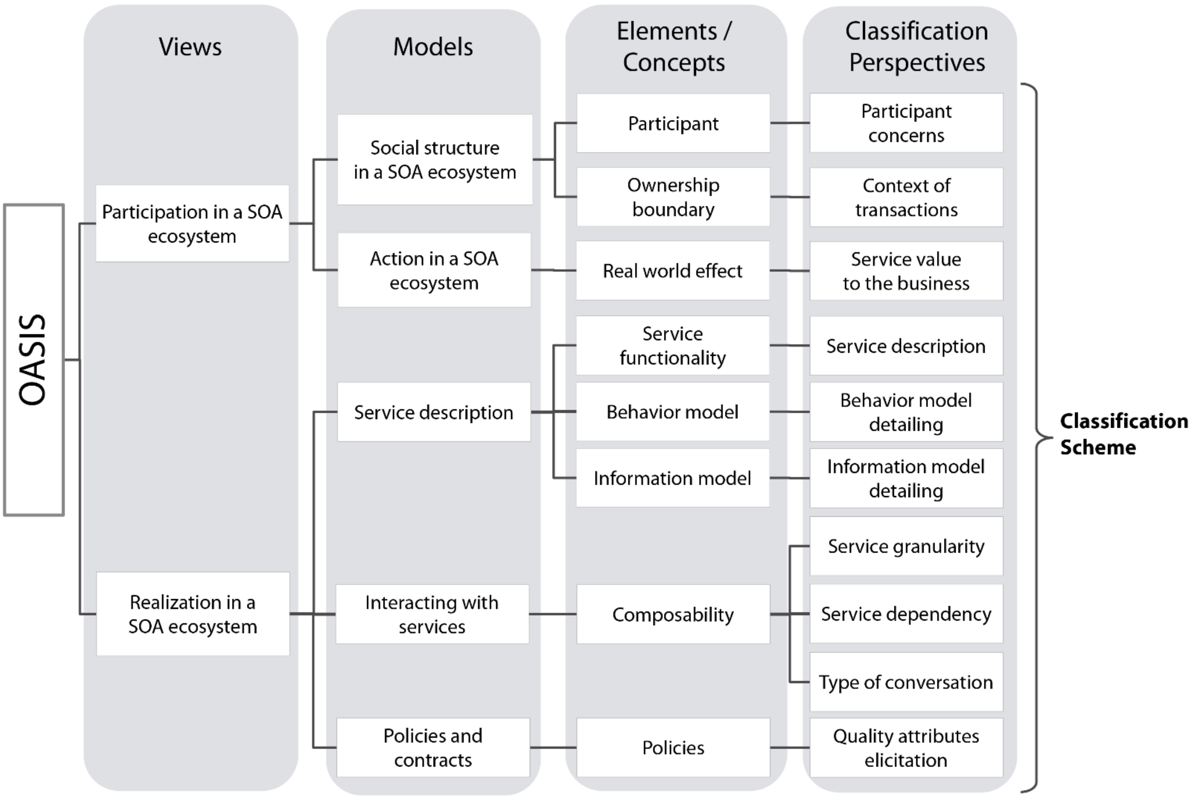

Figure 4: Classification scheme based on the OASIS' reference architecture for SOA.

For each element/concept, a classification perspective was derived according to service identification concerns in order to compose our classification scheme. For the *Participation in a SOA ecosystem* view, we have chosen the *Participant* and *Ownership boundary* elements, which are part of the *Social structures in a SOA ecosystem* model, and the *Real world effect* element, which is part of the *Actions in a SOA ecosystem* model. For the *Realization of a SOA ecosystem* view, we have chosen the *Service functionality*, *Behavior* and *Information model* elements, which are part of the *Service description model*. In addition, we have also chosen the *Composability* element, which is part of the *Interacting with services model*, and the *Policy* element, which is part of the *Policies and contracts model*.

Table 4 - Classification schema.

Model element or concept (from[OASIS])	**Derived classification perspective**
A *Participant* is an actor or stakeholder concerned with expressing needs and seeing those needs fulfilled. A *provider* role is played by a participant that offers a service and a *consumer* role is played by a participant that interacts with a service in order to fulfill a need.	*Participant Concerns* analyze whether the SIM addresses concerns regarding either the service provider or the consumer perspectives. The most common concerns are (i) the identification of business functions with high potential of reutilization, and (ii) implementation issues, such as either service internal details or guidelines to service realization.
The *Ownership Boundary* is the extent of ownership asserted by a stakeholder over a set of resources. The interaction between the provider and the consumer of a service crosses an ownership boundary when a service is owned by a provider that is different from the consumer.	The *Context of Transactions* Describes which interactions are analyzed by the SIMs (across or inside enterprise boundaries).
A *Real World Effect* describes the result of interacting with a service. The real world effect is the same regardless the consumer.	The *Service Value to the business* describes the effect of a service to the business. Such an effect can be: (i) *direct*, when it is related to the business strategy and then directly perceived by the client, or; (ii) *indirect*, when it is related to technical aspects and is commonly used to fulfill a need of a service that produces a direct value to the business.
The *Service Functionality* is a clear expression of service functions and the real world effects of invoking them. Functions represent business activities in some domain that produce the desired real world effects.	The *Service Description* analyzes which types of descriptions are elicited by the SIMs. Such descriptions can be: (i) *syntax-based*, thus hiding what happens inside of the service and exposes the necessary input and output values of service interfaces, and; (ii) *semantic-based*, which describes service based on semantic languages such as ontology annotation and context information based methods [Teka et al. 2012].
A *Behavior Model* is comprised of an action model and a process model. An *action model* characterizes services actions and a *process model* characterizes the temporal sequence of actions and their dependencies.	*Behavior Model Detailing* analyzes whether the SIM addresses concerns regarding the service internal or external behavior. *External behavior* focus on service inputs and outputs, while *internal behavior* focuses on internal actions, events and conditions.

An *Information Model* describes the syntax and semantics of the messages and data payloads, exception conditions and error handling in the event of faults.	*Information Model Detailing* describes whether the SIM elicits the syntax and semantics of the messages and data (detailed business perspective).
Composability is the ability of combining individual services that provide specific business functionality, so that their composition can provide more complex business solutions.	*Service Granularity* identifies which types of services are elicited by the SIMs. A service can be: (i) *atomic*, when it is visible to a consumer through a single interface and is described through a single service description that does not use or interact with other services, and; (ii) *composite*, when it is visible to a consumer through a single interface and is described through a single service description comprised by the aggregation or composition of one or more other services.
	Service Dependency describes whether the SIM elicits relations between services and/or resources required or provided by them.
	Type of Conversation identifies types of service collaboration techniques addressed by the SIMs. Collaboration is achieved by through: (i) *choreography*, which characterizes and composes business collaborations based on ordered message exchanges between peer entities in order to achieve a common business goal, or; (ii) *orchestration*, which composes hierarchical and self-contained service-oriented business processes that are executed and coordinated by a single agent.
Policies are constraints or the conditions of use of a service. Policies are potentially applied to many aspects of SOA, e.g., security, privacy, manageability, quality of service etc.	*Quality Attributes Elicitation* analyzes whether the SIMs elicits quality attributes that influence the design, quality, policies or execution contexts of the services.

It is important to mention that the *Realization of a SOA ecosystem* view has a fourth model named *Service visibility*. Achieving visibility is one of the key requirements for participants interacting with each other in the context of a SOA. This model analyzes visibility in terms of awareness, willingness, and reachability. Since these concepts are more related to processes aiming to maintain services descriptions available and to service deployment issues than to the identification phase, such a model is out of the scope of this work.

Although the aim of this survey is to analyze perspectives pointed out as relevant by industry, the classification schema proposed illustrated in Figure 4 is not disconnected of the criteria proposed by the surveys mentioned in Section 2. Several perspectives in Table 4 encompass the criteria used by the aforementioned surveys as follows:

- *Participant Concerns encompasses perspective Regard* to Stakeholders in [Kohlborn et al. 2009];
- *Context of Transactions* encompasses the *Types of categorization* in [Boerner and Goeken 2009];
- *Service Value to the business* encompasses *Consideration of Strategic Perspectives* in [Boerner and Goeken 2009] and *Inputs* in [Gu and Lago 2010], [Vale et al. 2012], [Zadeh et al. 2012];
- *Behavior Model Detailing* and *Information Model Detailing* encompass *Model Views* in [Birkmeier et al. 2009] and *Output Format* in [Gu and Lago 2010], [Vale et al. 2012];
- *Service Granularity* encompasses *Granularity* in [Boerner and Goeken 2009], [Vale et al. 2012], *Service Hierarchy* in [Birkmeier et al. 2009], [Kohlborn et al. 2009], and Service Types in [Gu and Lago 2010], [Vale et al. 2012];
- Service Dependency encompasses Supported Objects in [Boerner and Goeken 2009] and Dependencies in [Birkmeier et al. 2009];
- Type of Conversation encompasses Orchestration vs. Choreography in [Boerner and Goeken 2009];
- Quality Attributes Elicitation encompasses Legal Compliance, Internal Policies, and Service Level Agreements in [Boerner and Goeken 2009].

Some perspectives as *Economic perspectives* [Boerner and Goeken 2009], [Vale et al. 2012], *Method degree of detail* [Boerner and Goeken 2009], [Birkmeier et al. 2009], [Kohlborn et al. 2009], *Tool support* [Birkmeier et al. 2009], *SOA lifecycle coverture* [Boerner and Goeken 2009], [Kohlborn et al. 2009] and *Industry sector* [Boerner and Goeken 2009], [Vale et al. 2012] do not have correspondence to our classification criteria and then were not analyzed in this survey. Finally, perspectives *Technique* [Boerner and Goeken 2009], [Birkmeier et al. 2009], [Gu and Lago 2010], [Cai et al. 2011], [Vale et al. 2012] and the *Identification strategy* [Boerner and Goeken 2009], [Birkmeier et al. 2009], [Kohlborn et al. 2009], [Gu and Lago 2010], [Vale et al. 2012] employed in the service identification process cannot be directly correlated to any element of the reference architecture. The perspective Technique implements the identification strategy and describe the method used to identify service candidates. Considering the relevance of these two perspectives to the service identification phase and their recurrence in the existent SIM surveys, this study analyses the *Identification Strategy* within the scope of almost all perspectives described in Table 4 and includes a specific section (Section 4.2.2) to analyze the perspective *Technique*.

4.2. Results

The following sections details the achieved findings of the conducted search starting with the evaluation of perspectives derived from the reference architecture (Section 4.2.1) and then detailing the techniques (Section 4.2.2).

4.2.1. Evaluation of the classification perspectives derived from the reference architecture

After defining our classification schema presented in Table 4, the findings were evaluated accordingly to it. Such an evaluation intends to identify which of those classification perspectives are evaluated by existing SIMs, thus answering RQ1. A summary of how each perspective was addressed is presented in the following subsections.

4.2.1.1. Participant concerns.

The Participant Concerns perspective evaluates providers' or consumers' concerns addressed by the SIMs. The most common concerns are reutilization and implementation issues. Reutilization concern is addressed in SIMs by the identification of functions or tasks with high potential of reuse, i.e., functions required by several stakeholders inside or outside the organization.

Implementation concern is addressed in the SIMs by: (i) the provision of steps and guidelines to service realization and implementation [Abdelaziz et al. 2011], [Wang et al.

2005], [Erradi et al. 2006], [Arsanjani et al. 2008], [Jamshidi et al. 2009], [Adamopoulos et al. 2002]; (ii) extraction of legacy code for service packaging [Ricca and Marchetto 2009], [Zhang et al. 2010], [Li and Tahvildari 2006], [Canfora et al. 2008], [Li and Tahvildari, 2008], [Bao et al. 2010], [Chen et al. 2005], [Zhang and Yang 2004], [Aversano et al. 2008], [Baghdadi 2006], [Zhang et al. 2005], [Chung et al. 2009], [Komondoor et al. 2012], [Abdelkader et al. 2013], [Upadhyaya et al. 2012], [Fuhr et al. 2013]; and (iii) identification of points of variability. Variability refers to assumptions about how members of a family of products may differ from each other [Park et al. 2009]. The most common types of variability addressed are variability of activities (activities can be optional or alternative to accomplish an action) [DongSu et al. 2008], [Chang 2007], [Park et al. 2009], [Ponnalagu and Narendra 2008], [Medeiros et al. 2010], [Asadi et al. 2011], [Asadi et al. 2009], [Abu-Matar and Gomaa 2011], variability in the interfaces [Chang 2007], [Ponnalagu and Narendra 2008], [Kenzi et al. 2008], [Abu-Matar and Gomaa 2011] and finally product dynamic reconfiguration based on context information [Lee et al. 2010].

Table 5: Participant concerns.

	SIM		
Concern	*Top-down*	*Bottom-up*	*Meet in the middle*
Reutilization issues only	[Dinh and Nguyen-Ngoc 2010], [Aier 2006], [Alizadeh et al. 2011], [Andersson et al. 2009], [Bianchini et al. 2009], [Bianchini et al. 2013], [Birkmeier et al. 2013], [Brzostowski et al. 2010], [Caetano et al. 2007], [Dwivedi and Kulkarni 2008], [Flaxer and Nigam 2004], [Gacitua-Decar and Lero 2009], [Gordijn et al. 2006], [Guan et al. 2012], [Han et al. 2009], [Huayou et al. 2009], [Ilayperuma and Zdravkovic 2010], [Jamshidi et al. 2008], [Jin and Zhu 2008], [Kaabi et al. 2004], [Kazemi et al. 2011a], [Kazemi et al. 2011b], [Kim and Doh 2007], [Kim and Doh 2009], [Kim and Doh 2012], [Kim and Doh 2013],	[Alahmari et al. 2010], [Chen et al. 2009], [de Bruin et al. 2009], [Jain et al. 2004], [Kannan and Srivastava 2008], [Mani et al. 2008], [Nakamura et al. 2011], [Vemulapalli and Subramanian 2009]	[Chaari et al. 2007], [Cho et al. 2008], [Erradi et al. 2007], [Inaganti and Behara 2007], [Nguyen et al. 2009], [Patig and Wesenberg 2009], [Shiang et al. 2009], [Shirazi et al. 2009]

	[Kim and Suhh 2010], [Kim et al. 2008], [Klose et al. 2007], [Kohlmann and Alt 2007], [Lee et al. 2011], [Liu et al. 2011], [Lo and Yu 2007], [Meertens et al. 2010], [Menascé et al. 2008], [Mosser et al. 2011], [Ren and Wang 2010], [Ruiz et al. 2006], [Samavi et al. 2008], [Schmidt 2011], [Sewing et al. 2006], [Souza et al. 2009], [Souza et al. 2011], [Stewart and Chakraborty 2010], [Wang et al. 2005], [Weigand 2011], [Weigand et al. 2009], [Wen et al. 2010], [Yousef et al. 2009], [Yun et al. 2009]		
Implementation and reutilization issues	[Abdelaziz et al. 2011], [Abu-Matar and Gomaa 2011], [Asadi et al. 2009], [Asadi et al. 2011], [DongSu et al. 2008], [Jamshidi et al. 2009], [Kenzi et al. 2008], [Lee et al. 2010], [Medeiros et al. 2010], [Park et al. 2009], [Ponnalagu and Narendra 2008], [Wang et al. 2005],	[Abdelkader et al. 2013],[Adamopoulos et al. 2002], [Aversano et al. 2008], [Baghdadi 2006], [Bao et al. 2010], [Canfora et al. 2008], [Chen et al. 2005], [Chung et al. 2009], [Fuhr et al. 2013], [Komondoor et al. 2012], [Li and Tahvildari 2006], [Li and Tahvildari, 2008], [Ricca and Marchetto 2009], [Upadhyaya et al. 2012], [Zhang and Yang 2004], [Zhang et al. 2005], [Zhang et al. 2010]	[Arsanjani et al. 2008], [Chang 2007], [Erradi et al. 2006]

Table 5 shows that the majority of the top-down approaches aims only to identify reusable services with no implementation concerns. Top-down SIMs that have implementations concerns are those that address variability as aforementioned. On the other hand, bottom-up SIMs usually address implementation concerns since they deal with legacy code extraction and reorganization to define the service packaging.

It is interesting to mention that despite of the legacy systems' evaluation carried out by *meet in the middle* approaches, the majority of them only use it to detail the business domain [Nguyen et al. 2009], [Chaari et al. 2007], or to identify if any existing function corresponds to a requirement that a service candidate must fulfill [Shirazi et al. 2009], [Nguyen et al. 2009], [Patig and Wesenberg 2009], [Cho et al. 2008], [Shiang et al. 2009], [Inaganti and Behara 2007], [Fareghzadeh 2008], [Erradi et al. 2007] without addressing implementation concerns.

Regarding the techniques detailed in section 4.2.2, only *Product line*, *Source code analysis* and *Wrapping* tackle implementation concerns. The other techniques only tackle reutilization issues.

4.2.1.2. Context of transactions

The Context of Transactions perspective classifies SIMs taking into account issues related to the distribution of resources and interactions of people and systems inside or outside the enterprise. The explicitness of these boundaries is important to identify the implications of crossing them, especially to analyze their impact on aspects related to governance and security.

An example of how the context of transactions is addressed inside of enterprise boundaries is the approach presented in [Caetano et al. 2010]. The proposed SIM uses Role Ontology in order to decompose business process, identifying services goals, services interactions with applications or persons and also the resources (inputs and outputs) of a service.

The context of transactions outside enterprise boundaries is exemplified by the SIM proposed in [Dinh and Nguyen-Ngoc 2010] that uses the Service Responsibility Analysis technique. This technique identifies services by using as inputs information sharing relationships between organizations, thus eliciting their responsibilities and rules which govern the interchange of information.

Table 6: Context of transactions.

	SIM	
Context of transactions	Interactions are not detailed	Interactions are detailed
Inside of enterprise boundaries	[Abdelaziz et al. 2011], [Abdelkader et al. 2013], [Adamopoulos et al. 2002], [Aier 2006], [Alahmari et al. 2010], [Arsanjani et al. 2008], [Asadi et al. 2009], [Asadi et al. 2011], [Aversano et al. 2008], [Baghdadi 2006], [Bao et al. 2010], [Bianchini et al. 2013], [Birkmeier et al. 2013], [Brzostowski et al. 2010], [Canfora et al. 2008], [Chaari et al.	[Abu-Matar and Gomaa 2011], [Alizadeh et al. 2011], [Caetano et al. 2007], [Huayou et al. 2009], [Jin and Zhu 2008], [Kim and Suhh 2010], [Wen et al. 2010]

	2007], [Chang 2007], [Chen et al. 2005], [Chen et al. 2009], [Cho et al. 2008], [Chung et al. 2009], [de Bruin et al. 2009], [DongSu et al. 2008], [Dwivedi and Kulkarni 2008], [Erradi et al. 2006], [Erradi et al. 2007], [Fareghzadeh 2008], [Flaxer and Nigam 2004], [Fuhr et al. 2013], [Gacitua-Decar and Lero 2009], [Guan et al. 2012], [Inaganti and Behara 2007], [Jain et al. 2004], [Jamshidi et al. 2008], [Jamshidi et al. 2009], [Kannan and Srivastava 2008], [Kazemi et al. 2011a], [Kazemi et al. 2011b], [Kenzi et al. 2008], [Kim and Doh 2007], [Kim and Doh 2009], [Kim and Doh 2012], [Kim and Doh 2013], [Kim et al. 2008], [Kohlmann and Alt 2007], [Komondoor et al. 2012], [Lee et al. 2010], [Li and Tahvildari 2006], [Li and Tahvildari, 2008], [Liu et al. 2011], [Mani et al. 2008], [Medeiros et al. 2010], [Meertens et al. 2010], [Menascé et al. 2008], [Mosser et al. 2011], [Nakamura et al. 2011], [Nguyen et al. 2009], [Park et al. 2009], [Patig and Wesenberg 2009], [Ponnalagu and Narendra 2008], [Ren and Wang 2010], [Ricca and Marchetto 2009], [Ruiz et al. 2006], [Sewing et al. 2006], [Shiang et al. 2009], [Shirazi et al. 2009], [Souza et al. 2009], [Souza et al. 2011], [Stewart and Chakraborty 2010], [Upadhyaya et al. 2012], [Vemulapalli and Subramanian 2009], [Wang et al. 2005], [Weigand 2011], [Yousef et al. 2009], [Yun et al. 2009], [Zhang and Yang 2004], [Zhang et al. 2005], [Zhang et al. 2010]	
Outside of enterprise boundaries	[Han et al. 2009], [Schmidt 2011]	[Andersson et al. 2009], [Bianchini et al. 2009], [Dinh and Nguyen-Ngoc 2010], [Gordijn et al. 2006], [Ilayperuma and Zdravkovic 2010], [Kaabi et al. 2004], [Klose et al. 2007], [Lee et al. 2011], [Lo and Yu 2007], [Samavi et al. 2008], [Weigand et al. 2009].

Table 6 shows that the majority of the studies do not detail interactions of actors (people or systems) with a service. The studies that make such a detailing are top-down approaches that apply ontologies to describe services [Jin and Zhu 2008], [Huayou et al. 2009], [Caetano et al. 2007], [Alizadeh et al. 2011], [Kim and Suhh 2010], [Wen et al. 2010], [Abu-Matar and Gomaa 2011] or employ techniques as *Value Analysis* [Andersson et al. 2009], [Lee et al. 2011], [Samavi et al. 2008], [Kaabi et al. 2004], [Gordijn et al. 2006], [Ilayperuma and Zdravkovic 2010], [Lo and Yu 2007], [Klose et al. 2007], [Weigand et al. 2009], [Bianchini et al. 2009] and *Service Responsibility Analysis* [Dinh and Nguyen-Ngoc 2010]. Ontologies provide means to describe services capabilities, resources and actors, and *Value Analysis* and *Service Responsibility Analysis* techniques identify services from the explicitness of participant's interaction.

SIMs that deal with identification outside enterprise boundaries can be adapted to identify services inside enterprise boundaries. This can be accomplished by replacing external economic entities by internal entities, such as departments. Nevertheless the opposite is harder to be achieved because SIMs that deal with identification inside enterprise boundaries usually use as input intra-enterprise business process or legacy application to identify services.

4.2.1.3. Service value to the business

The SOA approach aims to align IT and business perspectives by building services that are business focused and can be reused and deployed across multiple software applications. The perspective *Service value to the business* evaluates whether the effect expected when a consumer interacts with a service is related to business strategy and, thus, is directly perceived by the client, or not. Services indirectly related to business are usually related to technical aspects or are fine-grained services that are used to compose other services that offer, in their turn, direct value to the business.

This classification perspective is directly related to the input used by the SIM, as shown in Table 7:

Table 7: Service value to the business.

Service value to the business	Input	SIMs
Direct	Business goal	[Andersson et al. 2009], [Kaabi et al. 2004], [Kazemi et al. 2011a], [Kim et al. 2008], [Meertens et al. 2010], [Shiang et al. 2009]
	Business model	[Alizadeh et al. 2011], [Bianchini et al. 2009], [Brzostowski et al. 2010], [Dinh and Nguyen-Ngoc 2010], [Flaxer and Nigam 2004], [Gordijn et al. 2006], [Han et al. 2009], [Ilayperuma and Zdravkovic 2010], [Jamshidi et al. 2008], [Jin and Zhu 2008], [Lee et al. 2011], [Lo and Yu 2007], [Samavi et al. 2008], [Stewart and Chakraborty 2010], [Weigand et al. 2009]
Indirect	Legacy system	[Abdelkader et al. 2013], [Alahmari et al. 2010], [Aversano et al. 2008], [Bao et al. 2010], [Canfora et al. 2008], [Chaari

		et al. 2007], [Chang 2007], [Chen et al. 2005], [Chen et al. 2009], [Cho et al. 2008], [Chung et al. 2009], [Erradi et al. 2006], [Erradi et al. 2007], [Fuhr et al. 2013], [Inaganti and Behara 2007], [Jain et al. 2004], [Komondoor et al. 2012], [Li and Tahvildari 2006], [Li and Tahvildari, 2008], [Nakamura et al. 2011], [Patig and Wesenberg 2009], [Ricca and Marchetto 2009], [Shiang et al. 2009], [Upadhyaya et al. 2012], [Zhang and Yang 2004], [Zhang et al. 2005]
	Legacy application documentation	[Arsanjani et al. 2008], [de Bruin et al. 2009], [Fareghzadeh 2008], [Kannan and Srivastava 2008], [Mani et al. 2008], [Nguyen et al. 2009], [Shirazi et al. 2009], [Vemulapalli and Subramanian 2009] , [Zhang et al. 2010]
Both	Business process	[Abdelaziz et al. 2011], [Aier 2006], [Arsanjani et al. 2008], [Asadi et al. 2009], [Bianchini et al. 2013], [Birkmeier et al. 2013], [Caetano et al. 2007], [Chaari et al. 2007], [Chang 2007], [Cho et al. 2008], [Dwivedi and Kulkarni 2008], [Erradi et al. 2006], [Erradi et al. 2007], [Fareghzadeh 2008], [Gacitua-Decar and Lero 2009], [Guan et al. 2012], [Inaganti and Behara 2007], [Jamshidi et al. 2009], [Kazemi et al. 2011b], [Kim and Doh 2009], [Kim and Doh 2012], [Kim and Suhh 2010], [Klose et al. 2007], [Kohlmann and Alt 2007], [Menascé et al. 2008], [Mosser et al. 2011], [Nguyen et al. 2009], [Park et al. 2009], [Patig and Wesenberg 2009], [Ponnalagu and Narendra 2008], [Ren and Wang 2010], [Sewing et al. 2006], [Shirazi et al. 2009], [Souza et al. 2009], [Souza et al. 2011], [Wang et al. 2005], [Weigand 2011], [Yousef et al. 2009]
	Features	[Abu-Matar and Gomaa 2011], [Asadi et al. 2011], [DongSu et al. 2008], [Lee et al. 2010], [Medeiros et al. 2010]
	Requirements	[Adamopoulos et al. 2002], [Huayou et al. 2009], [Kenzi et al. 2008], [Kim and Doh 2007], [Kim and Doh 2013], [Liu et al. 2011], [Ruiz et al. 2006], [Schmidt 2011], [Wen et al. 2010], [Yun et al. 2009],
	Database	[Baghdadi 2006]

The majority of the *Top-down* approaches identified services both directly and indirectly related to business. The other part of the *Top-down* approaches identified only services directly related to business. *Meet in the middle* approaches always identified services both directly and indirectly related to business and *Bottom-up* indirectly related. Although it is possible to find out some correlation of Service value perspective with the strategy, this perspective is directly related to the input used by the SIM.

Business goals are part of business strategy. A SIM can identify services to achieve these goals, thus having direct traceability to the business needs (e.g., [Shiang et al. 2009]).

The same principle applies to business models. As business models describe enterprise mission, business requirements and organizational architecture, they can be used as inputs to identify services that support information exchanges among organizations (e.g., [Lee et al. 2009]) or inside them (e.g., [Alizadeh et al. 2011]). On the other hand, legacy code (existing software assets of an enterprise) or documentation inputs derive services that usually have indirect relation to business. These services correspond to technical aspects or are fine-grained services used to compose other services (e.g., [Zhang ET al. 2010]).

Inputs as business processes, features, requirements or database assets can originate services with both direct and indirect effects. The service effect depends on the process decomposition level. High-level process originates services directly related to the business, while subsequent decompositions originate more fine-grained services that tend to be indirect related to the business as presented in [Bianchini et al. 2013]. When using features as inputs, the effect depends on their relevance to the business of the feature's product.

4.2.1.4. Service description

A good description is essential to enable service reutilization by matching user requirements against service capabilities. Service descriptions can be started within the identification phase or the specification phase. The advantage of describing services in the identification phase is to have a detailed perspective of service capabilities by identifying if the service really delivers the expected value, and also providing a better input to the specification phase. Descriptions can be semantic or syntax-based. Semantic descriptions describe a service based on semantically enriched formats such as ontology annotation [Huang et al. 2012] and context information based [Lee et al. 2010] methods. Syntax-based descriptions hide service implementation details and expose the externally observable service behavior as input and output values of service interfaces [Abelneh et al. 2012]. Semantic descriptions convey real world meaning to the services. This kind of description is important to give clear understanding of the effects of invoking a service and a consistent interpretation of data handled, especially when interaction occurs across ownership boundaries. It can also enable automatic service discovery and composition [Lee et al. 2001].

Table 8: Service description perspective.

Service description	SIM
Syntax-based	[Abdelaziz et al. 2011], [Abdelkader et al. 2013], [Adamopoulos et al. 2002],

	[Aier 2006], [Alahmari et al. 2010], [Andersson et al. 2009], [Arsanjani et al. 2008], [Asadi et al. 2009], [Asadi et al. 2011], [Aversano et al. 2008], [Baghdadi 2006], [Bao et al. 2010], [Bianchini et al. 2009], [Bianchini et al. 2013], [Birkmeier et al. 2013], [Canfora et al. 2008], [Chaari et al. 2007], [Chang 2007], [Chen et al. 2005], [Cho et al. 2008], [Chung et al. 2009], [de Bruin et al. 2009], [Dinh and Nguyen-Ngoc 2010], [DongSu et al. 2008], [Dwivedi and Kulkarni 2008], [Erradi et al. 2006], [Erradi et al. 2007], [Fareghzadeh 2008], [Flaxer and Nigam 2004], [Fuhr et al. 2013], [Gacitua-Decar and Lero 2009], [Gordijn et al. 2006], [Guan et al. 2012], [Han et al. 2009], [Ilayperuma and Zdravkovic 2010], [Inaganti and Behara 2007], [Jain et al. 2004], [Jamshidi et al. 2008], [Jamshidi et al. 2009], [Kaabi et al. 2004], [Kazemi et al. 2011a], [Kazemi et al. 2011b], [Kenzi et al. 2008], [Kim and Doh 2007], [Kim and Doh 2009], [Kim and Doh 2012], [Kim and Doh 2013], [Kim et al. 2008], [Klose et al. 2007], [Kohlmann and Alt 2007], [Komondoor et al. 2012], [Lee et al. 2011], [Li and Tahvildari 2006], [Li and Tahvildari, 2008], [Liu et al. 2011], [Lo and Yu 2007], [Mani et al. 2008], [Medeiros et al. 2010], [Menascé et al. 2008], [Nakamura et al. 2011], [Nguyen et al. 2009], [Park et al. 2009], [Patig and Wesenberg 2009], [Ponnalagu and Narendra 2008], [Ren and Wang 2010], [Ricca and Marchetto 2009], [Ruiz et al. 2006], [Samavi et al. 2008], [Schmidt 2011], [Sewing et al. 2006], [Shiang et al. 2009], [Shirazi et al. 2009], [Souza et al. 2009], [Souza et al. 2011], [Stewart and Chakraborty 2010], [Upadhyaya et al. 2012], [Vemulapalli and Subramanian 2009], [Wang et al. 2005], [Yun et al. 2009], [Zhang and Yang 2004], [Zhang et al. 2005], [Zhang et al. 2010]
Semantic-based	No standard language [Lee et al. 2010], OWL-S [Jin and Zhu 2008], [Huayou et al. 2009], BPAOnt and QoSOnto [Yousef et al. 2009], AoURN and ADORE [Mosser et al. 2011], Archmate and Mendix [Meertens et al. 2010], RoleOntology [Caetano et al. 2007], IDEAS [Alizadeh et al. 2011], GBPO and SBPO [Kim and Suhh 2010], SORSO [Wen et al. 2010], Ontoextract [Kannan and Srivastava 2008], WSMO [Brzostowski et al. 2010], Functionality Ontology and SCO [Chen et al. 2009], Unified Event Ontology [Weigand 2011], REA ontology [Weigand et al. 2009], SOAML [Abu-Matar and Gomaa 2011]

As shown in Table 8, only 16 studies present semantic descriptions of services. Among them, only two are bottom-up approaches [Kannan and Srivastava 2008], [Chen et al. 2009] and the other 14 are *top-down*. This result might reflect an influence of the traditional methodologies for process modeling and software engineering used for modeling and developing systems, which do not have a strategy to document semantic aspects of information. The studies that describe services semantically usually adopt ontology-based techniques to identify services because ontologies are able to provide semantic classes to organize relevant domain aspects to service description, such as participants, resources, operations, etc.

Only one study [Lee et al. 2010] explicitly identifies and describes context information. This is accomplished by defining attributes (as data types and validity conditions) of each context parameter identified and specifying each situation as a logical expression of contextual parameters to enable service dynamic reconfiguration.

4.2.1.5. Behavior model

A behavior model is part of the service description and is fundamental to understand and facilitate interaction with the service. A well-defined behavior model characterizes the knowledge of the (i) actions invoked against the service, (ii) events, and (iii) temporal relationships associated in a service interaction. It should also describe activities involved in a workflow that represents a work unit [OASIS]. Since the service external behavior is dependent of service internal actions, sequence and events, SIMs should elicit these perspectives to support the behavior model description.

A behavior model description can be done after the service implementation, i.e., outside SIMs scope, but this description helps to elicit and understand the service scope and to enable assessment of service responsibilities (cohesion) and dependency from other services. Existing SIMs deal with the behavior model in three ways:

- External behavior only: The main purpose is to identify the external behavior of a service, but not its internal operation. The focus is on inputs and outputs or on the service purpose (real world effect resulting from the service execution) (as in [Andersson et al. 2009]).

- Action description: SIMs identify and describe internal service actions, but do not detail their external behavior (as in [Perin-Souza et al. 2011]).

- Actions and behavior description: SIMs identify and describe service internal actions, and their behavior. Internal actions behavior can be described as events, pre and post conditions and actions sequence (as in [Weigand 2011]).

Table 9: Behavior model description.

Behavior model handling	Behavior description	SIM

External behavior only	Not applicable	[Andersson et al. 2009], [Flaxer and Nigam 2004], [Gordijn et al. 2006], [Han et al. 2009], [Ilayperuma and Zdravkovic 2010], [Lee et al. 2011], [Lo and Yu 2007], [Samavi et al. 2008], [Schmidt 2011], [Stewart and Chakraborty 2010], [Weigand et al. 2009]
Actions description	Not applicable	[Aier 2006], [Asadi et al. 2011], [Chaari et al. 2007], [de Bruin et al. 2009], [DongSu et al. 2008], [Dwivedi and Kulkarni 2008], [Gacitua-Decar and Lero 2009], [Kazemi et al. 2011a], [Park et al. 2009], [Ponnalagu and Narendra 2008], [Souza et al. 2011], [Upadhyaya et al. 2012], [Yousef et al. 2009], [Yun et al. 2009], [Wen et al. 2010]
Actions and behavior description	Events, pre and post conditions	[Alizadeh et al. 2011], [Brzostowski et al. 2010], [Dinh and Nguyen-Ngoc 2010], [Jin and Zhu 2008], [Kannan and Srivastava 2008], [Kim and Suhh 2010], [Meertens et al. 2010], [Nguyen et al. 2009], [Weigand 2011]
	Sequence of actions	[Abdelaziz et al. 2011], [Abdelkader et al. 2013], [Abu-Matar and Gomaa 2011], [Adamopoulos et al. 2002], [Alahmari et al. 2010], [Arsanjani et al. 2008], [Asadi et al. 2009], [Aversano et al. 2008], [Baghdadi 2006], [Bao et al. 2010], [Bianchini et al. 2009], [Bianchini et al. 2013], [Birkmeier et al. 2013], [Caetano et al. 2007], [Canfora et al. 2008], [Chang 2007], [Chen et al. 2005], [Chen et al. 2009], [Cho et al. 2008], [Chung et al. 2009], [Erradi et al. 2006], [Erradi et al. 2007], [Fareghzadeh 2008], [Fuhr et al. 2013], [Guan et al. 2012], [Huayou et al. 2009], [Inaganti and Behara 2007], [Jain et al. 2004], [Jamshidi et al. 2008], [Kaabi et al. 2004], [Kazemi et al. 2011b], [Kenzi et al. 2008], [Kim and Doh 2007], [Kim and Doh 2009], [Kim and Doh 2012], [Kim and Doh 2013], [Kim et al. 2008], [Klose et al. 2007], [Kohlmann and Alt 2007], [Komondoor et al. 2012], [Lee et al. 2010], [Li and Tahvildari 2006], [Li and Tahvildari, 2008], [Liu et al. 2011], [Mani et al. 2008], [Medeiros et al. 2010], [Menascé et al. 2008], [Mosser et al. 2011], [Nakamura et al. 2011], [Patig and Wesenberg 2009], [Ren and Wang 2010], [Ricca and Marchetto 2009], [Ruiz et al. 2006], [Sewing et al. 2006], [Shiang et al. 2009], [Shirazi et al. 2009], [Souza et al. 2009], [Vemulapalli and Subramanian 2009], [Wang et al. 2005], [Zhang and Yang 2004], [Zhang et al. 2005], [Zhang et al. 2010]

As presented in Table 9, all SIMs that elicited only external behavior were top-down approaches. Among the SIMs that elicit actions descriptions, [de Bruin et al. 2009] is *bottom-up*, [Chaari et al. 2007] is *meet in the middle*, and the other SIMs are *top-down*. Regarding events' and conditions' elicitation, most of SIMs are *top-down* approaches, excepting [Nguyen et al. 2009] that is *meet in the middle* and [Kannan and Srivastava 2008] that is *bottom-up*. Except those aforementioned, all *bottom-up* and *meet in the middle* approaches elicit the sequence of actions for a service.

Some of the techniques detailed in section 4.2.2 have correlation with behavior model description. Among the studies that only detail external behavior, the majority uses *value analysis* technique [Andersson et al. 2009], [Lee et al. 2011], [Samavi et al. 2008], [Gordijn et al. 2006], [Ilayperuma and Zdravkovic 2010], [Lo and Yu 2007], [Weigand et al. 2009], [Han et al. 2009], [Flaxer and Nigam 2004]. Within the studies that provide internal actions and internal behavior description, techniques as *model-driven* ones, *ontology mapping* and *service responsibility analysis* tend to favor description of aspects as events, pre and post wrapping tend to focus in the description of internal actions sequence. In such techniques, the sequence of actions is identified by process activities sequence [Shirazi et al. 2009], [Menascé et al. 2008], [Souza et al. 2009], [Kazemi et al. 2011b], [Abdelaziz et al. 2011], [Kim and Doh 2009], [Klose et al. 2007], [Caetano et al. 2007], [Wang et al. 2005], [Patig and Wesenberg 2009], [Ren and Wang 2010], [Cho et al. 2008], [Shiang et al. 2009], [Kim et al. 2008], [Inaganti and Behara 2007], [Erradi et al. 2006], [Arsanjani et al. 2008], [Jamshidi et al. 2008], [Birkmeier et al. 2013], [Guan et al. 2012], [Bianchini et al. 2013], [Kim and Doh 2012], [Kohlmann and Alt 2007], [Fareghzadeh 2008], [Bianchini et al. 2009], [Erradi et al. 2007], [Sewing et al. 2006] in *decomposition* approaches, by process diagrams [Chang 2007], [Ruiz et al. 2006] or use case description [Huayou et al. 2009], [Kenzi et al. 2008], [Kim and Doh 2007], [Liu et al. 2011], [Kim and Doh 2013], [Adamopoulos et al. 2002] in *requirement analysis* and by implementation sequence in *source code analysis* [Alahmari et al. 2010], [Li and Tahvildari 2006], [Jain et al. 2004], [Li and Tahvildari, 2008], [Bao et al. 2010], [Chen et al. 2005], [Zhang and Yang 2004], [Aversano et al. 2008], [Baghdadi 2006], [Chen et al. 2009], [Chung et al. 2009], [Nakamura et al. 2011], [Komondoor et al. 2012], [Abdelkader et al. 2013], [Fuhr et al. 2013], [Vemulapalli and Subramanian 2009] as well as in *wrapping* [Ricca and Marchetto 2009], [Canfora et al. 2008], [Zhang et al. 2005], [Upadhyaya et al. 2012].

4.2.1.6. Information model:

As the behavior model, the information model is part of the service description. A well-defined service information model describes the syntax and semantics of the messages and data payloads, exception conditions and error handling in the event of faults [OASIS], thus enabling meaningful exchange of information by matching the model semantics with the semantics of the service consumers.

SIMs deal with the information model in a variety of ways as explained below:

- Not detailed: Information handled by the service is not identified or it is identified but is not detailed. The focus is on the service functionalities or in identifying the service purpose and not its internal operation (as in [Andersson et al. 2009]).

- Information structure description: Information model is detailed with focus on the description of the information structure (classes and attributes) (as in [Shirazi et al. 2009]).

- Messages and parameters: Information model is detailed with focus on identification and description of service messages and parameters structure (as in [Canfora et al. 2008]).

- Semantic description: Information model is detailed with focus on semantic description of the handled information or service messages (as in [Kim and Suhh 2010]).

Table 10: Information model description.

Information model handling	Description
Not detailed	[Abu-Matar and Gomaa 2011], [Aier 2006], [Alahmari et al. 2010], [Andersson et al. 2009], [Asadi et al. 2009], [Asadi et al. 2011], [Bianchini et al. 2013], [Birkmeier et al. 2013], [Chang 2007], [DongSu et al. 2008], [Dwivedi and Kulkarni 2008], [Flaxer and Nigam 2004], [Gacitua-Decar and Lero 2009], [Gordijn et al. 2006], [Guan et al. 2012], [Han et al. 2009], [Ilayperuma and Zdravkovic 2010], [Inaganti and Behara 2007], [Kazemi et al. 2011a], [Kazemi et al. 2011b], [Kim and Doh 2009], [Kim and Doh 2012], [Kim and Doh 2013], [Kim et al. 2008], [Kohlmann and Alt 2007], [Lee et al. 2011], [Liu et al. 2011], [Lo and Yu 2007], [Mani et al. 2008], [Menascé et al. 2008], [Nakamura et al. 2011], [Park et al. 2009], [Patig and Wesenberg 2009], [Ponnalagu and Narendra 2008], [Samavi et al. 2008], [Schmidt 2011], [Sewing et al. 2006], [Shiang et al. 2009], [Souza et al. 2011], [Stewart and Chakraborty 2010], [Vemulapalli and Subramanian 2009], [Weigand et al. 2009], [Yousef et al. 2009], [Yun et al. 2009], [Zhang et al. 2010]
Information structure description	[Abdelaziz et al. 2011], [Abdelkader et al. 2013], [Adamopoulos et al. 2002], [Aversano et al. 2008], [Baghdadi 2006], [Bao et al. 2010], [Bianchini et al. 2009], [Canfora et al. 2008], [Chaari et al. 2007], [Chen et al. 2005], [Cho et al. 2008], [Chung et al. 2009], [de Bruin et al. 2009], [Dinh and

	Nguyen-Ngoc 2010], [Erradi et al. 2006], [Fareghzadeh 2008], [Huayou et al. 2009], [Jain et al. 2004], [Jamshidi et al. 2008], [Jin and Zhu 2008], [Kannan and Srivastava 2008], [Kenzi et al. 2008], [Kim and Doh 2007], [Kim and Suhh 2010], [Komondoor et al. 2012], [Li and Tahvildari 2006], [Li and Tahvildari, 2008], [Meertens et al. 2010], [Mosser et al. 2011], [Nguyen et al. 2009], [Ren and Wang 2010], [Ruiz et al. 2006], [Shirazi et al. 2009], [Souza et al. 2009], [Wen et al. 2010], [Zhang and Yang 2004], [Zhang et al. 2005]
Messages and parameters	[Alizadeh et al. 2011], [Arsanjani et al. 2008], [Aversano et al. 2008], [Canfora et al. 2008], [Chung et al. 2009], [de Bruin et al. 2009], [Fuhr et al. 2013], [Jamshidi et al. 2008], [Jamshidi et al. 2009], [Kaabi et al. 2004], [Kenzi et al. 2008], [Kim and Doh 2007], [Kim and Suhh 2010], [Klose et al. 2007], [Komondoor et al. 2012], [Lee et al. 2010], [Medeiros et al. 2010], [Mosser et al. 2011], [Ricca and Marchetto 2009], [Ruiz et al. 2006], [Upadhyaya et al. 2012], [Wang et al. 2005], [Wen et al. 2010], [Zhang and Yang 2004], [Zhang et al. 2005]
Semantic description	[Alizadeh et al. 2011], [Brzostowski et al. 2010], [Caetano et al. 2007], [Chen et al. 2009], [Jin and Zhu 2008], [Kim and Suhh 2010], [Weigand 2011], [Wen et al. 2010]

The majority of the studies that do not detail information model is top-down approaches (Table 10). Bottom-up approaches detail information structure or messages and parameters. Finally, meet in the middle approaches do not detail neither information model nor information structure.

In regards to the techniques, we found some correlation with the information model description perspective. The techniques that traditionally do not focus on describing an information model are: *Pattern matching* which focus on identifying patterns in processes [Gacitua-Decar and Lero 2009], [Aier 2006] or in legacy code [Zhang et al. 2010], *Product line* which focus on functionalities performed by services [DongSu et al. 2008], [Park et al. 2009], [Ponnalagu and Narendra 2008], [Asadi et al. 2011], [Asadi et al. 2009], [Abu-Matar and Gomaa 2011] and *Value analysis* [Andersson et al. 2009], [Lee et al. 2011], [Samavi et al. 2008], [Gordijn et al. 2006], [Ilayperuma and Zdravkovic 2010], [Lo and Yu 2007], [Weigand et al. 2009], [Han et al. 2009], [Flaxer and Nigam 2004] that focus on identifying what a service should be and not on its internal description.

Another correlation was found with the SIMs that detail the behavior model. Those that used syntax- based service descriptions (Table 8) tend to focus on detailing information or message structures and the ones that used semantic-based tend to focus on semantic description of information and messages or on message structure.

4.2.1.7. Service granularity

Services can be atomic (visible to a consumer via a single interface and described via a single service description that does not use or interact with other services) or composite (visible to a consumer via a single interface and described via a single service description comprised by the aggregation or composition of one or more other services) [OASIS]. Service composition can be either performed by composing atomic or composite services. When composing services, the business logic is implemented by several services. This allows the definition of increasingly complex solutions by progressively aggregating components at higher levels of abstraction.

The study in [Alahmari et al. 2010] is an example of how a SIM can address the identification of both atomic and composite services. Such a SIM defines a service layers model, which is a natural composition hierarchy. The definition of which functions are part of a service is done accordingly to the layer responsibility. Thus, the organization in layers contributes to the identification of services with "right" granularity and cohesion. Table 11 presents an overview of how SIMs deal with the granularity perspective.

Table 11: Service granularity perspective.

Granularity	SIMs
Both atomic and composite	[Abu-Matar and Gomaa 2011], [Alahmari et al. 2010], [Arsanjani et al. 2008], [Asadi et al. 2009], [Asadi et al. 2011], [Baghdadi 2006], [Bianchini et al. 2013], [Birkmeier et al. 2013], [Brzostowski et al. 2010], [Caetano et al. 2007], [Canfora et al. 2008], [Chang 2007], [Cho et al. 2008], [de Bruin et al. 2009], [Dinh and Nguyen-Ngoc 2010], [Dwivedi and Kulkarni 2008], [Erradi et al. 2006], [Erradi et al. 2007], [Fareghzadeh 2008], [Huayou et al. 2009], [Inaganti and Behara 2007], [Jamshidi et al. 2009], [Kaabi et al. 2004], [Kannan and Srivastava 2008], [Kazemi et al. 2011b], [Kim and Doh 2013], [Kim and Suhh 2010], [Kim et al. 2008], [Klose et al. 2007], [Kohlmann and Alt 2007], [Komondoor et al. 2012], [Lee et al. 2010], [Lee et al. 2011], [Li and Tahvildari 2006], [Li and Tahvildari, 2008], [Liu et al. 2011], [Lo and Yu 2007], [Mani et al. 2008], [Medeiros et al. 2010], [Meertens et al. 2010], [Mosser et al. 2011], [Nakamura et al. 2011], [Park et al. 2009], [Ren and Wang 2010], [Ricca and Marchetto 2009], [Shiang et al. 2009], [Shirazi et al. 2009], [Souza et al. 2009], [Wang et al. 2005], [Weigand 2011], [Weigand et al. 2009], [Wen et al. 2010], [Yun et al. 2009], [Zhang et al. 2005]

Atomic	[Abdelaziz et al. 2011], [Abdelkader et al. 2013], [Adamopoulos et al. 2002], [Aier 2006], [Alizadeh et al. 2011], [Andersson et al. 2009], [Aversano et al. 2008], [Bao et al. 2010], [Bianchini et al. 2009], [Chaari et al. 2007], [Chen et al. 2005], [Chen et al. 2009], [Chung et al. 2009], [DongSu et al. 2008], [Flaxer and Nigam 2004], [Fuhr et al. 2013], [Gacitua-Decar and Lero 2009], [Gordijn et al. 2006], [Guan et al. 2012], [Han et al. 2009], [Ilayperuma and Zdravkovic 2010], [Jain et al. 2004], [Jamshidi et al. 2008], [Jin and Zhu 2008], [Kazemi et al. 2011a], [Kenzi et al. 2008], [Kim and Doh 2007], [Kim and Doh 2009], [Kim and Doh 2012], [Nguyen et al. 2009], [Patig and Wesenberg 2009], [Ruiz et al. 2006], [Samavi et al. 2008], [Schmidt 2011], [Sewing et al. 2006], [Souza et al. 2011], [Stewart and Chakraborty 2010], [Upadhyaya et al. 2012], [Vemulapalli and Subramanian 2009], [Yousef et al. 2009], [Zhang and Yang 2004]
Composite	[Menascé et al. 2008], [Ponnalagu and Narendra 2008], [Zhang et al. 2010]

It was not possible to identify any correlation between the techniques or approaches with the identification of atomic or composite services. All techniques were able to identify both types of services. The decision of either grouping functions in a coarse grained service or creating only fine grained services or creating fine grained and composing them seems to be related only to the scope chosen by each SIM.

4.2.1.8. Service dependency

This perspective aims to describe whether SIMs identify relations between services or resources required by them. Resources and relationship elicitation is important to give a broader knowledge of service utilization and operation, thus making service design and implementation easier. SIMs are classified in four categories according to the dependencies they elicit:

- None: SIMs do not identify any relation between services or resources required by them (as in [Andersson et al. 2009]).
- Resources: SIMs identify resources required by the candidate service as informational entities, databases that store them and legacy code or classes (as in [Zhang and Yang 2004]).

- Relationships: SIMs identify any kind of relations between services as compositions or collaborations to achieve a purpose (as in [Menascé et al. 2008]).

- Both: SIMs identify both resources required and relationships between services (as in [Chang 2007]).

Table 12: Service dependencies.

Dependency	SIM
None	[Aier 2006], [Andersson et al. 2009], [Bianchini et al. 2009], [Flaxer and Nigam 2004], [Gacitua-Decar and Lero 2009], [Gordijn et al. 2006], [Guan et al. 2012], [Han et al. 2009], [Ilayperuma and Zdravkovic 2010], [Kim and Doh 2009], [Kim and Doh 2012], [Kim and Doh 2013], [Lo and Yu 2007], [Samavi et al. 2008], [Sewing et al. 2006], [Stewart and Chakraborty 2010], [Vemulapalli and Subramanian 2009]
Resource	[Abdelaziz et al. 2011], [Abdelkader et al. 2013], [Adamopoulos et al. 2002], [Alizadeh et al. 2011], [Aversano et al. 2008], [Baghdadi 2006], [Bao et al. 2010], [Chen et al. 2005], [Chen et al. 2009], [Chung et al. 2009], [Jain et al. 2004], [Kazemi et al. 2011a], [Kenzi et al. 2008], [Patig and Wesenberg 2009], [Ruiz et al. 2006], [Upadhyaya et al. 2012], [Zhang and Yang 2004]
Relationship	[Abu-Matar and Gomaa 2011], [Asadi et al. 2009], [Asadi et al. 2011], [Birkmeier et al. 2013], [de Bruin et al. 2009], [Dinh and Nguyen-Ngoc 2010], [DongSu et al. 2008], [Dwivedi and Kulkarni 2008], [Huayou et al. 2009], [Jamshidi et al. 2009], [Kaabi et al. 2004], [Kim and Doh 2007], [Kim et al. 2008], [Kohlmann and Alt 2007], [Lee et al. 2010], [Lee et al. 2011], [Liu et al. 2011], [Meertens et al. 2010], [Menascé et al. 2008], [Mosser et al. 2011], [Park et al. 2009], [Ponnalagu and Narendra 2008], [Ren and Wang 2010], [Schmidt 2011], [Souza et al. 2009], [Souza et al. 2011], [Wang et al. 2005], [Weigand 2011], [Wen et al. 2010], [Yun et al. 2009], [Zhang et al. 2010]
Both	[Alahmari et al. 2010], [Arsanjani et al. 2008], [Bianchini et al. 2013], [Brzostowski et al. 2010], [Caetano et al. 2007], [Canfora et al. 2008], [Chaari et al. 2007], [Chang 2007], [Cho et al. 2008], [Erradi et al. 2006], [Erradi et al. 2007], [Fareghzadeh 2008], [Fuhr et al. 2013], [Inaganti and Behara 2007], [Jamshidi et al. 2008], [Jin and Zhu 2008], [Kannan and Srivastava 2008], [Kazemi et al. 2011b], [Kim and Suhh 2010], [Klose et al. 2007], [Komondoor et al. 2012], [Li and Tahvildari 2006], [Li and Tahvildari, 2008], [Mani et al. 2008], [Medeiros et al. 2010], [Nakamura et al. 2011], [Nguyen et al. 2009], [Ricca and Marchetto 2009], [Shiang et al. 2009], [Shirazi et al. 2009], [Weigand et al. 2009], [Yousef et al. 2009], [Zhang et al. 2005]

All SIMs that did not identify any relationships are top-down approaches. Among the SIMs that used a *bottom-up* approach, almost all (except 2) identified either resources required by candidate services or resources and relationships. The majority of the *top-down* approaches identified service relationships and meet in the middle identified both resources required and relationships between services.

Table 12 shows that the majority of the SIMs elicits relationships between services. This happens because the majority of the techniques used by the SIMs elicits relationships (dependencies) within the several inputs, and these dependencies can derive services relationships. Process-oriented approaches as *decomposition* [Shirazi et al. 2009], [Menascé et al. 2008], [Souza et al. 2009], [Kazemi et al. 2011b], [Klose et al. 2007], [Caetano et al. 2007], [Wang et al. 2005], [Ren and Wang 2010], [Cho et al. 2008], [Shiang et al. 2009], [Kim et al. 2008], [Inaganti and Behara 2007], [Erradi et al. 2006], [Arsanjani et al. 2008], [Jamshidi et al. 2008], [Chaari et al. 2007], [Birkmeier et al. 2013], [Guan et al. 2012], [Bianchini et al. 2013], [Kim and Doh 2012], [Kohlmann and Alt 2007], [Fareghzadeh 2008], [Bianchini et al. 2009], [Erradi et al. 2007], [Sewing et al. 2006] and *model driven* [Dwivedi and Kulkarni 2008], [Nguyen et al. 2009], [Meertens et al. 2010], [Weigand 2011], [Jamshidi et al. 2009] usually identify control flow between activities. Those activities or group of activities are performed by services compositions and collaborations. Relationships between activities also reflect relationships between services that implement them. *Product line* identifies services relationships from similarities and relationships between functions [Lee et al. 2010], [DongSu et al. 2008], [Park et al. 2009], [Ponnalagu and Narendra 2008], [Medeiros et al. 2010], [Asadi et al. 2011], [Asadi et al. 2009], [Abu-Matar and Gomaa 2011]. Functions can be achieved by services collaborations and compositions. *Requirements analysis* [Yun et al. 2009], [Huayou et al. 2009], [Schmidt 2011], [Chang 2007], [Wen et al. 2010], [Kim and Doh 2007], [Liu et al. 2011], [Kim and Doh 2013], [Adamopoulos et al. 2002] and *SODA* (Service-Oriented design aspect) [Mosser et al. 2011], [Souza et al. 2011] identifies services collaborations in order to fulfill a requirement or an aspect, and service responsibility identifies dependencies between task and data services [Dinh and Nguyen-Ngoc 2010].

Techniques such as *asset identification* [Shirazi et al. 2009], [Nguyen et al. 2009], [de Bruin et al. 2009], [Arsanjani et al. 2008], [Mani et al. 2008], [Fareghzadeh 2008], [Vemulapalli and Subramanian 2009] and *ontology mapping* [Jin and Zhu 2008], [Yousef et al. 2009], [Alizadeh et al. 2011], [Kim and Suhh 2010], [Kannan and Srivastava 2008], [Brzostowski et al. 2010] focus on identification of both relationships and resources. While *source code analysis* [Alahmari et al. 2010], [Li and Tahvildari 2006], [Jain et al. 2004], [Chang 2007], [Li and Tahvildari, 2008], [Bao et al. 2010], [Chen et al. 2005], [Zhang and Yang 2004], [Aversano et al. 2008], [Baghdadi 2006], [Patig and Wesenberg 2009], [Cho et al. 2008], [Shiang et al. 2009], [Chen et al. 2009], [Inaganti and Behara 2007], [Chung et al. 2009], [Erradi et al. 2006], [Nakamura et al. 2011], [Komondoor et al. 2012], [Abdelkader et al. 2013], [Fuhr et al. 2013] and *wrapping* [Ricca and Marchetto 2009], [Canfora et al. 2008], [Zhang et al. 2005], [Upadhyaya et al. 2012] tend to identify legacy resources (source code, data and existing services) that might be used in implementing a service.

Finally, *pattern matching* [Gacitua-Decar and Lero 2009], [Aier 2006] and *value analysis* [Andersson et al. 2009], [Samavi et al. 2008], [Gordijn et al. 2006], [Ilayperuma and Zdravkovic 2010], [Lo and Yu 2007], [Han et al. 2009], [Flaxer and Nigam 2004] usually do not identify services' resources or relationships. The focus is on defining which operations should be grouped within a service or the service purpose, respectively.

4.2.1.9. Type of conversation

Services can be composed in a variety of ways, including direct consumer-to-service interaction by using programming techniques, or they can be aggregated by means of an aggregation approach such as choreography or orchestration. Choreography is used to characterize and to compose business collaborations based on ordered message exchanges between peer entities in order to achieve a common business goal. Orchestration is used to compose hierarchical and self-contained service-oriented business processes that are executed and coordinated by a single agent [OASIS].

Table 13: Type of conversation perspective.

Type of conversation	SIMs
None	[Abdelaziz et al. 2011], [Abdelkader et al. 2013], [Abu-Matar and Gomaa 2011], [Adamopoulos et al. 2002], [Aier 2006], [Alahmari et al. 2010], [Alizadeh et al.

	2011], [Andersson et al. 2009], [Asadi et al. 2011], [Aversano et al. 2008], [Baghdadi 2006], [Bao et al. 2010], [Bianchini et al. 2009], [Bianchini et al. 2013], [Birkmeier et al. 2013], [Brzostowski et al. 2010], [Canfora et al. 2008], [Chaari et al. 2007], [Chen et al. 2005], [Chen et al. 2009], [Chung et al. 2009], [de Bruin et al. 2009], [DongSu et al. 2008], [Erradi et al. 2006], [Erradi et al. 2007], [Fareghzadeh 2008], [Flaxer and Nigam 2004], [Fuhr et al. 2013], [Gacitua-Decar and Lero 2009], [Gordijn et al. 2006], [Guan et al. 2012], [Han et al. 2009], [Huayou et al. 2009], [Ilayperuma and Zdravkovic 2010], [Inaganti and Behara 2007], [Jain et al. 2004], [Jamshidi et al. 2008], [Kannan and Srivastava 2008], [Kazemi et al. 2011a], [Kazemi et al. 2011b], [Kenzi et al. 2008], [Kim and Doh 2007], [Kim and Doh 2009], [Kim and Doh 2012], [Kim and Doh 2013], [Kim and Suhh 2010], [Kim et al. 2008], [Klose et al. 2007], [Kohlmann and Alt 2007], [Komondoor et al. 2012], [Li and Tahvildari 2006], [Li and Tahvildari, 2008], [Liu et al. 2011], [Mani et al. 2008], [Nakamura et al. 2011], [Nguyen et al. 2009], [Park et al. 2009], [Patig and Wesenberg 2009], [Ruiz et al. 2006], [Samavi et al. 2008], [Schmidt 2011], [Sewing et al. 2006], [Souza et al. 2009], [Souza et al. 2011], [Stewart and Chakraborty 2010], [Upadhyaya et al. 2012], [Vemulapalli and Subramanian 2009], [Wang et al. 2005], [Yousef et al. 2009], [Yun et al. 2009], [Zhang and Yang 2004], [Zhang et al. 2010]
Both	[Arsanjani et al. 2008], [Dwivedi and Kulkarni 2008], [Medeiros et al. 2010]
Choreography	[Chang 2007], [Jin and Zhu 2008], [Ponnalagu and Narendra 2008], [Shiang et al. 2009], [Zhang et al. 2005]
Orchestration	[Asadi et al. 2009], [Caetano et al. 2007], [Cho et al. 2008], [Dinh and Nguyen-Ngoc 2010], [Jamshidi et al. 2009], [Kaabi et al. 2004], [Lee et al. 2010], [Lee et al. 2011], [Lo and Yu 2007], [Meertens et al. 2010], [Menascé et al. 2008], [Mosser et al. 2011], [Ren and Wang 2010], [Ricca and Marchetto 2009], [Shirazi et al. 2009], [Weigand 2011], [Weigand et al. 2009], [Wen et al. 2010]

SIMs can detail the type of conversation as orchestration, choreography by using both types or using none. According to Table 13, most studies does not detail the type of conversation between services. Among the studies that detail this perspective, the majority is top-down approaches. Studies that mention both types of conversation do not detail how service collaborations are identified or implemented. SIMs that mention choreography identify service collaborations by interactions flows [Jin and Zhu 2008], [Ponnalagu and Narendra 2008], [Shiang et al. 2009] or define service adapters' types to enable dynamic composition [Chang 2007]. Service orchestrations are also identified by interactions flows [Ricca and Marchetto 2009], [Shirazi et al. 2009], [Dinh and Nguyen-Ngoc 2010], [Lee et al. 2010], [Menascé et al. 2008], [Lee et al. 2011], [Lo and Yu 2007], [Meertens et al. 2010], [Caetano et al. 2007], [Asadi et al. 2009], [Wen et al. 2010], [Ren and Wang 2010], [Cho et al. 2008], [Weigand 2011], [Jamshidi et al. 2009], [Weigand et al. 2009] or by defining composition patterns [Kaabi et al. 2004], [Mosser et al. 2011]. We did not find any correlation between conversation type detailing and the technique or approach employed by the SIM.

4.2.1.10. Quality attributes elicitation

This perspective is related to the elicitation of the quality attributes that influence the design, policies or execution contexts of the services. Quality is dependent of the stakeholders' requirements, but some general service quality attributes can be identified in a SOA context. Erl [Erl 2007] emphasizes that the basic software quality design principles of low coupling and high cohesion should be observed during all service creation cycle. Service granularity is also pointed out as a quality attribute, because the granularity level of a service can affect its capabilities, performance, reusability and coupling.

SIMs deal with service candidate quality by using metrics of coupling [Bianchini et al. 2009], [Bianchini et al. 2013],[Kazemi et al. 2011a], [Medeiros et al. 2010], cohesion [Abdelkader et al. 2013], [Bianchini et al. 2009], [Bianchini et al. 2013], [Kazemi et al. 2011a], [Medeiros et al. 2010], granularity [Kazemi et al. 2011a], [Kim and Doh 2012], modularity [Kazemi et al. 2011b], [Li and Tahvildari 2006], reusability using the semantic distance between features [DongSu et al. 2008] and QOS by estimating a weight to execute activities [Menascé et al. 2008]. Only 10 studies identify quality attributes as presented in Table 14:

Table 14: Quality attributes elicitation perspective.

Quality attribute elicitation	SIMs
Coupling	[Bianchini et al. 2009], [Bianchini et al. 2013], [Kazemi et al. 2011a], [Medeiros et al. 2010]
Cohesion	[Abdelkader et al. 2013], [Bianchini et al. 2009], [Bianchini et al. 2013], [Kazemi et al. 2011a], [Medeiros et al. 2010]

Granularity	[Kazemi et al. 2011a], [Kim and Doh 2012]
Modularity	[Kazemi et al. 2011b], [Li and Tahvildari 2006]
Reusability	[DongSu et al. 2008]
QOS	[Menascé et al. 2008]

Table 14 shows that few of the reported SIMs address service quality attributes, suggesting the service identification research field is still not mature. Almost all SIMs intend to identify candidate services, but they usually do not assess their quality, nor do any effort to improve identified candidates. Quality attribute elicitation seems not to be related to the technique or approaches, being a consequence of the scope chosen by the SIM.

4.2.2. Evaluation of the techniques employed in the service identification process

In order to answer the RQ2, the studies were categorized in the light of software engineering techniques employed in the service identification process and the proposed classification scheme (Section 4.1). To categorize each technique, we used the "High-Value activities" criteria proposed in [Cai et al. 2011], with some improvements: (i) the addition of Product line approach and Artifact-centric approach techniques, and (ii) the adjustments of the delivery strategy for Ontology mapping and Service classification techniques in order to include the bottom-up strategy, because these techniques can be used both with top-down and bottom-up strategies. Table 15 presents all selected techniques. For each one, we show their description, artifacts generated by them, and the delivery strategy adopted (top-down or bottom-up).

Table 15: Techniques and descriptions.

Technique	**(*)**	**Description**	**Artifacts**
Model-driven approach	T	Based on generating various business models based on requirements and business rules. Transform the business models to business services.	CIM/PIM, business models, Service model
Decomposition	T	Based on decomposing high-level, coarse grained processes/ goals/features etc. into low-level, fine-grained sub processes/goals/features etc.	Sub processes/ goals/features etc.

Value analysis	T	Based on the analysis of business value and value change with value chain, value network etc. to decide which business services are of high value.	Value model, high-value services
Pattern matching	T	Identify services from business models by matching existing business patterns.	Business services
Service responsibility analysis	T	Make use of tools such as service responsibility table to identify service owner and responsibilities.	Service responsibility specification
Product line approaches	T	Identifying commonalities or refine variation points from business level or from a portfolio of existing services by specifying the action semantics of the variation and its specific applicability.	Feature model, variability model
Artifact-centric approach	T	Identify services from the information model and the lifecycle of business entities.	Information model, information lifecycle model
SODA (service oriented design aspect)	T	Based on the decomposition of interactions, concerns and features into aspects and composing them according to requirements.	Aspect specification
Assets identification	B	Based on the analysis of informative entities (diagrams, schemas, models, documents etc.) of legacy system to identify reusable assets.	Reusable assets
Source code analysis	B	Based on the analysis of structure, data flow and dependency in source code.	Software Services
Wrapping	B	Wrapping existing applications directly as candidate services.	Wrapped services
Ontology mapping	T/B	Based on the definition of a set of ontology to help identify business service from business requirements or legacy documentation.	Business services
Requirement analysis approaches	T	Other activities in requirement engineering for requirement elicitation, such as scenario and gap analysis etc.	Refined requirements, gap report
Service Classification/ categorization	T/B	Based on the classification services to different types such as process service, composite service, molecular service etc. to help decide the function and granularity of services.	Classified business services

(*)Delivery strategy: T – *Top-down*, B – *Bottom-up*

Decomposition is the most used technique with 38 occurrences among the analyzed SIMs, followed by *Source code analysis* (27), and *Value analysis* (13). More than a half of the studies combine two or more techniques. The most used combination is *Decomposition* and *Source code analysis*, which is usually applied in *meet in the middle* approaches. *Service classification, SODA, Asset identification,* and *Wrapping* are always used together with other techniques, but only *Service classification* acts as a complimentary technique, often used to guarantee services with right granularity and high cohesion.

Service classification and *Ontology mapping* techniques can be applied in both top-down and bottom-up strategies. Considering all techniques, the *top-down* is the most used approach, being employed by 56 SIMs, followed by the *bottom-up* with 30 studies. The *meet in the middle* approach is less used, being employed only by 12 studies.

Table 16 was designed to correlate the multiple perspectives of OASIS' reference architecture for SOA described in Section 4.1 with the software engineering techniques employed by existent SIMs. Table 16 aims to assist practitioners to reason about how the techniques can be applied to the service identification process in the light of the OASIS' reference architecture classification scheme. The techniques presented in Table 15 (except service classification since it is a complementary technique) were analyzed in order to find out influences or correlations to each perspective. Since we found out no correlation among Service granularity, Type of conversation and Quality attributes elicitation perspectives and software engineering techniques employed by existent SIMs they are not listed in Table 16.

Table 16 : Techniques and classification perspectives.

Technique/Classification perspective	Participant concern	Context of transactions	Service description	Service Value	Behavior model	Information model	Service dependency
Decomposition	Reutilization only	Inside	Syntax-based	Both	Internal actions sequence	No tendency	Both (resources and service relationship)
Model-driven architecture	Reutilization only	Inside	Syntax-based	Both	Internal events, pre and post conditions	Information messages and parameters structures	Relationship
Value analysis	Reutilization only	Outside	Syntax-based	Direct	External behavior only	Not detailed	Relationship
Ontology-mapping	Reutilization only	Inside	Semantic-based	Both	Internal events, pre and post conditions	Semantic description	Both (resources and service relationship)
Pattern matching	Reutilization only	Inside	Syntax-based	Both	No tendency	Not detailed	None

Product line	Implemen-tation	Inside	Syntax-based	Both	No tendency	Not detailed	Relationship
Requirements analysis	Reutiliza-tion only	Inside	Syntax-based	Both	Internal actions sequence	No tendency	Relationship
Service responsibility analysis	Reutiliza-tion only	Outside	Syntax-based	Direct	Internal events, pre and post conditions	Information structure description	Relationship
SOAD	Reutiliza-tion only	Inside	Syntax-based	Both	No tendency	No tendency	Relationship
Source code analysis	Implemen-tation	Inside	Syntax-based	Indirect	Internal actions sequence	Information structure description	Resources
Wrapping	Implemen-tation	Inside	Syntax-based	Indirect	Internal actions sequence	Information messages and parameters structures	Both (resources and service relationship)
Assets identification	Reutiliza-tion only	Inside	Syntax-based	Indirect	No tendency	Information structure description	Resources

Although we have identified no direct correlations, some relevant observations can be made from the analysis of these perspectives. The identification of services with different granularities (Service granularity perspective) can be accomplished by segregating services in layers as suggested in [Shirazi et al. 2009]. A precondition to address the Type of conversation perspective is to elicit service relationship. Therefore, a technique that enables such type of identification must be chosen (for instance, Model-driven approach or Value analysis). Quality attributes elicitation perspectives can be addressed by defining metrics to assess services quality accordingly to the requirements elicited, independently of the technique applied. The remainder perspectives are addressed according to the correlations between techniques and classification perspectives presented in Table 16.

For example, as described in Section 4.2.1.1, the Participant perspective presents two concerns: reutilization issues only and implementation and reutilization issues. If a practitioner wants to know which Participant concerns can be used from the Decomposition technique, Table 16 shows that the answer is Reutilization only. Moreover, if a practitioner wants to identify services' implementation concerns and the resources required or provided by them (service dependency perspective), the Table indicates that (s)he can use Product Line or Source code analysis or Wrapping combined with Decomposition or Ontology Mapping or Assets Identification.

Table 16 can also assist practitioners in the selection of software engineering techniques to be used depending on the type of project they are developing. For example, if a practitioner is in charge of a SOA project focused on delivering fast results, s(h)e can choose the Source code analysis technique, since such technique favors the elicitation of implementation concerns (from Participant concerns perspective) contributing to speed up the construction process. Besides, in this same example, according to Table 16, the identified services will be indirectly related to business (Service Value perspective). Therefore, such services tend to deliver value to IT department instead of the core enterprise business. As another example, if the practitioner is involved in a SOA project to integrate inter-organizational processes Table 16 tells s(h)e that Value Analyses or Service Responsibility Analysis must be used because they are the only ones capable of eliciting cross enterprise interactions (Context of Transaction perspective).

These examples show how this type of correlation can be a useful tool to shed light on the consequences of practitioner decisions of using different software engineering techniques in the service identification phase of SOA projects.

5. Threats to Validity

The main threats to the validity of this survey are related to the:

- *Its completeness*. The electronic databases used in this systematic survey (see Table) are considered the most relevant available sources [Kitchenham and

Charters 2007], but some studies may have been missed due to technical limitations of the search engines themselves, which are out of the control of the researchers. Furthermore, these databases do not represent an exhaustive list of publication sources, so that other databases might also be included.

- *Reviewers' reliability*. Although the conclusions might have been influenced by the researchers' opinions, it was adopted a dual-revision strategy in order to minimize the effect of any bias or misinterpretation. Therefore, the studies were evaluated more than once, each time by a different researcher.

- *Data extraction*. Since not all information was obvious to answer the established research questions, some data had to be interpreted. Nevertheless, discussions were conducted whenever a disagreement between the researchers occurred in order to ensure the validity of this systematic survey.

6. Conclusion

This paper reported the results of a systematic survey that quantitatively characterizes service identification methods published from 2002 to June 2013. Four electronic libraries were used and we have identified 105 studies reporting service identification methods. A classification scheme based on a reference architecture adopted by industry was proposed as a way to suggest which issues should be considered during services identification phase and how the existing approaches address them. SIMs differ in the way they address the proposed classification perspectives. Nevertheless, the evaluation presented in Section 4.2.1 demonstrates that existing SIMs address many perspectives of the adopted SOA reference architecture, thus suggesting that these SIMs are aligned with the concerns related to SOA adoption in the context of service identification phase.

More than a half of the proposed methods use more than one software engineering technique, but few are meet in the middle approaches. Meet in the middle approaches are the most complete one since they evaluate the models from the highest level to the most detailed, thus allowing reuse of existing assets (services and applications) as well as the generation of fine-grained services (more reusable) and coarse grained services that generate immediate value to the business.

The technique chosen by the SIM can influence on how each classification perspective is addressed. Perspectives as *Participant Concern, Context of Transactions, Service Description, Behavior Model, Information Model* and *Service Dependency* have a correlation with the technique. *Service Value* perspective is influenced by the input used by the technique, but not by the technique itself, and perspectives as *Service Granularity, Conversation Type* and *Quality attributes elicitation* seem not to have correlation with the techniques. Furthermore, we intend to aid practitioners to understand the consequences of using software engineering techniques employed by different SIMs and also to encourage researchers to promote improvements in this field by combining techniques or creating new ways to address the service identification perspectives.

Finally, the majority of evaluated SIMs does not elicit service quality attributes, suggesting service identification research field is still not mature, especially regarding the quality of the identified services candidates. Therefore, the results of our study suggest that future research should follow the direction of hybrid methods.

References:

ABDELAZIZ, E.F. et al. A service-oriented approach for information systems development, Multimedia Computing and Systems, ICMCS (2011) 1-6.

ABDELKADER, M. et al. A heuristic approach to locate candidate web service in legacy software, International Journal of Computer Applications in Technology 47 (2013) 152-161.

ABELNEH Y. et al. A systematic literature review on service description methods, Proceedings of the 18th international conference on Requirements Engineering: foundation for software quality REFSQ'12 (2012) 239-255.

ABU-MATAR, M., GOMAA, H. Variability Modeling for Service Oriented Product Line Architectures, Software Product Line Conference (SPLC), 2011 15th International, (2011) 110-119.

ACM Digital Library. http://dl.acm.org. Last access: December 2013.

ADAMOPOULOS, D.X. et al. Advanced service creation using distributed object technology, IEEE Communications Magazine 40, 3 (2002) 146-154.

AIER, S. How clustering enterprise architectures helps to design service oriented architectures, Proceedings - 2006 IEEE International Conference on Services Computing, SCC (2006) 269-272.

ALAHMARI, S. et al. A Model-Driven Architecture Approach to the Efficient Identification of Services on Service-oriented Enterprise Architecture, Enterprise Distributed Object Computing Conference Workshops (EDOCW) (2010) 165-172.

ALIZADEH, K. et al. Mapping service concept and enterprise ontology in service identification, Networked Computing (INC) (2011) 22-27.

ANDERSSON, B. et al. Aligning goals and services through goal and business modeling, Information Systems and E-Business Management 7 (2) (2009) 143-169.

ARSANJANI, A. et al. SOMA: A method for developing service-oriented solutions, IBM Systems Journal 47 (3) (2008) 377-396.

ARSANJANI, A. Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA (2004) In: http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/ Accessed in: August 2013.

ASADI, M et al. Model-driven development of families of service-oriented architectures, ACM International Conference Proceeding Series (2009) 95-102.

ASADI, M. et al. Developing families of method-oriented architecture, IFIP Advances in Information and Communication Technology, 351 AICT (2011) 168-183.

AVERSANO, L. et al. Mining candidate web services from legacy code, Web Site Evolution WSE (2008) 37-40.

BAGHDADI, Y. Reverse engineering relational databases to identify and specify basic Web services with respect to service oriented computing, Information Systems Frontiers 8 (5) (2006) 395-410.

BAO, L. et al. Extracting reusable services from legacy object-oriented systems, Software Maintenance (ICSM), IEEE International Conference, (2010) 1-5.

BIANCHINI, D. et al. P2S: A methodology to enable inter-organizational process design through web services, International Conference on Advanced Information Systems (2009) 334-348.

BIANCHINI, D. et al. Service Identification in Inter-Organizational Process Design, IEEE Transactions on Services Computing (2013).

BIRKMEIER, D. et al. A Survey of Service Identification Approaches - Classification Framework, State of the Art, and Comparison, Enterprise Modeling and Information Systems Architectures 4 (2) (2009) 20-36.

BIRKMEIER, D.Q. et al. Alignment of Business and IT Architectures in the German Federal Government: A Systematic Method to Identify Services from Business Processes, Proceedings of the 2013 46th Hawaii International Conference on System Sciences (2013) 3848-3857.

BOERNER, R., GOEKEN, M. Service identification in SOA governance literature review and implications for a new method, Int. Conference on Digital Ecosystems and Technologies (2009) 588–593.

BRZOSTOWSKI, K. et al. Service discovery in the SOA system, Proceedings of the Second international conference on Intelligent information and database systems: Part II (ACIIDS'10) (2010) 29-38.

CAETANO, A. et al. Identification of Services through Functional Decomposition of Business Processes, Business Information Systems Lecture Notes in Business Information Processing 47 (5) (2010) 144-157.

CAI, S. et al. A Survey of Service Identification Strategies, APSCC (2011) 464-470.

CANFORA, G. et al. A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures, Journal of Systems and Software 81 (4) (2008) 463-480.

CHAARI, S. et al. Towards a service-oriented enterprise based on business components identification, Enterprise Interoperability II (2007) Part V, 495-506.

CHANG, S.H. A Systematic Analysis and Design Approach to Develop Adaptable Services in Service Oriented Computing, In: Services, 2007 IEEE Congress on, 2007, 375-378.

CHEN, F. et al. Feature analysis for service-oriented reengineering, Software Engineering Conference, APSEC '05. 12th Asia-Pacific (2005) 8 pp.

CHEN, F. et al. Identification via Ontology Mapping, Computer Software and Applications Conference, COMPSAC '09 33rd Annual IEEE International (2009) 486-491.

CHO, M. et al. Service identification and modeling for service oriented architecture applications, Proceedings of the 7th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (2008) 193-199.

CHUNG, S. et al. Service-oriented reverse reengineering: 5W1H model-driven re-documentation and candidate services identification, Service-Oriented Computing and Applications, SOCA (2009) 1-6.

DBLP Computer Science Bibliography. http://www.dblp.org/search/index.php. Last access: December 2013.

DE BRUIN, J. et al. On the Design of Knowledge Discovery Services Design Patterns and Their Application in a Use Case Implementation, Leveraging Applications of Formal Methods, Verification and Validation, Communications in Computer and Information Science 17 (2009) p. 649.

DEMIRKAN, H et al. Service-oriented technology and management: Perspectives on research and practice for the coming decade, Electronic Commerce Research and Applications 7 (4) (2008) 356-376.

DIESTE et al. Developing search strategies for detecting relevant experiments. Empirical Software Engineering 14(5), (2009) 513-539

DINH, T.L., NGUYEN-NGOC, A.V. A conceptual framework for designing service-oriented inter-organizational information systems, Proceedings of the 2010 Symposium on Information and Communication Technology, ACM (2010) 147-154.

DONGSU, K. et al. A Method of Service Identification for Product Line, Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology 2 (2008) 1040-1045.

DYBÅ T et al. Applying systematic reviews to diverse study types: An experience report. Proc. of the First Int. Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, Washington, DC, USA, (2007) pp 225-234

DWIVEDI, V., KULKARNI, N. A Model Driven Service Identification Approach for Process Centric Systems, Congress on Services Part II (2008) 65-72.

ERL, T. SOA Principles of Service Design, first ed., Prentice Hall, 2007.

ERRADI, A. et al. Service Design Process for Reusable Services: Financial Services Case Study, Proceedings of the 5th international conference on Service-Oriented Computing (2007) 606-617.

ERRADI, A. et al. SOAF: An Architectural Framework for Service Definition and Realization, Services Computing, SCC '06 IEEE International Conference (2006) 151-158.

FAREGHZADEH, N. Service Identification Approach to SOA Development, World Academy of Science, Engineering and Technology 45 (2008).

FLAXER, D., NIGAM, A. Realizing Business Components, Business Operations and Business Services, Proceedings of the E-Commerce Technology for Dynamic E-Business, IEEE International Conference (2004) 328-332.

FUHR, A. et al. Model-driven software migration into service-oriented architectures, Computer. Science, 28 (2013) 65-84.

GACITUA-DECAR, V. LERO, C.P. Automatic Business Process Pattern Matching for Enterprise Services Design, Services - II, SERVICES-2 '09 World Conference (2009) 111-118.

GORDIJN, G.J. et al. E-Service Design Using i* and e3 value Modeling, IEEE Software 23 (2006) 26-33.

GU, Q. LAGO, P. Service Identification Methods: A Systematic Literature Review, ServiceWave (2010) 37-50.

GUAN, Q., et al. A Network Topology Clustering Algorithm for Service Identification, Proceedings of the 2012 International Conference on Computer Science and Service System (2012) 1583-1586.

HAN, F. et al. Organizational Interoperability Supported through Goal Alignment with BMM and Service Collaboration with SoaML, Proceedings of the 2009 International Conference on Interoperability for Enterprise Software and Applications China (2009) 268-274.

JABREF http://jabref.sourceforge.net/ Last access: December 2013.

HUANG, C. et al. A semantic enhanced service exposure model for a converged service environment, Communications Magazine, IEEE, 50 (3) (2012) 32-40.

HUAYOU, S. et al. A Service-Oriented Analysis and Modeling Using Use Case Approach, Computational Intelligence and Software Engineering, CiSE (2009) 1-6.

IEEE Xplore. http://ieeexplore.ieee.org. Last access: December 2013.

ILAYPERUMA, T., ZDRAVKOVIC, J. Exploring business value models from the inter-organizational collaboration perspective, Proceedings of the 2010 ACM Symposium on Applied Computing, SAC'10 (2010) 99-105.

INAGANTI, S., BEHARA, G.K. Service Identification: BPM and SOA Handshake (2007) In: http://w.bptrends.com/publicationfiles/THREE%2003-07-ART-BPMandSOAHandshake-InagantiBehara-Final.pdf (last accessed August 2013)

JAIN, H. et al. A Spanning Tree Based Approach to Identifying Web Services, International Journal of Web Services Research 1 (2004) 1-20.

JAMSHIDI, P. et al. To Establish Enterprise Service Model from Enterprise Business Model, Services Computing, SCC '08. IEEE International Conference (2008) 93-100.

JAMSHIDI, P. et al. Toward automatic transformation of enterprise business model to service model, Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, IEEE Computer Society (2009) 70-74.

JIN, Z. ZHU, H. A framework for agent-based service-oriented modeling, Proceedings of the 4th IEEE International Symposium on Service-Oriented System Engineering, SOSE (2008) 160-165.

KAABI, R.S., et al. Eliciting service composition in a goal driven manner, Proceedings of the 2nd international conference on Service oriented computing, ACM (2004) 308-315.

KANNAN, K., Srivastava, B. Promoting Reuse via Extraction of Domain Concepts and Service Abstractions from Design Diagrams, Services Computing SCC '08. IEEE International Conference (2008) 265-272.

KAZEMI, A. et al. A Genetic Algorithm Based Approach to Service Identification, Proceedings of the 2011 IEEE World Congress on Services (2011a) 339 -346.

KAZEMI, A. et al. A metric suite for measuring service modularity, Computer Science and Software Engineering (CSSE) (2011b) 95-102.

KENZI, A. et al. Multi-functional service oriented system development for user-driven adaptability, Information and Communication Technologies: From Theory to Applications, ICTTA 2008 3rd International Conference, (2008) 1-7.

KIM, G., SUHH, Y. Ontology-based semantic matching for business process management, ACM SIGMIS Database 41 (4) (2010) 98-118.

KIM, S. et al. Service Identification Using Goal and Scenario in Service Oriented Architecture, Software Engineering Conference, APSEC '08. 15th Asia-Pacific (2008) 419-426.

KIM, Y. DOH, K. The service modeling process based on use case refactoring, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2007) 108-120.

KIM, Y., DOH, K. Formal Identification of Right-Grained Services for Service-Oriented Modeling, Proceedings of the 10th International Conference on Web Information Systems Engineering (2009) 261-273.

KIM, Y., DOH, K. Pragmatic granularity decision for right-grained services in service-oriented modeling, International Journal Web Grid Services 8, 2 (2012) 111-133.

KIM, Y., DOH, K. Use-case driven service modeling with XML-based tailoring for SOA, International Journal of Web and Grid Services 9, 1 (2013) 35-53.

KITCHENHAM B. Procedures for performing systematic reviews. Technical report, Keele University/National ICT Australia, Ltd (2004).

KITCHENHAM B. et al. Evidence-Based Software Engineering. Proc. of the 26th Int. Conf. on Software Engineering, IEEE Computer Society, Washington, DC, USA, (2004) pp 273-281

KITCHENHAM B., CHARTERS S. Guidelines for performing systematic literature reviews in Software Engineering. Technical report, Keele University/University of Durham (2007).

KITCHENHAM, B. et al. Systematic literature reviews in software engineering – A systematic literature review, Information and Software Technology 51 (1) (2009) 7-15.

KLOSE, K. et al. Identification of Services - A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning, Proceedings of the 15th European Conference on Information Systems, ECIS (2007) paper 116.

KOHLBORN, T. et al. Service analysis - A critical assessment of the state of the art, Proceedings of ECIS'2009 (2009) 1583-1594.

KOHLMANN, F., Alt, R.: Business-Driven Service Modeling - A Methodological Approach from the Finance Industry, SABRE 2007 (2007) pp. 14.

KOMONDOOR, R. et al. Identifying services from legacy batch applications, Proceedings of the 5th India Software Engineering Conference (2012) 13-22.

LEE, J. et al. A feature-oriented approach for developing reusable product line assets of service-based systems, Journal of Systems and Software 83 (2010) 1123-1136.

LEE, J. et al. An Approach for Service Identification Using Value Co-creation and IT Convergence, Computers Networks Systems and Industrial Engineering, CNSI (2011) 441-446.

LEE, T.B. et al. The Semantic Web, The Scientific American, May 2001.

LI, S. TAHVILDARI, L. E-BUS: a toolkit for extracting business services from java software systems, Companion of the 30th international conference on Software engineering, ACM (2008) 961-962.

LI, S., TAHVILDARI, L. A service-oriented componentization framework for Java software systems, Proceedings - Working Conference on Reverse Engineering, WCRE (2006) 115-124.

LIU, W. et al. Use Case-Based Service-Oriented Analysis and Modeling, Internet Computing Information Services ICICIS (2011) 94-96.

LO, A. YU, E. From business models to service-oriented design: a reference catalog approach, Proceedings of the 26th international conference on Conceptual modeling ER'07 (2007) 87-101.

MANI, S. et al. Using User Interface Design to Enhance Service Identification, Web Services ICWS '08, IEEE International Conference (2008) 78-87.

MEDEIROS, F.M. et al. Designing a set of service-oriented systems as a Software Product Line, Proceedings - 4th Brazilian Symposium on Software Components, Architectures and Reuse, SBCARS (2010) 70-79.

MEERTENS, L.O. et al. Goal and model driven design of an architecture for a care service platform, Proceedings of the 2010 ACM Symposium on Applied Computing, ACM (2010) 158-164.

MENASCÉ, D.A. et al. A heuristic approach to optimal service selection in service oriented architectures, Proceedings of the 7th international workshop on Software and performance, ACM (2008) 13-24.

MOSSER, B. et al. From aspect-oriented requirements models to aspect-oriented business process design models: An iterative and concern-driven approach for software engineering, Proceedings of the 10th International Conference on Aspect-Oriented Software Development, AOSD.11 (2011) 31-42.

NAKAMURA, M. et al. Identifying Services in Procedural Programs for Migrating Legacy System to Service Oriented Architecture, International Journal of Information Systems in the Service Sector, Vol. 3 (2011) 54-72.

NGUYEN, D.K. et al. Gap analysis methodology for business service engineering, 2009 IEEE Conference on Commerce and Enterprise Computing, CEC (2009) 215-220.

PARK, J. et al. An approach to enhancing reusabilities in service development, Proceedings of the 2009 International Conference on Hybrid Information Technology (2009) 143-150.

PATIG, S., WESENBERG, H. Role of process modeling in software service design, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2009) 420-428.

PONNALAGU, K., NARENDRA, N.C. Deriving service variants from business process specifications, Proceedings of the 1st Bangalore Annual Compute Conference, ACM (2008) 4:1-4:9.

OASIS Reference Architecture Foundation for Service Oriented Architecture. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-v1.0.pdf. Last access: December 2013.

REN, M. WANG, Y. Rule based business service identification using UML analysis, Information Management and Engineering (ICIME) (2010) 199-204.

RICCA, F., MARCHETTO, A. A "quick and dirty" meet-in-the-middle approach for migrating to SOA, Proceedings of the joint international and annual ERCIM workshops on Principles of software evolution (IWPSE) and software evolution (Evol) workshops, ACM (2009) 73-78.

RUIZ, M. et al Designing web services for supporting user tasks: A model driven approach, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4231 LNCS (2006) 193-202.

SAMAVI, R. et al. Applying strategic business modeling to understand disruptive innovation, Proceedings of the 10th international conference on Electronic commerce, ACM, (2008) 15:1-15:10.

SCHMIDT, R. A Service-System Based Identification of Meta-services for Service-Oriented Enterprise Architecture, Enterprise Distributed Object Computing Conference Workshops, EDOCW (2011) 293-300.

SCOPUS http://www.scopus.com . Last access: December 2013.

SEWING, J. et al. Process-Oriented Assessment of Web Services, International Journal of E-Business Research, 2, 1 (2006) 19-44.

SHIANG, W. et al. Service identification of a collaborative design workflow in a dynamically changing environment, Networking, Sensing and Control, ICNSC '09 |(2009) 685-690.

SHIRAZI, H.M. et al. A combinational approach to service identification in SOA, Journal of Applied Sciences Research (2009) 1390-1397.

SOUZA, A. et al. A method for service identification from business process models in a SOA approach, Lecture Notes in Business Information Processing (2009) 99-112.

SOUZA, A. Service identification in aspect-oriented business process models, Service Oriented System Engineering (SOSE) IEEE 6th International Symposium (2011) 164-174.

STEWART, G., CHAKRABORTY, A. Service identification through value chain analysis and prioritization, Proceedings of the 16th Americas Conference on Information Systems (2010).

UPADHYAYA, B., et al. Extracting RESTful services from Web applications, Proceedings of the 2012 5th IEEE International Conference on Service-Oriented Computing and Applications – SOCA (2012) 1-4.

VALE, T. et al. A study on service identification methods for software product lines, Proceedings of the 16th International Software Product Line Conference (SPLC '12), Vol. 2 (2012) 156-163.

VEMULAPALLI, A., SUBRAMANIAN, N. Transforming Functional Requirements from UML into BPEL to Efficiently Develop SOA-Based Systems, Proceedings of the Confederated International Workshops and Posters on On the Move to Meaningful Internet Systems (2009) 337-349.

WANG, Z. et al. Normal Forms and Normalized Design Method for Business Service, Proceedings of the IEEE International Conference on e-Business Engineering, ICEBE '05 (2005) 79-86.

WEIGAND, H. et al. Value-Based Service Modeling and Design: Toward a Unified View of Services, CAiSE '09 Proceedings of the 21st International Conference on Advanced Information Systems Engineering (2009) 410-424.

WEIGAND, H. The pragmatics of event-driven business processes, Proceedings of the 7th International Conference on Semantic Systems, ACM (2011) 211-218.

WEN, B. et al. Process semantic-enabled customisation for active service provisioning, Proceedings - 2010 13th IEEE International Conference on Computational Science and Engineering, CSE (2010) 375-381.

YOUSEF, R. et al. BPAOntoSOA: A generic framework to derive software service oriented models from business process architectures, Applications of Digital Information and Web Technologies ICADIWT'09 (2009) 50-55.

YUN, Z. et al. A Service-Oriented Analysis and Design Approach Based On Data Flow Diagram, Computational Intelligence and Software Engineering, CiSE (2009) 1-5.

ZADEH, A.T. et al. A Systematic Input Selection for Service Identification in SMEs, Journal of Applied Sciences 12 (2012) 1232-1244.

ZHANG, Z. et al. A service composition approach based on sequence mining for migrating e-learning legacy system to SOA, International Journal of Automation and Computing 7 (4) (2010) 584-595.

ZHANG, Z. et al. Service Identification and Packaging in Service Oriented Reengineering, In Proceedings of the 7th International Conference on Software Engineering and Knowledge Engineering (SEKE) (2005) 241-249.

ZHANG, Z., YANG, H. Incubating services in legacy systems for architectural migration, Software Engineering Conference, 11th Asia-Pacific (2004) 196-203.

APPENDIX B – MDCSIM PLUG-IN CODE

The transformation *IdentifyServices.atl* implements the rules described in the steps 5.2.4 and 5.2.6 in order to identify candidate services. The complete code is presented as follows:

```
--@path MM=/Model2Service/Estereotipos.profile.uml
--@path MM1=/Model2Service/Estereotipos.profile.uml

-- Created in 24/10/2013. Author: Rosane Sfair Huergo
-- This transformation reads a class diagram and state machines diagrams and
creates a class diagram of suggested Entity services, Task services and Process
services.
-- The input class diagram describes master data of a business domain and its
relatioships with other master data or classes that do not represent master data.
-- The state machines describe the behavior of each master data. Master data that
do not change states do not need a state machine diagram as input.

module IdentifyServices;
create OUT : MM1 from IN : MM, IN2 : MM1;

helper def: operations : Sequence(String) =
Sequence{'create','read','update','delete'};
helper def: attribute : Sequence(String) = Sequence{};
helper def: opentry: MM1!Operation = OclUndefined;
helper def: param : MM1!Parameter = OclUndefined;
helper def: inputmodel : MM1!Model  = OclUndefined;
helper def: controller : MM1!Class = OclUndefined;
helper def: processService : MM1!Class = OclUndefined;
helper def: methods: Sequence(String) = Sequence{};
helper context OclAny def: hasStereotype(name : String) : Boolean =
	not self.getAppliedStereotype(name).oclIsUndefined();


rule modelo {
	from
		i : MM!Model
	to
		o : MM1!Model (
			packagedElement <- i.packagedElement
		)

	do {

		-- Retrieves the classes marked with the stereotype Master Data to
create Entity services.
		for(c in i.packagedElement -> select(h |
h.hasStereotype('Profile::MasterData'))){

			thisModule.inputmodel <- OclUndefined;

			-- Retrieves the State Machine diagram of the Master data
classes
```

```
                    for (diag in  MM1!Model.allInstances()->select(m | (m.name =
c.name))){
                          thisModule.inputmodel <- diag;

                          -- Verifies if there are any transitions to originate
task service (controller)
                          if (thisModule.inputmodel.ownedElement -> select(oe
|oe.oclIsTypeOf(MM!StateMachine))->first().ownedElement.first().ownedElement->
select(s | s.oclIsTypeOf(MM!Transition))-> select(t | t.source<>t.target)-
>size()>1){
                                --A task service (controller) is created
                                thisModule.controller <- MM1!Class.newInstance();
                                thisModule.controller.name <- diag.name +
'_Controller';
                                o.packagedElement <-thisModule.controller;
                          }

                          -- Verifies if there is any state transition that
participates of orchestrations in order to create the Process services.
                          -- Orchestrated transitions are marked with the
stereotype "Orchestration" and the name of the orchestration is assigned to the
parameter "servicename".
                          for(t in thisModule.inputmodel.ownedElement -> select(oe
|oe.oclIsTypeOf(MM!StateMachine))->first().ownedElement.first().ownedElement->
select(s | s.oclIsTypeOf(MM!Transition))-> select( h |
h.hasStereotype('Profile::Orchestration'))){

                                --Retrieves the parameter "serviceName" to assign
the name of the Process service and the name of the operation that handles the
orchestration.
                                for (n in
t.getValue(t.getAppliedStereotype('Profile::Orchestration'),'serviceName')){

                                -- Verifies if the process service that handles
the orchestration defined in "servicename" parameter already exists.
                                thisModule.processService <- o.packagedElement ->
select(s | (s.name = (n + '_Process'))).first();

                                if (thisModule.processService.oclIsUndefined()) {
                                      -- Creates the process service
                                            thisModule.processService <-
MM1!Class.newInstance();
                                            thisModule.processService.name <- n
+ '_Process';

                                            --Creates the operation that will
handle the orchestration
                                            thisModule.opentry <-
MM1!Operation.newInstance();
                                            thisModule.opentry.name <- n;


      thisModule.processService.ownedOperation <- thisModule.opentry;
                                            o.packagedElement <-
thisModule.processService ;


                                }
                                else {
```

```
                                        -- The Process service already exists. Gets
the operation that that handles the orchestration.
                                        thisModule.opentry <-
thisModule.processService.ownedOperation.first();

                                        }

                                -- Adds the parameters to the operation that
handles with the orchestration.
                                -- Parameters are extracted from the orchestrated
transitions if they were expressed in the model.

                                if(not t.effect.oclIsUndefined()){
                                        for (p in t.effect.ownedElement){
                                                if
(thisModule.opentry.ownedParameter -> select(s | s.name = p.name)->
first().oclIsUndefined()){
                                                        thisModule.param <-
MM1!Parameter.newInstance();
                                                        thisModule.param.name <-
p.name;
                                                thisModule.opentry.ownedParameter <-
thisModule.param;
                                                }
                                                }
                                }
                        }

                }
        }

                -- Assign the Entity service class to the new service diagram and
calls the rule to format the Entity service and the Task service created above.
                        o.packagedElement <- thisModule.formatEntityTaskServices(c);

                }
        }

}


lazy rule formatEntityTaskServices {
        -- This rule creates preformatted CRUD operations in the Entity services
and analyzes each transition of the state machine diagram,
        -- in order to identify if the operation will be assigned to the Entity
service or Task service. Transitions having the same source
        -- state and target states are assigned to the Entity Service and having
different sorce and target states are assigned to the Task service.

        from
                i : MM!Class

        to
                o : MM1!Class
                (
                        name <- i.name
                )

                do {
```

```
                    -- Creates the paramenters of the CRUD operations
                    for (op in thisModule.operations) {

                          -- Creates the parameters
                          if (op.toString() = 'read'){
                            thisModule.attribute <-
Sequence{'tablelist','attributes','clause'};
                     }else {
                                if (op.toString() = 'delete') {
                                      thisModule.attribute <-
Sequence{'table','clause'};
                                } else {
                                      if (op.toString() = 'update') {
                                            thisModule.attribute <-
Sequence{'table','attributes','value','clause'};
                                      } else {
                                            thisModule.attribute <-
Sequence{'table','attributes','value'};
                                }
                          }
                     }

                          -- Creates the CRUD operation and assigns it to the
Entity service
                          thisModule.opentry <- MM1!Operation.newInstance();
                          thisModule.opentry.name <- op;

                          for (p in thisModule.attribute) {
                                thisModule.param <- MM1!Parameter.newInstance();
                              thisModule.param.name <- p;
                              thisModule.opentry.ownedParameter <-
thisModule.param;
                          }
                          o.ownedOperation <- thisModule.opentry;

                    }


                          if (not thisModule.inputmodel.oclIsUndefined()){
                   -- Analyzes the state transitions (compares the transition
source state and target states) in order
                          -- to identify to which service the operation will be
added.

                                -- Gets all the state transitions

                                for(t in thisModule.inputmodel.ownedElement ->
select(oe |oe.oclIsTypeOf(MM!StateMachine))-
>first().ownedElement.first().ownedElement-> select(s |
s.oclIsTypeOf(MM!Transition))){

                                -- gets all services operations
                                thisModule.methods <-o.ownedElement;
                                if(not thisModule.controller.oclIsUndefined()){
                                      thisModule.methods<- o.ownedElement-
>union(thisModule.controller.ownedElement);
                                }
```

```
                                -- Verifies if the operation was not already
created in the service (some operations can appear in different transtions)
                                if (thisModule.methods -> select(s | (s.name =
t.name)).first().oclIsUndefined()){

                                      thisModule.opentry <-
MM1!Operation.newInstance();

                                --Verifies if the parameters where expressed in
the state machine model and adds them to the service operation.
                                if (not t.effect.oclIsUndefined()) {
                                  thisModule.opentry.name <- t.effect.name;

                                  for (p in t.effect.ownedElement){
                                      thisModule.param <-
MM1!Parameter.newInstance();
                                      thisModule.param.name <- p.name;
                                    thisModule.opentry.ownedParameter <-
thisModule.param;
                                  }

                                } else {
                                      -- If the action transition was not
modelled, the operation will have the transition name.
                                      thisModule.opentry.name <- t.name;
                                             }


                       if (t.source = t.target) {
                          -- Assigns the operation to the Entity service, because
source state is equal to the target state
                                      o.ownedOperation <- thisModule.opentry;
                                      } else {
                                             -- Assigns the operation to the Task
service if it does not already exists in this service
                                             thisModule.controller.ownedOperation
<- thisModule.opentry;
                                      }

                                      }
                                }
                          }

                   }

}
```