POZNAŃ UNIVERSITY OF ECONOMICS, POLAND
MACQUARIE UNIVERSITY, SYDNEY, AUSTRALIA

Monika Kaczmarek

SELECTION OF SEMANTIC WEB SERVICES FOR BUSINESS PROCESSES

PhD Dissertation

Supervisor:
Prof. Witold Abramowicz,
Poznan University of Economics, Poland

Cotutelle Supervisor:
Prof. Leszek Maciaszek,
Macquarie University, Sydney, Australia

Faculty of Informatics and Electronic Economy
Department of Information Systems

Poznań 2009

Dissertation outline

In the changing world of business, the competitiveness of companies depends heavily on their capability of quickly meeting clients' requirements and adapting to changing business environment. Considering new challenges, companies outsource internal applications and processes to external service providers in order to focus on the growth of their core activities and competencies. This process leads to the creation of service ecosystems with new markets of vendors offering IT services developed following the Service Oriented Architecture (SOA) paradigm and using the Web services (WS) technology. In addition, as far as automation is envisioned, the usage of Semantic Web services (SWS) (i.e. semantically annotated Web services) is considered.

The evolution of the market of Web services is going in the direction of open marketplaces of business-to-business SWS supporting the concept of dynamic (abstract) business processes. Such processes are implemented by selecting and combining services offered on the market. The selection is based on defined requirements and it dynamically conforms to the current state of the environment.

This dissertation delves into the selection mechanisms for business processes and proposes an original integrated approach, namely a two-step business conditions aware selection mechanism of SWS. The developed model takes into account service provisioning conditions as well as expectations of both the service requesters and the service providers. It allows for automated assignment of services to a business process in accordance to business requirements. The approach is based on the idea of integrating all necessary information into one extensible upper level model expressed using the appropriate knowledge representation techniques. The information model is able to represent high-level requirements, describe relevant artefacts, decompose them into the lower levels, and to filter out infeasible solutions using the reasoning mechanism. In the final step, the informed identification of the optimal assignment of services may be performed. The assignment takes into account global constraints and uses one of the numerical optimization techniques.

This thesis tackles three main research goals, namely: evaluation and structuring of business requirements and expectations of potential actors of SWS e-marketplace; definition of an integrated information model for the specification of artefacts involved in the selection process, which can be formalized and processed automatically and finally, definition of two-step semantic-enabled selection mechanism allowing for automating assignment of services to a process according to the business requirements.

The research approach taken in this thesis follows the design science paradigm proposed by Hevner et al, 2004.

The thesis' structure reflects the Hevner's research methodology. First, the environment and knowledge base that are relevant for this thesis are presented in Part I: Background and Motivation. On this basis, the conceptual model is described in Part II: Conceptual Model. Afterwards, the validation of the conceptual model is given in Part III: Validation.

Table of contents

List of figures

List of tables

List of abbreviations

AMPL	A Mathematical Programming Language
B2B	Business to Business
B2C	Business to Consumer
BPEL4WS	Business Process Execution Language for Web Services
CORBA	Common Object Request Broker Architecture
CSF	Critical Success Factors
EU	Effective Utility
HP	Highest Probability
HTTP	Hypertext Transfer Protocol
ILP	Integer Logic Programming
IS	Information Systems
KPI	Key Performance Indicator
LTB	Larger-The-Better
MCDM	Multiple Criteria Decision Making
MMKP	Multi-dimension Multi-choice 0/1 Knapsack Problem
MILP	Mixed Integer Logic Programming
NFP	Non-Functional Property
NTB	Nominal-The-Best
OASIS	Organization for the Advancement of Structured Information Standards
OWL	Web Ontology Language
OWL-S	Web Ontology Language for Services
QoE	Quality of Execution
QoR	Quality of Result
QoS	Quality of Service
RDF	Resource Description Framework
ROI	Return On Investment
RPC	Remote Procedure Call
SaaS	Software as a Service
SAW	Simple Additive Weighting
SAWSDL	Semantically Annotated Web Services Description Language
SLA	Service Level Agreement
SLO	Service Level Objective
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOC	Service Oriented Computing
STB	Smaller-The-Better
SWS	Semantic Web Service
TCP	Trade of Enhanced Conditional Preferences
UCP	Utility Conditional Preferences
UDDI	Universal Description Discovery and Integration
UML	Unified Modelling Language

URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WfMC	Workflow Management Coalition
WS	Web Service
WSDL	Web Services Description Language
WSLA	Web Service Level Agreement
WSML	Web Services Modelling Language
WSMO	Web Services Modelling Ontology
XML	eXtensible Markup Language

1 Introduction

This chapter introduces the topic of the dissertation. First, an overview of the subject together with the motivation for writing this thesis is presented. Next, the main research questions and a short description of the general solution are discussed. In the subsequent subchapter, the research methodology is described and applications of its main guidelines and principles used in the preparation of the thesis are outlined. Finally, the overview of the thesis' structure is given.

1.1 Motivation

The ever-growing competition on global markets forces companies to face changing market conditions and threats, new regulations that require compliance as well as pressures to innovate, deliver more value to customers, decrease costs and shorten product time-to-market. Considering the challenges, companies often outsource internal applications and processes to external service providers in order to focus on the growth of their core activities and competencies (Stankiewicz, 2005, Dan et al., 2008). While the number of partners that are directly or indirectly involved in the creation of goods and services increases, outsourcing is said to lead to a better exploitation of competitive advantages. Businesses can become more cost-efficient by exploiting economies of scale and improving their productivity (Dan et al., 2008).

The challenging market situation as well as the widespread adoption of outsourcing requires a change in how companies should approach the IT systems development. The development of enterprise information systems has undergone a great change in recent years. In order to respond to demands of business for flexibility and agility, traditional monolithic applications are being replaced with smaller composable units of functionality known as services (Vitvar et al., 2007). The drive is toward a development of information systems based on the paradigm of Service Oriented Architecture (SOA).

The SOA paradigm aims at encapsulating business functions and requirements, and making them available to collaborating partners as services (OASIS, 2006a). The notion of a service exists in many different domains. For instance, according to the ISO[1] 9004 standard, services are "the results generated by activities at the interface between the supplier and the customer and by supplier-internal activities to meet customer needs". This thesis is concerned with a special kind of services, namely Web services (WS). The term refers to a service provided using the Web infrastructure. WS are defined as loosely coupled, platform-independent, self-contained elements that allow dynamic processes to be built from network-connected internal and external service components.

[1] http://www.iso.org/.

From the IT perspective, the key features of Web services are that they take advantage of standard XML-based protocols, which can run over the Internet, and that the description of a Web service is distinct from its actual implementation. From a business perspective, the key feature is that Web services can be viewed as implementations of business services. Thus, on one hand, organizations can use the Web services technology to expose elements of their business processes (Galizia et al., 2007). On the other hand, easy available and simultaneously standardized in some way Web services make it possible to realize Business-to-Business (B2B) interoperability. This is achieved by combining Web services provided by different business partners according to some business process (Casati and Shan, 2001a).

A business process is a collection of related, ordered activities that produce a service or a tangible product. The implementations of a business process with computer systems are called workflows (Hollingsworth, 1995)[2]. With development of SOA, business processes are realized by invoking a variety of available Web services (Ardagna and Pernici, 2005) as depicted in Figure 1. Activities in a workflow are linked to Web services coming from organization's internal or external sources.

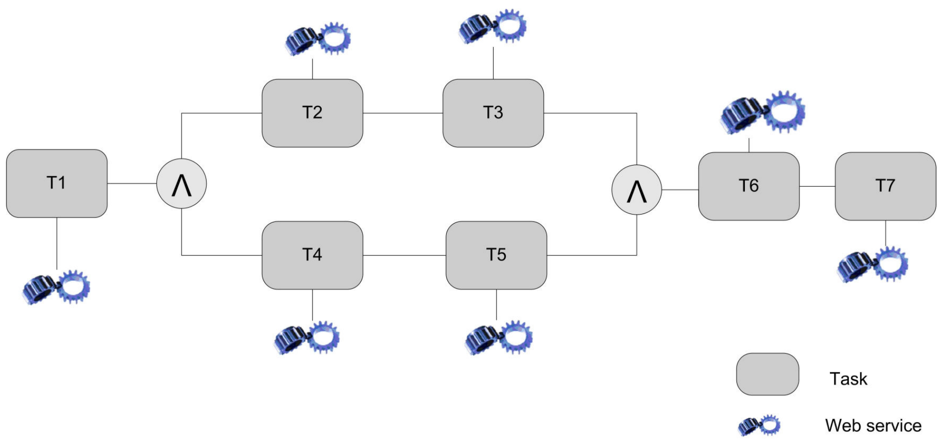


Figure 1 A business process implemented using Web services

A Web service is characterized by the offered functionality (the goal that it fulfils) as well as its non-functional properties, such as cost, execution duration, supported security level etc. Therefore, a process quality depends on the quality of WS used to implement it. The outsourcing of IT services to external vendors causes dependency and imponderability. In particular, this applies to the reliability and quality of in-sourced services. If a mission-critical activity is implemented by a service provided by an external vendor, then the provisioning of the required service quality is outside the influence of an organization. Therefore, the enterprises depend greatly on the ability to discover and select the right service for the needs of their business processes.

[2] Thus, workflows, not business processes per se, are in the focus of this dissertation.

The external services used to implement a workflow are discovered on markets of services. The services markets may take various forms, e.g., dynamic business networks (Bichler and Lin, 2006), Web services ecosystems (Barros et al., 2005), IT service parks (Petrie and Bussler, 2008) or B2B e-marketplaces (Abramowicz et al., 2008b, Papazoglou and Heuvel, 2007, Papazoglou et al., 2006, Petrie and Bussler, 2008). Within these markets, various service providers offer multiple services that customers can dynamically and on-demand bind into their business processes. Referring to (Papazoglou et al., 2006), "the visionary promise of services technologies is a world of cooperating services where application components are assembled with little effort into a network of services that can be loosely coupled to create dynamic business processes and agile applications that span organizations and computing platforms". Also (Aib et al., 2004) believe that "future IT business markets will evolve towards pervasive multi-party service interactivity [... with ...] complex multiparty service relationships between clients, servers, peers, and third parties".

The acceptance of WS technology by major industrial players[3] has lead to the situation in which enterprises represent more and more of their business functionalities as Web services (Al-Masri and Mahmoud, 2007a, Lausen and Haselwanter, 2007). Simultaneously the Software as a Service (SaaS) model of software delivery is frequently followed (Chong and Carraro, 2006). Business partners offer different service versions in order to meet varying requirements of their customers, each demonstrating different quality features and offered at varying prices (Tsesmetzis et al., 2007). Usually, a set of functionally equivalent WS can be selected, i.e. service substitutes, which implement the same functionality but differ in values of quality parameters. Therefore, service selection needs to be performed to identify the best set of services that should be used to implement a process, taking into consideration various requirements and execution contexts (Ardagna and Pernici, 2005). Once, the appropriate set of services is selected, the services are linked to the process definition.

However, neither service markets, nor business requirements are static. Service providers and their services come and go, their quality properties change in time, new services may become available offering better quality, some physical changes in the network or environment may occur, causing a need for a redesign of the process flow. In consequence, the need to replace services in the business process definition occurs. If a workflow is defined at the syntactic level and includes interface description of the participating services then, using other service during process execution requires manual changes to the workflow definition. The strong coupling between a process and an interface of the participating services, does not allow businesses dynamically changing partners and services.

[3] For instance SAP A.G. , Microsoft, IBM

As indicated in (Vitvar et al., 2007), traditional Web service technologies provide only partial solution to interoperability and quick adaptation (mainly by means of unified technological environments). Without machine-understandable semantics [4], services must be selected and bound to service requesters at the design-time, which in turn limits possibilities for automation and fast reaction to changes. In order to address these drawbacks, the extension of SOA with semantics is postulated.

Semantic extensions offer scalable integration as well as better adaptability to changes that might occur in the environment (Vitvar et al., 2007). Semantics for SOA allow the definition of semantically rich and formal service models, so called Semantic Web services (SWS), where semantic representations are used to describe both services offered and capabilities required by potential consumers of those services. In addition, the data to be exchanged between business partners can be also described in an unambiguous manner.

The ultimate goal is to develop a process by specifying component Web services through their required functional characteristics described using semantics, and only later deciding which Semantic Web service from the ones included in the registry of available services should be executed (Ardagna and Pernici, 2005) (Figure 2). Thus, business process definitions are enhanced by using semantic process templates to capture the semantic requirements of a process.

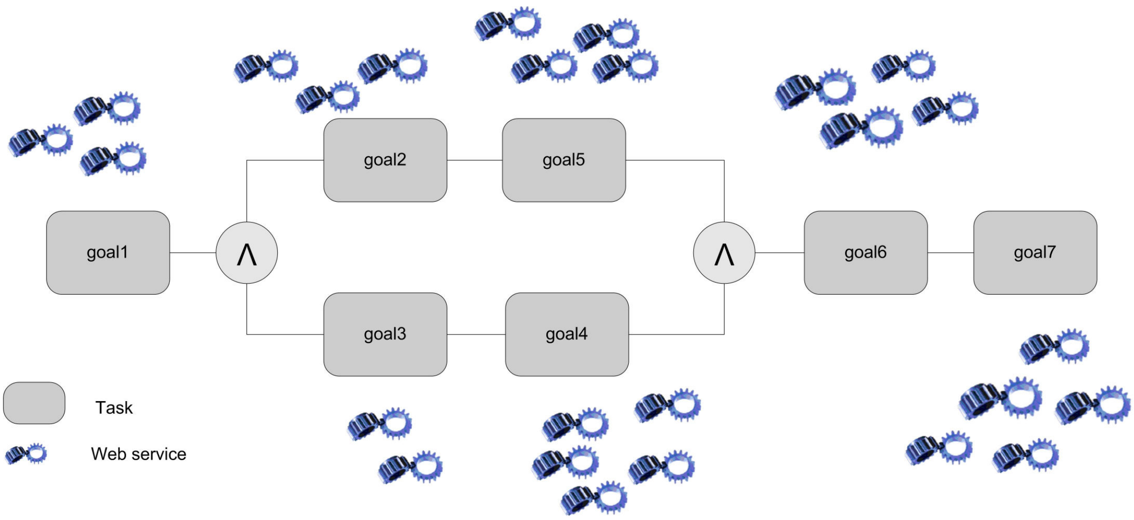


Figure 2 An abstract business process

The semantic process templates (also called abstract processes) act as configurable modules for business processes. They capture the semantics of participating activities, control flow etc. The templates are instantiated using SWS to form executable processes according to the semantics of the activities in the templates. It means that Web services

[4] According to Gartner's Emerging Trends and Technologies Roadshow, semantics is one of ten top most disruptive technologies of the future.

are not hard-coded to a certain activity but instead, thanks to the machine-understandable description, different services depending on the current condition can be selected on the fly to execute a certain activity. This leads to the SWS selection issues addressed in this dissertation.

SWS selection is an essential element of Service-Oriented Computing (Wu and Chang, 2007). As such it has been extensively studied in the past few years and has become a part of the business process optimization[5] problem. SWS selection may be defined as the process of identifying the best configuration of services to perform certain process from a number of available services offering the same or similar set of functionalities. To be more specific, the selection focuses on automatically linking each activity within a business process to SWS meeting all requirements attached either to a single activity, process fragment or an entire business process. This needs to be done in a way that ensures the maximal value of a process from a business user's point of view.

The service selection starts from an abstract business process definition and a defined set of potential SWS candidates which have been identified to be relevant to each activity by the service discovery process (Friesen, 2007). The selection is a decision for the best alternative that is the most appropriate to implement certain task or process. Being a type of decision, it requires a choice, criteria by which different choices are judged (Lamparter and Ankolekar, 2007) and the judging procedure. As selection criteria, the non-functional properties of a service are used. As underlined by (Vu et al., 2005), quality-based service selection mechanisms plays an essential role in service-oriented architectures as users want to use services that most accurately meet their requirements. The requirements can be divided into constraints defining desired values of non-functional properties of certain artefacts, and preferences indicating which non-functional property should be optimized (minimized or maximized).

The service selection process needs to be able to identify suitable services based on the requirements taking into account various optimisation criteria. The selection problem for the needs of abstract business processes is quite challenging, as the following issues need to be considered:

- potentially there may exist a large number of SWS offering the same functionality either coming from various providers or from the same providers offering different classes of their services (Yu and Molina, 2007);
- various and changing in time business users' requirements attached either to the entire process or its fragments (Aalst2003) need to be taken into account;
- multiple attributes need to be taken into account and optimized at the same time. The attributes may have different character and importance. When multiple criteria need

[5] In general, business process optimization approaches allow specification of complex applications composed by abstract services, which act as placeholders of Web service components that should be invoked during runtime. In that case, the best set of services, selected by solving an optimization problem, is invoked at runtime by implementing a dynamic/late binding mechanism.

to be considered simultaneously, an optimisation problem arises that can result in an unfeasible computational effort;

- selection should consider various and changing in time characteristics of a service and/or service providers and thus operate on reliable and up-to-date service description;
- service selection should aim at optimal assignment of services from the global point of view and not optimal assignment to each process activity (thus, workflow structures need to be considered);
- to ensure its applicability, the selection method must not be limited to specific criteria used in particular application. It must be capable of processing different processes, referring to the structure and size; use various criteria and aggregation methods.

In addition, the selection problem should focus also on business expectations and service provisioning conditions.

The evolution of the Web services market is going into the direction of open marketplaces of SWS supporting the concept of dynamic business processes. However, a robust selection mechanism that could become a part of those marketplaces and addressed at least some of the above-mentioned challenges, is currently missing. The currently existing approaches to service selection either focus too much on the technical side of the issue and neglect the business needs or quite the opposite, making the solutions not practical from the IT perspective. Most selection approaches feature a lack of representation to capture business requirements and preferences (Zhang and Li, 2004) as well as are not tailored to the needs and expectations of business users.

In this dissertation the problem of SWS selection for business process on B2B e-marketplaces is addressed, as discussed in the subsequent subchapter.

1.2 Goal of the dissertation and research questions

This dissertation focuses on the Semantic Web services selection problem as described in the previous subchapter. The dissertation proposes an integrated original approach that enables inclusion of expectations of both service requesters and providers into a selection of Semantic Web services for business processes on the B2B SWS e-marketplaces. The main thesis of this dissertation has been formulated as follows:

Thesis: *Two-step business conditions aware selection of Semantic Web services to business processes, taking into account service provisioning conditions as well as expectations of both service requesters and providers, allows for automated assignment of services to a business process in accordance with business requirements.*

To address the defined thesis, the following three research goals have been set:

Goal 1: *Evaluation and structuring of business requirements and expectations of potential actors of SWS marketplace towards the selection mechanism.*

Goal 2: *Definition of an integrated information model for the specification of artefacts involved in the selection process, which can be formalized and processed automatically.*

Goal 3: *Definition of two-step semantics enabled selection mechanism allowing for automating assignment of services to a process according to the business requirements.*

The last two goals are connected to the development of a conceptual framework for the automated business-oriented service selection for the needs of business processes.

Addressing the above research goals has required investigating the following research problems:

- establishing definitions within the problem domain of service oriented computing, service markets, SWS e-marketplace and selection mechanisms,
- analysis of requirements for SWS e-marketplace's selection mechanisms (especially from the business perspective) and evaluation of current efforts,
- development of a conceptual framework for the automated business-oriented service selection for the needs of business processes,
- definition of an integrated information model for the specification of artefacts involved in the selection process,
- definition of business-oriented selection mechanism.

Therefore, the following detailed research questions resulting from the main research goals needed to be covered:

- What are the requirements and expectations of potential actors of SWS marketplace towards both selection criteria and selection mechanism? (Goal 1)
- How should the problem of selection of SWS be modelled? (Goal 2 and Goal 3)
- How should the information model be structured for the needs of the business-oriented SWS selection? (Goal 2)
- Which knowledge representation technique should be used to express it? (Goal 2)
- How should requirements of a business user be expressed and formalized? (Goal 2)
- Which dependencies and information should be used to perform the selection and which should be discarded? (Goal 2 and Goal 3)
- How can the overall characteristics of a process or its structure in terms of non-functional properties be determined? (Goal 2 and Goal 3)
- Which optimization algorithm should be used to perform the selection? (Goal 3).

The solution approach is based on the idea of integrating all necessary information into extensible general model expressed using the appropriate knowledge representation technique. The information model is able to represent high-level requirements, describe relevant artefacts, decompose them into the lower levels, and to filter out infeasible solutions using the reasoning mechanism. In the final step, the informed identification of the optimal assignment of services may be performed. The assignment takes into account global constraints and uses one of the numerical optimization techniques.

In order to ensure the adequate research quality of the performed studies as well as to ensure that answers to the above-mentioned questions are valid (and defined research goals realized), the research conducted has been driven by the design science paradigm, as explained in the next section.

1.3 Research methodology

(Hevner et al., 2004) divide research in information systems (IS) into behavioural and design science.

The objective of the behavioural science approach is to analyze the application and effects of existing information systems onto individuals, groups, and organizations. In the focus are theories that explain the relationship between organizational aspects and the analysis, design, implementation, and management of information systems.

In contrast, the design science paradigm is oriented towards solving a problem and it originates in the engineering disciplines (Hevner et al., 2004). The objective of the design science approach is the creation and evaluation of IT artefacts, intended to solve identified organizational problems. In this context, design is defined as an intentional organization of resources and components to reach a goal.

This dissertation creates and evaluates IT artefacts that aim to solve the described business-oriented SWS selection problem. It has been developed under the design science paradigm.

As stated by (Hevner et al., 2004), the design science research addresses important unsolved problems in unique or innovative ways or solved problems in more effective and efficient way. According to (Hevner et al., 2004), artefacts that result from a design science paradigm-driven research are divided into constructs, models, methods and instantiations. Constructs provide vocabulary and symbols used to define problems and solutions. Models use the constructs to represent the problem and solution space. Methods define processes to solve a problem, i.e. algorithms and informal textual descriptions. Instantiations prove that constructs, models and methods can be realized in a working system and demonstrate their feasibility.

In this dissertation all of the mentioned artefacts have been developed. The artefacts that result from this thesis are described in Table 1.

Table 1 Artefacts resulting from this thesis

Artefacts created	**Description**
Constructs	As the general background and concept formation, Chapters 2 and 3 provide the vocabulary, principles, and terminology used in this thesis. They provide clear definitions, classifications, and discussions of established concepts and principles, i.e. service orientation, quality of service, or service selection.
Models	Based on the introduced constructs, Chapter 4 provides a high-level description of two-step selection mechanism, proposed in this thesis. It draws the general picture, defines assumptions and emphasizes the benefits of the model. Then, it proposes a framework and information model used to solve the problem of selection for the needs of enterprises.
Methods	In Chapter 4, practices for selection for the needs of business process are presented. These practices are presented informally in plain text and using activity and sequence diagrams as well as in a more formal way using the set of equations.
Instantiation	In order to verify whether the proposed models and methods are able to solve the proposed problem, Chapter 5 presents the performed evaluation of both the information model as well as the selection technique proposed, using use case scenarios and simulation setup.

According to (Hevner et al., 2004), the research framework should consist of the following elements:

- environment – it defines the problem space and is composed of people, business organizations, and their existing or planned technologies. It encompasses also goals, tasks, problems as well as opportunities that define business needs of people within organization. Those needs depend on the roles, capabilities and characteristics of people within the organization. The business needs and expectations should be evaluated and assessed. Then, they are positioned relative to existing technology infrastructure, applications, communication architectures and development capabilities. Together these mentioned elements define the research problem. As stated by (Hevner et al., 2004) framing research activities to address business needs assures research relevance;
- knowledge base – it provides the raw materials from and through which IS research is accomplished. It is composed of foundations (encompassing prior IS research and results from relevant areas and providing foundational theories, frameworks, instruments, constructs, models, method etc. that may be used during the performed research) and methodologies (providing guidelines used in the evaluation phase).

Figure 3, based on the Information Systems Research Framework by (Hevner et al., 2004), depicts the framework followed in this thesis. The framework includes the above-mentioned elements. The mentioned framework is also reflected in the organization of the conducted research and the thesis structure. First, the environment and knowledge base that are relevant for this thesis were investigated, as shown in

Part I: Background and Motivation. On this basis, the conceptual model of SWS selection has been developed and it is presented in Part II: Conceptual Model. Afterwards, the validation of the conceptual model (justify/evaluate) has been conducted and it is presented in Part III: Validation.

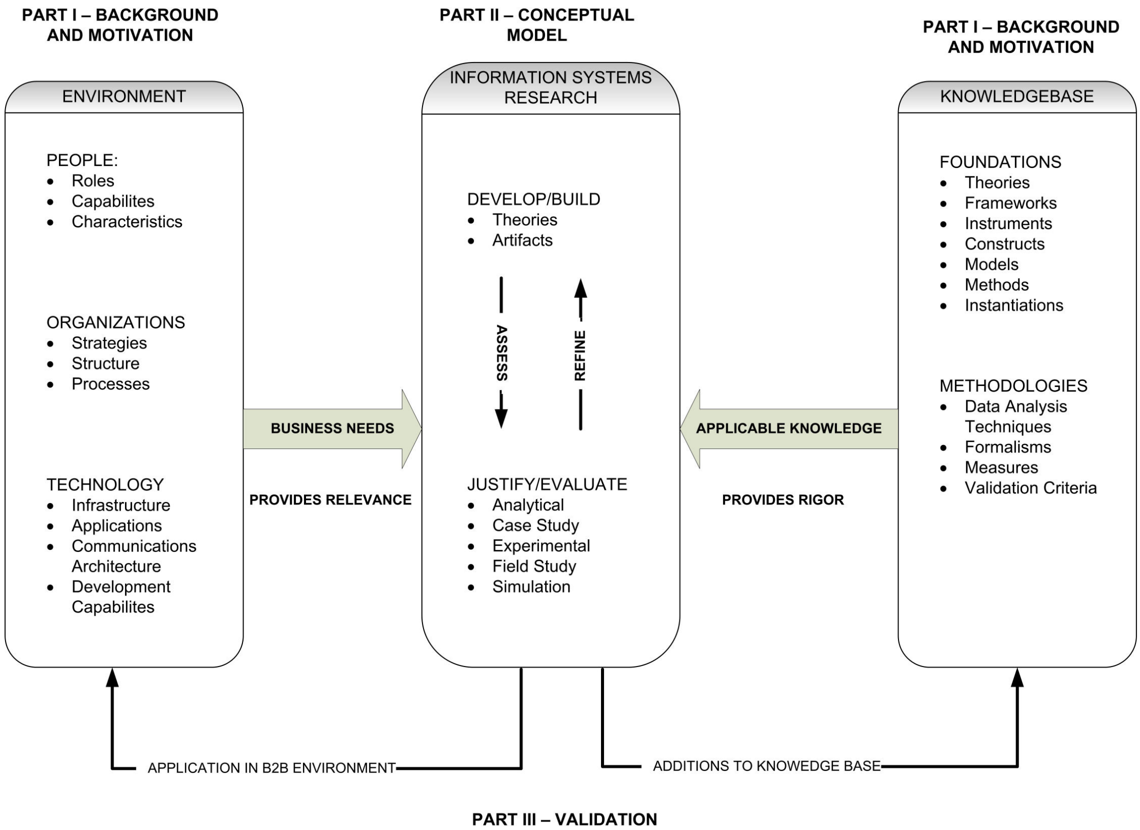


Figure 3 Information Systems Research Framework

In the centre of the design science paradigm are seven guidelines that should be adhered to by researchers while building and applying artefacts. As stressed by (Hevner et al., 2004) not all of these guidelines need to be mandatorily applied and researchers should use their own judgment to determine when, where and how to apply each of the guidelines in a given research problem. The application of the suggested guidelines within this thesis is summarized in Table 2.

The conducted research has been based on various sources. The in-depth analysis of the literature from the relevant fields has been performed. The recent research outcomes from the field of economics, computer science and information systems, presented during leading conferences (e.g. VLDB, WWW, ICWS) as well as in journal papers have been analysed. Apart from the literature and publication studies, the use cases analysis and additional surveys have been carried out in order to get to know the expectations and requirements of participants of SWS e-marketplace. In addition, the performed evaluation of the algorithms allowed to gather raw data that could be analysed in order to verify the soundness of the formulated thesis.

Table 2 Design science guidelines applied in this thesis

Guideline	Description
Guideline 1 Design as an artefact	As described above, the result of this thesis is a set of constructs, models, methods, and instantiations. Established paradigms, terms and principles are used within the several artefacts.
Guideline 2 Problem relevance	Developing a solution for selection for the needs of business processes is relevant as it represents an enabler and essential pre-requisite for a SWS e-marketplace. Numerous authors evaluate appropriate selection for the needs of composition in service-oriented computing environments as a key challenge in future IT systems. For instance, (Papazoglou et al., 2006) even state that service composition and selection are "grand challenges" and will "help shape, modern society as a whole, especially in the areas of dynamic and on-demand business".
Guideline 3 Design evaluation	The presented artefacts have been validated. However, the concepts and mechanisms underlying this thesis are rather young and are topics of on-going research. The evaluation is rather explorative and based on use case scenarios, which do not yet exist in a production environment.
Guideline 4 Research contributions	Chapter 4 presents requirements analysis for the selection mechanism and evaluates the current efforts against the identified requirements. This chapter shows that existing approaches can only partially fulfil the requirements. Thus, this thesis contributes a novel approach to the selection of SWS for the needs of business processes.
Guideline 5 Research rigor	The approach presented in this thesis is based on theoretically established and practically verified concepts and mechanisms. It uses standards and best practices as far as possible and extends them in an appropriate way. Thus, only rigorous methods are used in the construction and evaluation of the design artefacts.
Guideline 6 Design as a search process	The proposed solution has been developed iteratively. Multiple reviews that were part of the research process in the SUPER project that used parts of the approach ensure that the models and mechanisms have been revised and reached the desired quality. Nevertheless, being an abstract approach, the proposed model needs to be extended and adapted to concrete usage scenarios as appropriate.
Guideline 7 Communication of research	The artefacts that result from this thesis have been and will be communicated in different publications. For instance the description of the market model was presented during the 6th International Conference on Service Oriented Computing in Sydney, the state of the art analysis were published in Wirtschaftsinformatik Special Issue etc., various parts of the approach were published in the SUPER deliverables. The readers of these publications have been both technology-oriented as well as management-oriented. For the full list of publications see Listing 8.7.

1.4 Dissertation organization

This thesis is divided into three parts (see Figure 4) as indicated in the previous section. Every part consists of at least one chapter with a few subchapters.

Part I provides the general background and concept formation for this thesis. It consists of two chapters. First chapter in this part, Chapter 2, gives a broad background of the service orientation paradigm and an application of the paradigm in service markets. For this, it discusses and defines terms that are relevant for the context of this thesis. The concept of Web services market evolution towards the e-marketplace is discussed. In addition, it introduces the concept of SWS e-marketplace, its elements and mechanisms. This chapter also provides the necessary terminology for the remaining chapters of this thesis. Finally, Chapter 3 discusses current efforts in the area of service selection mechanisms and models.

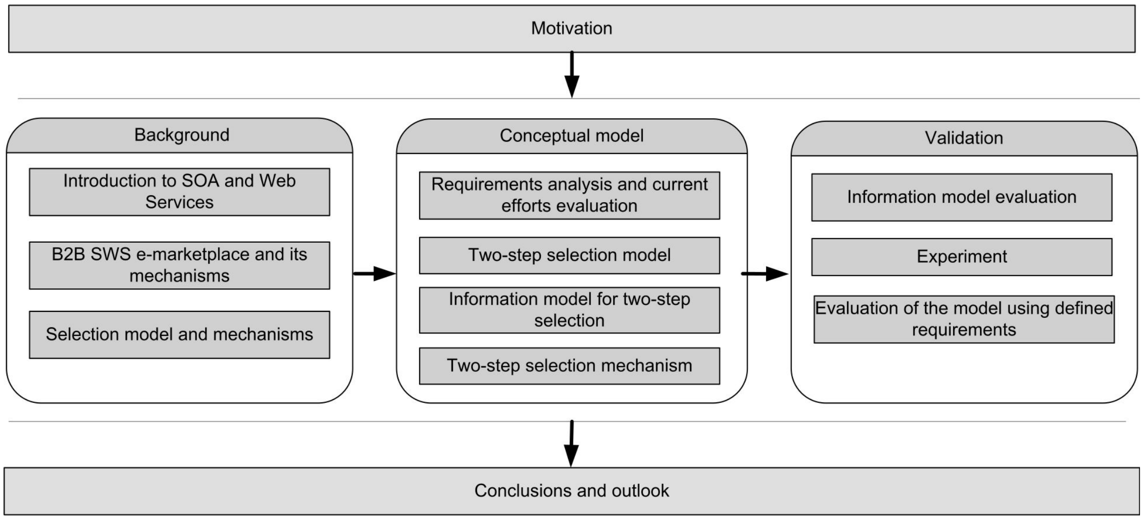


Figure 4 Organization of the dissertation

Part II presents the conceptual model developed within this thesis. It describes the proposed solution to support business users' expectations during selection process. It consists of one chapter. First, the chapter presents requirements analysis on SWS e-marketplace mechanisms (with the focus on service selection) and evaluates current efforts against the identified requirements. Then, it introduces the SWS selection approach developed within this thesis. It describes the general idea and lists assumptions of the approach, as well as presents a generic information model that integrates all necessary information for the needs of the two-step selection process. Then, the selection mechanism is presented.

Part III provides the validation of research results. It comprises of one chapter providing the validation of the developed approach regarding the information model validation and the mechanism validation.

Finally, Chapter 6 summarizes the thesis, draws major conclusions, and gives an outlook on further research topics.

PART I – BACKGROUND

In accordance with the chosen research methodology, this part presents the general background and concept formation for the thesis.

First, the service orientation paradigm together with the relevant basic definitions is presented. Then, the evolution of the service orientation is discussed. Next, the concept of the B2B Semantic Web services e-marketplace is described together with its elements and state-of-the-art mechanisms. Finally, the current efforts in the area of service selection are presented.

For reasons of limited space, only a subset of the involved topics that are particularly relevant to the context of this thesis is discussed in detail. For related topics, the respective literature in the discussed areas should be consulted.

2 Introduction to SOA and Web services technology

In this chapter, the concept of Service Oriented Architecture (SOA) along with a Web services technology often used to realize it, is discussed. This introduction is by no means complete. Its purpose is solely to provide a comprehensive overview of the service oriented paradigm, Web services technology as well as evolution of Web services market to B2B Semantic Web services marketplaces as relevant to the topic of this dissertation.

2.1 Service Oriented Architecture

Service oriented architecture (SOA) is an architectural approach for building systems consisting of independent components being services, service users and/or service providers (OASIS, 2006a, 2008). SOA may be defined as an ecosystem i.e. “a space where people, machines and services inhabit in order to further both their own objectives and the objectives of the larger community” (OASIS, 2008), where nobody is really in control of the whole system, however, each stakeholder involved has some control and influence over the community. Three key principles of SOA ecosystem are as follows (OASIS, 2008):

- SOA is a medium for exchange of value between independently acting participants;
- participants have legitimate claims to ownership of resources made available;
- behaviour and performance of the participants is subject to rules of engagement, which are captured in a series of policies and contracts.

2.1.1 SOA features

OASIS defines a service as a mechanism to enable access to one or more capabilities by using a prescribed interface, consistent with constraints and policies as specified by the service description. This definition encompasses purely business functions, business transactions composed of lower-level functions as well as system service functions (Channabasavaiah et al., 2003).

The fundamental logical view of a service within the SOA paradigm is twofold, a service interface and a service implementation (Bih, 2006). A service interface defines the identity of a service and its invocation logic whereas an implementation implements the work the service is designed to do (Papazoglou and Heuvel, 2007).

Service interfaces represent complete business functions and as such are intended to be reused, assembled into larger and new configurations, and engaged in new transactions at the level of an individual program, applications, enterprise or across enterprises (Krafzig et al., 2005). The service interface is invocable what means that at an architectural level it is irrelevant whether the communication is local (within a system) or remote (external to the system), which schema or protocol is used to realize the invocation, or which infrastructure's components are required to build the connection (Channabasavaiah et al., 2003).

In turn, a service implementation is the actual code that fulfils the functionality of the service being "the logic that is available somewhere on the network and executes when called" (OASIS, 2006a). Unlike the service interface defined in a neutral format, the service implementation is inherently platform-dependent. However, in order to utilize a service only information on an interface is required and a user does not know and does not need to know, how the requested business function is implemented (OASIS, 2006a).

Therefore, the SOA approach, based on loose coupling[6], allows enterprise architects to abstract from implementation details. An important consequence of loose coupling is that services are interoperable i.e. they can run anywhere on the network and they are not restricted to a specific hardware or software platform or programming language and services may (and, in many cases, will) originate from different technology suppliers. In addition, through encapsulation, SOA allows hiding internal data from the external access. It uses a standard service interface describing how to call the service, specifying, among others, where the service is located and the format of input/output parameters. The service interface is what provides another program with the information it needs to make a request to the service and get a response.

Therefore, the SOA paradigm allows for reduction of time to market via consolidation and reuse of services. In fact, the reusability of services (components) is the basic idea of SOA. Within the SOA paradigm, all services are independent and operate as black boxes. The only thing users need to care about is whether the service delivers exactly what is promised in its published interface (i.e. contract) (Papazoglou and Heuvel, 2007).

2.1.2 SOA actors

The SOA paradigm distinguishes a few actors (Bih, 2006, OASIS, 2008):

- service provider – the entity that implements a service and offers to carry it out on behalf of a requester;
- service registry – a place where available services are listed and which allows providers to publish/advertise their services and requesters to query for service offers;
- service requester – a potential user of a service, also called an end user or customer.

---

[6] All services/processes are organized as modular components and may be linked easily as business requirements demand, also dynamically at run-time, with few dependencies on how the services are actually implemented.

Both a service provider and a requester may be roles played by software agents[7] on behalf of their owners (He, 2003).

Figure 5 shows interactions between actors in the SOA architecture. The interactions involve *publish*, *find* and *bind* operations (Bih, 2006). These operations act upon two service artefacts: service description and service implementation. In a typical service-based scenario, a service provider hosts a network accessible software module (a service implementation). The service provider defines a service description and publishes it to a registry through which the description may be potentially discovered (OASIS, 2006a). The requester uses a find operation to retrieve the description from the registry and uses the description to bind with the provider and invoke the service or interact with a service implementation. A service contract describes what the service offers.

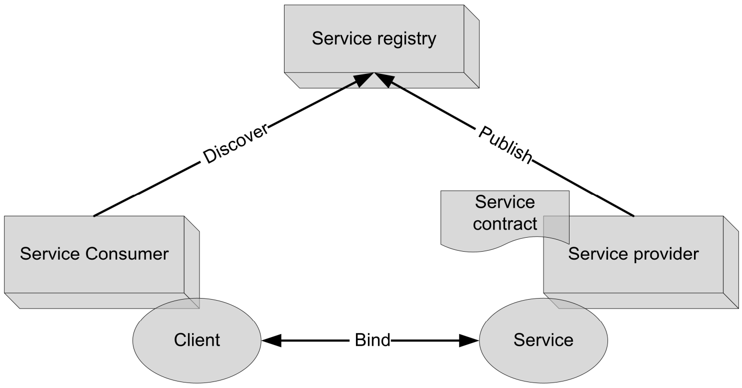


Figure 5 SOA architecture

Sometimes, the fourth actor is distinguished, namely a service broker or service aggregator (Papazoglou and Heuvel, 2007). It may be perceived as a party offering services coming from many service providers to clients (O'Sullivan et al., 2002). A service aggregator performs a dual role. First, it acts as a service provider as it offers a complete solution by creating higher-level services, which it provides to the client (Papazoglou and Heuvel, 2007). Second, it acts as a service requester as it may need to request and reserve services from other providers (Papazoglou and Heuvel, 2007). The role of service brokers is, however, something more than just offering a complete solution. Service brokers provide guidelines to requesters on which service provider and which service to select. Service brokers are trusted parties that force service providers to adhere to information practices that comply with privacy laws and regulations or in the absence of such laws, industry best practices (Papazoglou and Heuvel, 2007).

Within this dissertation, the following definitions are used:

A service provider is an entity offering a service. A service provider is responsible for technical provision and management of provided service as well as providing a proper service description.

[7] A software agent is a piece of software that acts on behalf of a user of other program.

A service requester is an entity that requests a service from a service provider (by invoking the service) and eventually consumes the service (sending request data and/or receiving the results). The term service requester is interchangeably used with terms service consumer and service client. The role of a service requester does not need to be played by a human. Also an agent (entity acting on behalf of a human) may act as service requester.

A service broker is an entity that offers services coming from many service providers to service requesters. A service broker performs dual role – it acts as a service provider offering services to service requesters and simultaneously as a service requester from the service providers' point of view. A role of a service broker may be fulfilled by an e-marketplace.

2.1.3 SOA and business processes

A business process is usually defined as "a transformation that adds value"[8]. This transformation consists of "a set of activities[9] that taken together produce a result of value to a customer" (Hammer and Champy, 2001). A business process emphasizes how the work is done within an organization and by its organization members. In turn, Business Process Management (BPM) includes methods, techniques and tools to support modelling, implementation, execution and analysis of business processes (Aalst et al., 2003). Nowadays, BPM is often combined with the SOA paradigm. While BPM specifies business directions, goals and processes that define how the organisational resources (including IT resources) are used to achieve business goals, SOA offers a flexible IT architecture that may be easily adapted to changing business requirements and helps to leverage IT investments through provision of reusable components (Papazoglou and Heuvel, 2007).

Therefore, the goal is to develop business processes by means of composition of services. A composition of services is thus considered as a workflow where software components available as services perform individual tasks of a business process. Workflow management coalition defines a workflow as the "computerized facilitation or automation of a business process". Business processes include both machine and human activities. Workflow focuses on machine activities only. Within this dissertation the focus is placed on the automated business processes, thus on workflows. However, throughout the dissertation, the terms business process and workflow are used synonymously[10].

[8] ISO 9000 standard.

[9] Within this dissertation the terms: activity and task, are used interchangeably.

[10] If both machine and human activities are taken into account, it is explicitly mentioned.

Various research attempts have been made to analyse the characteristics of a business process and workflows with regard to service compositions. The main points in favour of using services and their compositions to implement business processes are:

- technology independence and loose coupling (Huhns and Singh, 2005, Papazoglou, 2003, Yang, 2003) – the SOA-based architecture establishes a middleware that ties together functionality offered by different systems, regardless of their hardware and software. This allows using services of various providers;
- implementation neutrality (Dijkmann and Dumas, 2004, Hunhs and Singh, 2005, Yang, 2003) – services hide the structure of the software that provides the functionality behind the interface. This leads to encapsulation of underlying objects. As a consequence, changes applied to the particular implementation do not necessarily result in a change to the service interface and therefore, do not require changes when establishing the interoperation between service requester and provider;
- process orientation and flexibility (Dijkmann and Dumas, 2004) – modelling a business process leads to a description of required tasks and a specification of the execution order of the tasks. A composition of services can provide a business process in which each task is provided as a service[11.] The resulting implementation neutrality conforms to the business process paradigm. Like services, business processes should have defined input and output. In addition, the implementation of each task in the process becomes secondary. It ensures relative ease of performing changes in the process, especially replacing one service with another.

The above-mentioned aspects facilitate outsourcing and collaboration between business partners. SOA concentrates on business functions and requirements by encapsulating them and making them available to collaborating partners and this eases the outsourcing and collaboration process. The surveys conducted by Forester (Haffner et al., 2007) showed that companies of all shapes and sizes broadly recognize the strategic value of SOA for business transformation and flexibility and, as reports indicate (Haffner et al., 2007), the level of adoption of SOA to implement business processes is increasing.

2.1.4 SOA summary

To summarize, we may define the understanding of the SOA concept used throughout this dissertation as follows:

Service oriented architecture is a software architecture and modelling paradigm that enables various interactions between distributed services provided by service

[11] As was already pointed out this presumes that human interactions are not required or can be encapsulated in the form of a service.

providers and consumed by service requesters. The main activities – discovery, binding as well as invocation are enabled by the standardized protocols.

A service is a distributed component with the following characteristics (O'Brien et al., 2007): is self-contained; has a published interface that abstracts the underlying logic; is location transparent; can be implemented in different languages or platforms and still interoperate with other services; is discoverable and dynamically bound.

Service Oriented Architecture provides benefits in three basic categories: it reduces integration expenses, increases asset reuse, and improves business agility (Anand et al., 2005, Gold et al., 2004, Malloy et al., 2006). However, the SOA paradigm is not a silver bullet. It is just another approach available to achieve business objectives. The SOA architecture is not tied to any specific technology. It may be successfully implemented using a number of technologies, e.g., Remote Procedure Call (RPC), Common Object Request Broker Architecture (CORBA) (O'Brien et al., 2007). However, the most prominent technology for realizing the SOA promise of maximum service sharing, reuse and interoperability seems to be the technology of Web services (Kreger, 2003) that is discussed further in this chapter.

2.2 Web services technology

A Web services technology has captured the attention of many companies and vendors and there are plenty of various activities connected with this technology and its possible application. There exist many definitions of Web services, showing how differently various communities perceive them. Some definitions focus on technical aspects of the Web services technology while others concentrate more on its potential business applications.

For instance, World Wide Web Consortium[12] defines a Web service as "a software application identified by a URI[13], whose interfaces and bindings are capable of being defined, described and discovered by XML[14] artefacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols". Thus, a Web service has an interface described in a machine-processable format (using WSDL[15]) and allows other systems to interact with it in a prescribed

[12] World Wide Web Consortium, widely known as W3C, deals with the standardization activities in the area of broadly understood world of the Internet and its possible usage.
[13] URI – Uniform Resource Identifier.
[14] XML – eXtensible Markup Language.
[15] WSDL – Web Service Description Language.

manner (using SOAP[16] messages). Typically, the SOAP messages are conveyed using HTTP[17] with an XML serialisation in conjunction with other Web-related standards.

From another perspective a Web service may be defined as a business function made available via the Internet by a service provider, and accessible by clients that could be human users or software applications (Casati and Shan, 2001b). Or it may be also defined as a set of actions that form a coherent whole from the point of view of service providers and service requesters (W3C, 2004b). And lastly, it may be seen as a loosely coupled application using open, cross-platform standards and which interoperate across organizational and trust boundaries (Tsur et al., 2001).

Therefore, Web services are easy to implement (the first definition), they make the business functions accessible (the second one) and finally allow for easy creation of service compositions i.e. business processes (the third and the fourth one) as described in previous sections. Whichever definition of the Web services technology will be followed, a Web service is a computational entity (a service interface in term of the SOA paradigm) which is able to achieve users' goals by invocation and allows flexible and dynamic software integration that is often referred to as the *Find-Bind-Execute* paradigm (Lamparter and Agarwal, 2005). In contrast, a service (to which a Web service provides an access) is the actual value provided by this invocation (Bellwood et al., 2002, Preist, 2004).

Throughout this thesis, the following definition of a Web service is used:

A Web service is a self-contained and platform-independent interface that can be described and published by a service provider, and discovered and invoked by a service requester. This interface provides an access to certain functionality (e.g. business functionality) or some computational units like computational resources, storage resources, networks, program or databases. A Web service takes advantage of standard protocols for data processing, transmission and description of its capabilities.

The Web services technology is very popular and gained much support from the industry and business (Kreger, 2003). There are a few reasons for the Web services technology popularity to implement SOA architectures. First, existing Web services protocols and technologies take advantage of XML for data representation. Therefore, they are interoperable and using XML eliminates any network, operating system or platform binding that usage of other protocols requires (Nandigam et al., 2005). In consequence, Web services, as required by SOA, are platform-independent. In addition, using standard Internet application level protocols as means of transport for messages (mainly HTTP), Web services can pass firewalls without many problems. The

[16] SOAP – Simple Object Access Protocol.
[17] HTTP – Hypertext Transfer Protocol.

standardised and lightweight Web services protocols allow for simple and quick system integration across heterogeneous platforms.

The real power of SOA paradigm lies in creating composite applications (also called Web processes (Verma et al., 2005b)) and reconfiguring them on demand. As indicated previously, a Web service technology allows for easy creation of service compositions, i.e. composite applications that may implement a given business process designed according to the SOA paradigm. A process built using Web services remains loosely coupled, so that unplugging a component and replacing it with another does not disrupt the long running interaction. The concept of a loose coupling refers also to the fact, that an end user is not tied to a service provider directly what eases the integration between different systems and makes the system itself more manageable (Medjahed et al., 2003a). The goal is to develop applications by specifying Web services in a process only through their required functional characteristics, and only later deciding which Web service from the ones included in a registry should be executed (Ardagna and Pernici, 2005).

2.2.1 Web services interactions

Interactions between Web services typically involve, as within the SOA paradigm, three or more parties: a requester, one or more providers and a registry (Paolucci et al., 2003a).

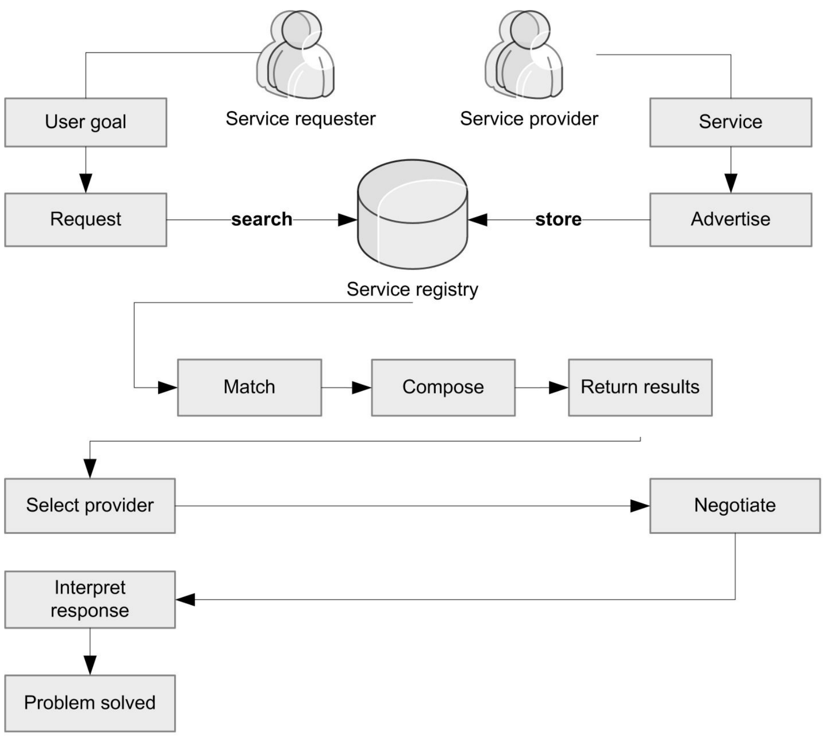


Figure 6 Web services interactions

The following interactions between actors or Web services may be distinguished (see Figure 6):

- publication – i.e. making a Web service (description) available. A provider advertises services in the registry to make them known to requesters (O'Sullivan et al., 2002);
- discovery and filtering – i.e. locating Web services that fulfil user needs. This process consists of at least three main stages: the requester has to compile a request for a service and send it to the registry; the registry has to match the request with advertisements of Web services it stores, the requester selects the provider that more closely fits his needs (Paolucci et al., 2003a);
- composition – i.e. creation of business processes out of atomic and composite services (see Figure 7) in case there is no single Web service identified by a registry to fulfil user needs. Atomic services (also called simple or basic) are Internet-based applications that do not rely on other services to fulfil consumer requests. They are pre-existing services whose instance execution is entirely under the responsibility of the service provider (Benatallah et al., 2003). A composite service may be defined as an ordered set of outsourced services (atomic as well as composite) working in tandem to offer a value-added service (Medjahed et al., 2003b). Such a composite service is in fact a workflow fulfilling a goal of a user. The composite services may be used as basic services in further compositions or may be utilized directly by end clients;

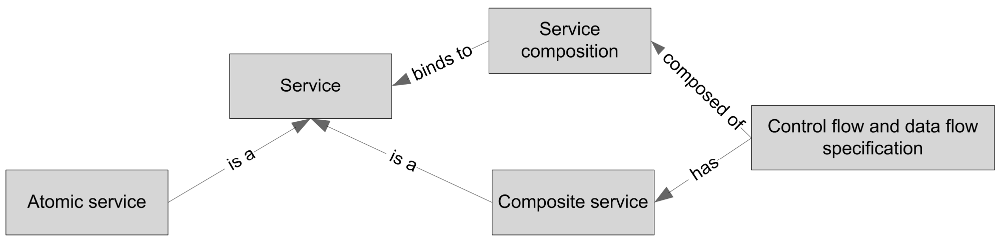


Figure 7 Service types

- selection – i.e. identifying relevant services from the set of discovered services. The result of service discovery or a composition may be a list of potential providers among which the requester has to make a selection. Services providing similar capabilities, are either called service substitutes (Abramowicz et al., 2005b) or alternative services (Medjahed et al., 2003a). There is no general rule for the selection of the provider and it is in fact domain specific decision (Paolucci et al., 2003a). In the most basic scenario, the provider with the highest score among the matches returned is selected. Another issue is the selection of services to business processes, which is a more complex task as one is not interested in selecting the best service to implement a certain task but to find the best configuration of services to implement the entire business process. This issue is discussed in more detail in Chapter 3;

- negotiation and contracting – i.e. negotiating terms of service provisioning and its characteristics. In general, this step ends with signing the contract that defines the rights and obligations of both parties;
- invocation, enactment and monitoring – invocation is the call for the execution of a service. Invocation of the contracted service usually entails service execution i.e. the actual provisioning of a service and monitoring its behaviour.

2.2.2 Semantic Web services

The advantages of SOA and Web services become apparent when various constituents are added, removed, replaced, or upgraded without adversely affecting the whole system. However, currently, programmers hardcode Web services with information about their interaction partners, messages to exchange and the interpretation of messages they receive. The result is a set of WS that cannot reconfigure dynamically to adapt to changes without direct human intervention. Such Web Services are designed to work with a definite set of providers and cannot modify their patterns of interactions when a new provider that is better or cheaper comes on-line. Similarly, they cannot react to problems of their providers: "when a Web service goes off-line, the whole supply chain is affected because the services that constitute the nodes of the chain cannot look for alternative providers" (Paolucci et al., 2003b).

Ideally, WS should act autonomously and require as minimal human intervention as possible. Web services should be able to register autonomously with infrastructure registries and in addition, should use the infrastructure registries to locate providers of services that they need. Finally, they should be able to transact with these services sending them information formatted in a way that they can understand, and be able to interpret the information that they receive as a response (McIlraith et al., 2001). Autonomous Web services not only minimize the human intervention by automating interaction with other WS, allowing programmers to concentrate on application development, but also they should be able to recover from failures more efficiently by automatically reconfiguring their interaction patterns. For example, if one of their partners is failing or it is becoming unreliable, they may be able to find other, more reliable partners. Similarly, if a new and better provider comes on line, Web services should be able to switch to work with the new provider.

Automation is of interest to providers as well as customers mainly because manual interactions are cumbersome, time-consuming and make the process error-prone. Another thing is the pursuit to gain the adaptability and flexibility required by companies.

The above-mentioned automation may be achieved taking advantage of a Semantic Web paradigm (Berners-Lee et al., 2001) and in consequence usage of Semantic Web services (Fensel et al., 2002, Nandigam et al., 2005). A Semantic Web service (SWS) may be defined as an extension of Web service description through the Semantic Web annotations, created in order to facilitate the automation of service interactions (McIlraith et al., 2001). Having such descriptions, it is possible to introduce at least semiautomatic interactions between them and reduce a human factor in service-oriented systems (Abramowicz et al., 2006b). This creates a few new possibilities in the field of service oriented computing. Throughout the thesis, the following understanding of a Semantic Web service is used:

A Semantic Web service is a semantic annotation (description) of a Web service. A semantic annotation is a basis to conduct automatic interactions with and between Web services[18].

Enhancing WS with semantics supports the vision of abstract business processes described in the motivation section, i.e. processes that are not hardcoded to specific Web service implementations but point to SWS capabilities instead. Thus, dynamic selection of Web services prior or during process execution is allowed.

The contribution of the Semantic Web paradigm in case of SWS is twofold. First, it provides ontologies that act like shared knowledge bases across the Web. Second, it provides a logic to infer how such terms may be combined to form complex concepts (Paolucci et al., 2003b).

Tom Gruber defines an ontology as a formal, explicit specification of a shared conceptualization (Gruber, 1993b). Following (Fensel et al., 2001) the conceptualization refers to an abstract model of some phenomenon in the world that identifies the phenomenon's relevant concepts. Explicit means here that the type of concepts used and the constraints on their use are explicitly defined. Formal means that the ontology should be machine understandable. Finally, shared reflects the notion that an ontology captures consensual knowledge i.e. is not restricted to some individual but is accepted by a group.

An ontology consists of a set of concepts, axioms, and relationships that describe a domain of interest. Formally, following (Maedche and Staab, 2001), an ontology consists of five elements, namely: a set of concepts (C), hierarchy of concepts (H^c), a set of relations (R), a set of non-taxonomical relations ($R \rightarrow CxC$) and a set of axioms (A^o).

From the point of view of Web services (or SWS), ontologies function as universal dictionaries so that all Web services share the same interpretation of the terms contained in the messages that they exchange. Furthermore, ontologies provide the bases for the description of capabilities of Web services that cannot be expressed using plain XML nor by any of the Web services standards (Paolucci et al., 2003b).

[18] Unless stated differently, if the term Semantic Web service is used, then a semantic description of a Web service that allows for reasoning is meant.

Therefore, the backbone of SWS constitutes an appropriate service description language that is on the one hand expressive enough to allow for precise automated interactions, but on the other hand, restricted enough to support efficient processing and reasoning (Kuester and Koenig-Ries, 2007). Several frameworks implementing this idea have been proposed. They are discussed further in the dissertation.

2.2.3 Web services description initiatives

To fully exploit the advantages of Web services (or Semantic Web services) technology their proper description, that could be used within the described interactions, is required. The common agreement is that a Web service should be represented by its surrogate describing its functional, non-functional and behavioural characteristics (O'Sullivan et al., 2002). The functional features focus on what a Web service does, the non-functional ones on how it does it and behavioural ones inform us which parties are involved in the process of service provisioning.

Currently, there are two types of service representation: syntactic and semantic ones. Reviewing through the plethora of Web service description initiatives at least two criteria should be taken into account. First of all, the scope of such a description and secondly formalism used to express it.

There are many initiatives (differing for the aspects mentioned above) in the area of service description. The earliest ones, e.g. WSDL (W3C, 2007), focused mainly on the strictly technical details needed in order to invoke a service (such as its interface, ports, bindings) and is expressed using XML. Then, other initiatives like description in the UDDI registry (UDDI, 2004) appeared. They were followed by semantic initiatives like OWL-S (W3C, 2004a), WSMO (Roman et al., 2006) or SAWSDL (Farrel and Lausen, 2006) expressed using the logic-based languages (Fensel et al., 2001) and trying to capture also information about the real-world service standing behind the WS interface.

OWL-S is an OWL-based (Antoniou and van Harmelen, 2003) Web service ontology, which supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form. This ontology is divided into three parts: Service Model, Service Profile and Service Grounding.

Another initiative, developed within a few European projects, is WSMO (Web Services Modelling Ontology) (Roman et al., 2006). The WSMO is a formal ontology (expressed using WSML (Web Service Modelling Language)[19]) for describing various aspects of services in order to enable the automation of service discovery, composition, mediation and invocation. The meta-model of WSMO defines four top-level elements: ontologies (providing the foundation for semantically describing various domains and

[19] http://www.wsmo.org/wsml/.

used by other three elements), goals (defining tasks a service requester expects a Web service to fulfil), Web service (descriptions representing the functional behaviour of an existing deployed Web service), mediators (handling data and process interoperability issues that arise when handling heterogeneous systems).

Other approach to Web service description that has gained momentum, is SAWSDL (Semantically Annotated WSDL) (Farrel and Lausen, 2006). It does not point to concrete ontology, but allows adding semantic annotations to WSDL elements. This kind of light-weighted solution seems to be quite popular, as adding semantics to WSDL is easier than creating WSMO or OWL-S descriptions from scratch.

Having a look at those initiatives, there seems to be a common agreement on how the description of the functional properties of a service should look like and its role in Web services interactions while there is still an ongoing discussion about the scope and the methods that should be used to express non-functional side of a service.

Functional properties

Functional properties represent capability of a Web service. These properties are mainly related to the input, output parameters and constraints i.e. state of the world before and after service execution (Paolucci et al., 2002). Ideally, they should express exactly what the service does and transformations of input that take place.

In most cases either the functionality is expressed as only inputs and outputs information (like in WSDL where input and output parameters are defined) or as the semantically annotated quadruple IOPE (Input, Output, Preconditions and Effects) in OWL-S or pre- and post-conditions defined within WSMO (see Table 3).

Although OWL-S and WSMO provide a framework to describe service functionality, it is by itself not enough to attach meaning to the elements of a Web Service. It is only useful if its value is linked to a domain ontology (Hess, 2006).

Table 3 Functional properties – approaches (own study)

Approach	Description
WSDL	Expressed using syntactically described input and output parameters. Additional information on the service may be derived from the textual description, if available.
OWL-S	ServiceProfile class defines the following properties of the Profile class for pointing to IOPE, namely: hasInput, hasOutput, hasPrecondition, hasEffects.
WSMO	Expressed in the form of service capability defined by pre and postconditions.
SAWSDL	Similarly to WSDL, but elements point to appropriate ontology. It supports the specification of pre and post conditions for operations.

The functional properties are used mainly during discovery and composition. The mechanisms operating on all of the above-mentioned formalisms have been already implemented and work more or less efficiently in various projects e.g. (Abramowicz et al., 2006b, Kuster et al., 2007, Liu et al., 2006) as is discussed within next sections.

Non-functional properties

The non-functional properties play a crucial role in all service interactions (O'Sullivan et al., 2002). Non-functional properties (NFP) may be defined, following (O'Sullivan et al., 2005), as anything that exhibits a constraint over the functionality offered. Non-functional properties are, therefore, distinctive criteria for the success of businesses offering their services using Web services technology as they allow differentiating between Web services offering the same (or similar) functionality.

Quality of Service (QoS) is a subset of non-functional properties. The QoS concept is difficult to define. The ISO 9004 standard offers a definition that is often considered as a starting point: “the quality is the totality of features and characteristics of a product or services that bear on its ability to satisfy stated or implied needs”. Another definition states that QoS denotes how well a service provides its functionality. Within UML notation, the QoS is described as the quantifiable non-functional characteristics of a system (OMG, 2004). Within this dissertation, QoS is understood as the extent to which the value of quality of service attributes meets the desired criteria.

Different kinds of services require different properties describing them. Which properties are necessary depends on the domain, intended use and users' requirements. Table 4 presents a few exemplary non-functional parameters.

Table 4 Exemplary non-functional parameters (Abramowicz et al., 2006e)

Parameter name	Definition
execution price	amount of money that needs to be paid to service provider for execution of a service
latency time	round-trip time between sending a request and receiving the response
average and maximum response time	the average (maximum) time needed for the packet of control data to get to the provider's server (where the service is executed) and then return to the requester
robustness	ability of the service to act properly if some of the input parameters are missing or incorrect
availability	probability whether the service is capable of processing the client's request or not at a certain time
reliability	overall measure of the service quality maintenance (needs to be defined in each case)
charging method	by execution unit, subscription or by data chunk size etc.
payment method	information on method of payment (wire transfer etc.)

The non-functional model for WS is still under development. Each of the already mentioned service descriptions like WSDL, UDDI, OWL-S or WSMO treats non-functional properties in different ways. Non-functional properties cannot be expressed using WSDL. A list of non-functional parameters provided by UDDI includes only some attributes such as e.g., provider name, service name and category. Coming towards the SWS proliferation, OWL-S and WSMO take into account wider range of NFP (than e.g. UDDI) including not only information on service providers but also some performance related information like execution time (Abramowicz et al., 2005a).

The overview of the support of the selected Web services description approaches regarding the description of non-functional aspects is presented in Table 5.

Table 5 Overview of the support of WS description approaches to NFP (own study)

Initiative	Approach to non-functional properties
WSDL	No non-functional properties are considered. However, WSOL (Web services offering language) (Tosic et al., 2002b), eQoS (Shen et al., 2004) and Extensible QoS Model (Liu et al., 2004) provide NFP extensions for use with WSDL
UDDI	Defines a set of non-functional properties of service provider (included in BusinessEntity) e.g.: address, phone number, e-mail address; and some metadata about the service, e.g., service category. UDDIe [20] and SWSQL (Bilgin and Singh, 2004) provide NFP extensions for use with UDDI
OWL-S	Includes such non-functional properties as: service name, text description, quality rating, service category; stored in the ServiceProfile class. The list may be extended using ServiceParameter from ServiceProfile.
WSMO	It recommends a set of NFP for each element of a Web service description e.g. contributor, creator, date etc. provided by Dublin Core Metadata Initiative. WSMO does not provide a model for the non-functional properties of a service (Toma, 2006) but there is an on-going work in this direction.
WSLA	WSLA (Ludwig et al., 2003) supports the specification and monitoring of NFP with the use of electronic SLA’s
O'Sullivan approach	In (O'Sullivan et al., 2005) a set of the most relevant non-functional properties for WS and their modelling are described. Exemplary concepts considered are: service provider, locative model, temporal model, service availability, obligations, price, payment, discounts, trust, security etc.

Currently, as there exists no standard for specifying NFP of a Web service (Walkerdine et al., 2007), non-functional properties are defined and represented in various ways. From this dissertation point of view, the most interesting are approaches aiming at creation of commonly accepted NFP ontologies as e.g. (Abramowicz et al., 2008a, Kim et al., 2005, Maximilien and Singh, 2004a, Maximilien and Singh, 2004b, Tian et al., 2003, Tsesmetzis et al., 2007, Zhou et al., 2004). In addition, approaches which do not provide a QoS ontology language or vocabulary specification, as e.g. (Menasce, 2002, O'Sullivan et al., 2002, Ran, 2003), present a classification of QoS parameters that can be useful when attempting to design a QoS ontology vocabulary. A detailed review of the relevant literature for QoS and NFP ontologies is, however, out of the scope of this dissertation and may be found in e.g. (Zyskowski, 2010).

[20] UDDIe project, http://www.wesc.ac.uk/projects/uddie/uddie/.

The lack of real support (languages, methodologies, tools) for NFP might be due to various factors (Eenoo et al., 2005, Rosa et al., 2002, Toma, 2006):

- NFP are usually too abstract and most of the time they are stated informally;
- in most cases there is no clear delimitation between the functional and non-functional aspects of a service;
- NFP are represented only after the functional and behaviour properties have been described;
- NFP very often conflict/compete with each other (e.g. availability and performance);
- complexity of modelling non-functional properties causing difficulties in their formalization.

Other initiatives

With the emergence and wide deployment of Web service based applications in different contexts, new requirements regarding their description arose. These include e.g., how to ensure the reliability of services, how to deal with security aspects such as authentication and authorisation as well how to describe and manage the composite services and describe their behaviour. To address all of these requirements, new specifications have been created on top of the already mentioned ones.

The most important initiative from this thesis point of view is Business Process Execution Language for Web services (BPEL4WS) (OASIS, 2006b). It provides a language for the formal specification of business processes and business interaction protocols. It extends the WS interaction model and enables it to support business transactions. BPEL4WS defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to-business spaces. A BPEL process definition specifies the technical details of a workflow that offers a composite application built from a set of atomic Web services.

Table 6 BPEL elements

BPEL element	Definition/Usage
Variables	They store process data and messages that are exchanged with WS
PartnerLinkTypes	They define the mutual required port types of message exchange by declaring which partner acts according to which role defined in a partner link
Basic activities	They specify the operations, which are performed in a process e.g. invoke, receive, reply, assign.
Structured activities	They are used to define the control flow i.e. to define concurrency of activities, alternative branches, and loops.
Handlers	They are defined to respond to the occurrence of a fault, an event or if compensation has been triggered.

The most important BPEL concepts are as follows (see Table 6): variables, partnerLinkTypes, basic activities, structured activities (e.g. alternative branches) as well as handlers (OASIS, 2006b). BPEL became a de facto standard for process execution and there are numerous platforms that support the execution of BPEL processes (Recker and Mendling, 2006).

There is also a plethora of other initiatives aiming at facilitation and improvement of WS usage process. The overview of other selected initiatives is given in Table 7.

Table 7 Other initiatives (own study)

Initiative	Description
WS-Addressing	WS-Addressing allows identifying nodes that exchange messages, independently of the protocol used to transport them, by means of endpoint references (W3C, 2006a).
WS-Policy	WS-Policy defined a general framework to describe and combine in an abstract way, different types of policies regarding the service access or execution features of services in different domains. This includes aspects such as security, reliable messaging, transactions etc. (W3C, 2006b)
WS-Security	WS-Security family of specifications (e.g. WS-Trust, WS-Federation, WS-Secure Conversation,) proposes an interoperable way to combine existing security techniques (Oasis, 2006c) .
WS-Reliability and WS-Reliable Messaging	Additional mechanism to guarantee a reliable end-to-end message exchange by specifying three basic semantics that may be combined, namely: ordered delivery, deliver of reach message at least once, and delivery of each message at most one.
WSRF	WS-Resource is a mean to express how service providers and requesters have to deal with the access to the resources they are wrapping when these resources are statefull.

2.3 SWS e-marketplace and its mechanisms

This section presents the model of the Semantic Web services e-marketplace. First, the evolution on the SOA and Web services market is discussed. Then the notion of an e-marketplace is presented. Next, a general model of the SWS e-marketplace is described along with its envisioned mechanisms. Finally, the short comparison of various initiatives in the direction of creation of open B2B SWS marketplace follows.

2.3.1 Evolution on the SOA and Web services market

The application of WS technology to implement the SOA approach has undergone evolution over time. Applications of the Web services technology started from an intra-organisational approach when companies adopted the WS technology for the needs of integration and development. The focus was assigned to decoupling of existing systems into services (Hidalga et al., 2006, Krafzig et al., 2005). Within this stage the discovery was performed manually and the composite applications (and also business processes) were hard-coded. This hindered maintenance and updating.

The second stage (Hidalga et al., 2006) that may be distinguished was the usage of Web services technology as integration drivers as they were used as a wrapping presentation layer to inter-communicate between different systems inside and outside enterprises (Krill, 2005, WebMethods, 2002). The main feature of this stage is that the registries based on the UDDI standard were used in the discovery process. The usage of the WS technology has brought about topics such as software reuse and discovery once again on the agenda of software engineers (Pan, 2005). However, there were neither many service providers nor Web services publicly available on the market. The main functionalities that were supported was management of a structured content, such as catalogues of WS, along with business information on service providers, contact information etc., and static interactions like publishing and discovery.

As UDDI showed its shortcomings in the service discovery process, other online WS registries initiatives that were to help to locate appropriate services appeared (Fan and Kambhampati, 2005) (see Table 8).

Table 8 Web services repositories and other initiatives (own study)

Type	Comment
Standard Web search engines	Web services may be discovered using standard Web search engines such as e.g. Google. e.g. the query “filetype: wsdl”, returned in a result the list of 22600 (the query tested on 21.06.2009)
Web services portals and marketplaces	The following selected portals offer an access to over 20 000 Web services: http://wsoogle.com; http://esigma.com; http://www.wsindex.org; http://seekda.com, http://www.webservicelist.com; http://www.xmethods.net; http://www.strikeiron.com/;
Portals offering company or technology specific services	Some exemplary portals offering an access to company/technology specific services: http://www.xmlrpc.com/directory/; http://developer.yahoo.com/; http://preview.xignite.com/Products/; http://solutions.amazonwebservices.com/
UDDI	Although some UDDIs are still listed in the Internet, most of them do not operate anymore: http://uddi.microsoft.com/; http://www-3.ibm.com/services/uddi/; http://www.bindingpoint.com/

As the SOA participants decided that the existing approaches do not meet their expectations (Paolucci et al., 2003b), along with the growth of the number of available Web services and their providers as well as repositories, the next phase of the evolution came along. The academic and industry research efforts have proposed many standards to formalize many aspects of WS technology, including communication, invocation, monitoring, discovery and composition of services (Bultan et al., 2003, Srivastava and Koehler, 2003, Traverso and Pistore, 2004). This stage was still offering only static interactions between services but there was more support on the client side and the registries such as UDDIs started to offer limited personalization mechanisms.

At this stage, also another change took place. At first, the emergence of the WS technology has had a major impact mainly on the way in which organizations integrate their applications, data and processes. The use of standards has reduced the cost of corporate integration projects by allowing the reuse of existing applications and benefiting from previous investments. Then, the WS adoption in the industry has allowed organizations to share information with their partners, providers and customers in a standardized manner. Finally, WS started to be also considered as a gateway to new business models (Chong and Carraro, 2006). Some big players on the software market noticed the potential of Web services and loosely coupling of their products allowing choosing only those components that are needed by the company. In consequence, Software as a Service (SaaS) model of software delivery where the software company provides maintenance, daily technical operation, and support for the software provided to their client (Chong and Carraro, 2006) appeared.

Then, the ability to efficiently and effectively share services on the Web became a critical step towards the development of the service on-line economy (Barros et al., 2005). In the pre-Web services era when a company decided to use a solution of some provider, it was also forced to use the platforms, system and technology suggested and used by the provider in question. It required some investments and when the company wanted to change the provider it was costly, as the replacement implied changing the technology and logic of business processes and that meant new investments. As a Web service may be used with any technology, the technological requirements do not longer tie the customer to the provider (Benatallah et al., 2003).

The increased interoperability between heterogeneous software components allows their reuse and composition, thus leading to the high success of the WS technology. However, a prerequisite to reusing and composing WS is the ability to find the right service(s) and thus the existence of an open service market and ecosystems[21] (Barros et al., 2005, Lamparter and Schnizler, 2006, Papazoglou and Georgakopoulos, 2003).

Starting from that point, the two of the most popular problems in WS technology addressed by both industry and academia are service discovery and composition (Fan and Kambhampati, 2005) as well as creation of an open service market.

Following (Fan and Kambhampati, 2005), at an abstract level these efforts could be classified into two main trends: one is promoted by the leading industry organizations in which the syntax of the service interfaces are specified in WSDL and the composition is done in a workflow style language such as BPEL4WS. On the other hand, the Semantic Web community argues for adding more semantics into the WS so their meaning and functionality are specified in an unambiguous and machine-interpretable way (using e.g.

[21] Web service ecosystems can generally be described as a logical collection of web services whose exposure and access are subject to constraints, which are characteristic of business service delivery (Barros et al., 2005).

OWL-S) (Fan and Kambhampati, 2005, Sirin et al., 2003). The main business area where companies started to experiment with SWS was telecommunication, retail and the government sector as they are all industries that are likely to gain substantial ROI[22] from SWS and SOA implementations (Fahringer et al., 2007). The reason is that all those industries are dealing with complex, distributed environments, and they need horizontal integration capabilities - both of which WS and SOA are primed to deliver (Gralla, 2005), and they need to quickly adapt to the changes in the environment (Fahringer et al., 2007). Moreover, the active standardization efforts in these domains (e.g. NGOSS[23] initiative in the telecommunication domain) give a promise to achieve the common agreement on the terminology (i.e. ontology) used.

To summarize, the above steps of the WS market evolution may be grouped into three main phases (Dustdar, 2005): tight coupling of intra-organizational systems; inter-organizational coupling of corporate partners (value chain) using well defined SLA; open market of services with new models of software licensing with abstract processes and dynamic service selection and instantiation. In the second as well as the third phase the SWS usage is envisioned and useful, however, not necessary.

Arguably, the market of WS is evolving into the direction of an open WS market as presented by (Papazoglou and Georgakopoulos, 2003) and depicted in Figure 8.

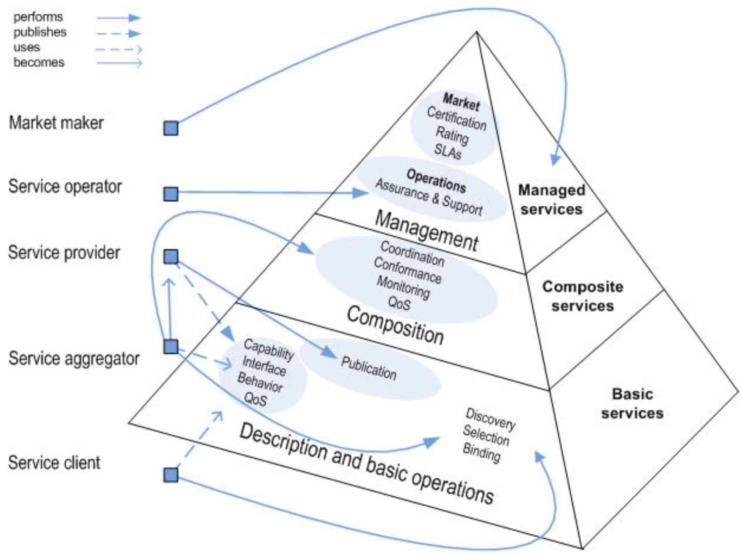


Figure 8 Extended SOA (Papazoglou and Georgakopoulos, 2003)

Three main layers in this approach may be distinguished. The first layer focuses on atomic services, their descriptions and basic operations such as: publication, discovery, selection and binding; producing or utilizing their description. This layer constitutes the SOA foundation. It has already been implemented not only for the Web services but also for the Semantic Web services (e.g. (DIP, 2007) (Kuropka and Weske, 2006)).

[22] ROI – Return on Investment.
[23] NGOSS – New Generation Operations Systems and Software – http://www.tmforum.org/tech/NGOSS.

Upper layers provide additional support required for service composition and service management. In the second layer, the composition of services requires existence of additional functionalities, namely: coordination of the composite service execution, monitoring as well as conformance that is to ensure the integrity of the composite services and finally assurance of the quality of the composite service.

In the top layer, organizations responsible for performing management functions (such as QoS assurance, overall maintenance etc.) are situated. They are known as service operators, which may be service clients or composite service creators. The aim of the third layer is also to provide support for open WS marketplaces. We are currently in the third layer and as the conducted research showed, the service-oriented computing requires an infrastructure that provides a mechanism for coordinating between service requesters and providers (Lamparter and Agarwal, 2005). Such a coordination mechanism is expected to be implemented in a form of a B2B e-marketplace.

2.3.2 B2B e-Marketplace

A marketplace allows customers and suppliers to meet (at a specific place and time) to announce their intentions regarding buying or selling. Those intentions may eventually match and be settled. Due to the evolution of communication technologies, the role of restrictions connected with the time and space on the marketplace has decreased (Lindemann and Schmidt, 1998) and the concept of electronic marketplace was defined.

An electronic marketplace is understood as an inter-organizational information system that allows the participating buyers and sellers in some market to exchange information and value (Bakos, 1997). This definition is very simple, but encompasses the essence of the marketplace activities. Other definitions stress that e-marketplaces are intermediaries allowing buyers and sellers to meet on an electronic platform that rests on the Internet infrastructure in order to: exchange information on products/services (e.g. prices, specifications), conduct transactions online and adhere to value-added services (e.g. settlement, distribution, integration) offered by the intermediary (Hadaya, 2006). Electronic marketplaces “are made up of several businesses that come together within the confines of open trusted environments to interact spontaneously and efficiently” (Tuokkola, 1998). Electronic business rules define the predictable terms of engagement and behaviour between electronic market players.

Along with the explosive growth of the Internet and e-commerce, the electronic business activity also increases. The e-marketplace activities have been evolving from the early matchmaking models to more complex interactive and interconnected marketplaces. Following (Raisch, 2001), four phases of e-marketplace evolution may be distinguished. It began with the transaction focus (the first phase) and evolved into the value–add marketplace that offers transaction support services (the second phase).

Within the third phase, the e-marketplace's services allowed for not only information exchange but also knowledge exchange facilitating cross-organizational collaboration. Functionalities such as discovery, personalization, payment, delivery, deal tracking became a must-have list of every modern Web-based marketplace (Shmueli, 2001). Finally, the ability to integrate the transaction exchange, the value-add services and the knowledge services moves the evolution of e-marketplaces into the fourth phase called Value Trust Networks.

The mentioned evolution leads to the creation of a few types of electronic marketplaces, namely (Tuokkola, 1998): virtual superstores (e.g. Amazon), electronic storefronts to existing services e.g. Internet banking, bid/ask marketplaces (any online auctions or classified marketplace), virtual order centres (e.g. Travelocity), intranet-based electronic business communities (e.g. Dell) and B2B virtual trading e-marketplaces.

In the recent years, e-marketplaces proved to be a sound solution to promote intra-organisational cooperation (Adams et al., 2001, Feldman, 2000, Grey et al., 2005) and collaboration-oriented e-marketplaces (also called B2B e-marketplaces). They are cited as an emerging approach to support online business-to-business transactions (Hidalga et al., 2006). Within this dissertation, a B2B e-marketplace is understood as follows:

B2B e-marketplace is an Internet-based platform enabling large, medium and small corporations to exchange goods, services and information in a far more efficient and effective way than on traditional marketplaces.

As indicated in (Mueller et al., 2002) the main goal of B2B e-marketplaces in their formation phase was to bring different trading partners together. However, the requirements on e-marketplaces increased. Companies demand additional features for lowering costs and for automation and optimization of their business processes (Mueller et al., 2002). Thus, the aim for e-marketplaces is to offer more automation and value add services, such as offering services for initiation, fulfilment, and completion of trading transactions including shipment, payment and logistic services.

The research in the area of e-marketplaces proved that the utilization of ontologies facilitates the processes of e-marketplace, from matchmaking, recommendation, to negotiation (Chiu et al., 2005) and helps to achieve the desired level of automation (Mueller et al., 2002). In addition, ontology allows to solve some typical problems in e-marketplaces (Chiu et al., 2005) as shown in Table 9.

Taking the above issues into account, the best form that the open service market may take is the ontology-based e-marketplace (Lamparter and Schnizler, 2006, Li et al., 2003) targeted at B2B interactions. As SWS are the good offered on the ontology-based e-marketplace, thus, instead a WS e-marketplace, a SWS e-marketplace should be considered.

Table 9 Contributions of ontology to e-marketplaces: an overview (Chiu et al., 2005)

Function	Problem	Contribution of Ontology
Matchmaking	Matchmaking is often ineffective due to rigid definitions of products of limited attributes	Shared and agreed ontology provides common, flexible, and extensible definitions of products and requirements for match-making and subsequent business processes
	It is difficult to specify complex product requirements because the relationships among attributes and values are ignored.	Complicated requirements can be decomposed into simple concepts for streamlining the elicitation of options
	User interactions are mainly limited to manual and time consuming operations.	Accessible by automated agents through Semantic Web specifications for more business opportunities
Recommendation	Recommendations are often only possible within the same category.	Ontology helps elicit alternatives for recommendation.
	Pre-set formulae for every type of product are needed for evaluation.	Ontology helps recommendation by evaluating offers in terms of flexible overall scaling
	Cross-sale and grouping of buyers and sellers with similar requests are difficult.	Matching groups of buyers and sellers as well as cross-sale is possible by using ontology-based inference.
Negotiation	No implicit ordering of alternatives.	Implicit ordering of alternatives is elicited via inheritance.
	Inefficient negotiation process and ineffective recognition due to inadequate negotiation support and manual interactions.	Machine understandable semantics facilitate negotiation and automatic configuration of products and services as specified.

2.3.3 B2B SWS e-marketplace

Web services promise users and developers greater choice of components and services (Orchard, 2002). Corporations expose their application software as WS so that other corporations can dynamically find and invoke them on B2B WS e-marketplaces (Papazoglou and Heuvel, 2007, Papazoglou et al., 2006, Petrie and Bussler, 2008). B2B SWS e-marketplace is not only to maintain an index of available service providers and their services but also add value by providing additional information about their services as well as added value functionalities (Papazoglou and Heuvel, 2007). In this dissertation, the following definition of a B2B SWS e-marketplace is used:

B2B SWS e-marketplace is an Internet-based platform enabling large, medium and small corporations to exchange information on functionalities that are made available using the Internet. Those functionalities are usually Web services annotated using semantics. In addition, the mechanisms of B2B SWS e-marketplace are *designed to allow for automation of interactions on the platform itself as well as among its actors.*

The main players on the B2B e-marketplace are service requesters and service providers i.e. business users. The SWS B2B e-marketplace is targeted at collaborating enterprises of all shapes and sizes (Noll, 2004, Papazoglou and Georgakopoulos, 2003, Papazoglou et al., 2006, Petrie and Bussler, 2008), as well as for the company internal purposes (Hepp et al., 2005, Papazoglou et al., 2006) (especially those having multiple branches in multiple locations). Service providers are organizations offering their services in the form of WS interface to other organizations/collaborating partners. Usually, they are interested in having many clients of their services. They publish their services description on the SWS e-marketplace in order to make their services available to clients and increase the chance of being discovered.

The service requesters are usually organizations that decided to focus on their core activities and use functions of other business partners. Those may be also organizations that collaborate with other parties and perform inter-organizational processes. Those organizations try to automate their business processes. Business users want to streamline or enhance their business processes by using the best available pieces of software (Abramowicz et al., 2007b, Abramowicz et al., 2005b). Business users want to find on the SWS e-marketplace services thus, also service providers that meet their requirements and may be used to perform some tasks within the process. As the processes are usually performed on the regular basis, the users are also interested in obtaining information on new services (new partners) that appear on the e-marketplace that may be potentially used instead of already used services.

The description of services is stored in the marketplace's repository. The description encompasses both the functional as well as non-functional properties. On the high level, it is not important whether the description is expressed using WSMO, OWL-S or SAWSDL etc. However, the e-marketplace's mechanisms need to ensure that all services are described using the same ontology [24] (Abramowicz et al., 2006a, Abramowicz et al., 2006b, Haniewicz et al., 2008) as well as that the domain ontologies used to describe the functionality of the service are known to e-marketplace and that it is possible to map between them.

In order to allow meaningful matchmaking, communication with the market has to take place on a semantic level. Usage of ontologies requires an introduction of reasoning functionalities to the mechanisms used by the e-marketplace. However, the mechanisms offered on the marketplace should still be tailored to both a human-user as well as machine-to-machine interactions.

[24] Otherwise ontology mediation techniques need to be used e.g., (Bruijn et al., 2006).

A few research groups focused on the creation of the ontology based open marketplaces for WS and investigated specific parts of such platforms as well as mechanisms that should be offered (Abramowicz et al., 2008b, Lamparter and Schnizler, 2006, Papazoglou and Heuvel, 2007). The identified minimal set of functionalities that the SWS e-marketplace should support is as follows:

- registering and advertising services and business processes;
- discovery of services and composite services - being the key functionality required as clients must have a tool or mechanism that helps them find a service (or their composition) according to the specification provided;
- filtering (personalization of content) – it should ensure that every customer of the marketplace feels unique by gathering the customer profiles (either the human user or acting on his behalf a software agent) and using semantics-aware algorithms to filter the incoming stream of new services in order to find the relevant ones;
- generation of reusable services and business processes (i.e. composition) - a mechanism creating abstract processes fulfilling business needs as well as allowing for efficient selection of services to the composed processes. New compositions may become again services offered on the market;
- support for negotiation and contracting between customers and sellers;
- monitoring and profiling;
- payment and other financial mechanisms.

From the ones listed above, the most important mechanisms are: discovery as well as composition (Feldman, 2000, Hidalga et al., 2006, Papazoglou and Heuvel, 2007). Composition needs to be combined with a well defined selection phase (Papazoglou and Heuvel, 2007).

There are many approaches to provide Semantic Web service systems performing various interactions e.g. (Bussler et al., 2002, Deng et al., 2004, Kuster et al., 2007, Lamparter and Agarwal, 2005, Li and Horrocks, 2003, Patil et al., 2004) that may become a framework for SWS e-marketplace. In addition, there is tremendous ongoing research effort in several EU-funded projects that deal with SWS (e.g. (Hepp et al., 2005, Kuropka and Weske, 2006, Verma et al., 2005a). Some of them provide sophisticated platforms that support and automate most of the interactions. Some of the solutions proposed by these projects were submitted to standardization bodies as industry standards while others are available as open source solutions. In order to compare the current systems a reader is referred to Table 10. It presents the most important aspects and their coverage by selected (S)WS market models.

Table 10 Comparison of selected initiatives (own study)

Criterion/Initiative	Focus	Form[25]	Targeted at/Actors	Formalism used	Supported mechanisms	NFP and SLA	Selection
ASG (Dominik Kuropka and Weske, 2008, Kuropka et al., 2008)	Adaptive service provisioning	M	End users and companies	WSML	Discovery, composition, profiling, negotiation, execution	Cost, duration, SLA are considered	+/- only selectio n on the local level is consi-dered
WSMX (Haller et al., 2005)	Execution environment for discovery, mediation, composition and mediation	F	Any entity	WSML	Discovery, composition and execution	Offers support for WSMO NFP, no SLA used	Not consi-dered
C-Cube (Canfora et al., 2005)	Corporate intranets	M	Companies	DL-based formal language	Discovery, composition and execution	Generic QoS attributes, no SLA	-
Lamparter (Lamparter and Ankolekar, 2007, Lamparter and Schnizler, 2006)	Policies, contracting. Market plays only a role of service repository	M	Business users	DOLCE ontology	Only negotiations and pricing mechanism is supported	Generic QoS attributes, SLA is considered	-
DIANE (Kuster et al., 2007)	Matchmaking - centred	F	n/a	DSD (Diane Service Descri-ptions)	Discovery, composition, and execution	Do not distinguish between functional and non-functional service properties, no SLA is considered	+/- (integral part of disco-very)
SWEET (Brambilla et al., 2007)	Creation of SWS applications	F	End users	WSML	Discovery, composition and selection	Not considered	+
METEOR-S (Verma et al., 2005a)	Discovery and composition	F	Business users	DAML-S	Discovery, composition and execution is supported	Generic QoS attributes, SLA not considered	-
Semantically-enabled Service Oriented Architecture (SESA) (Vitvar et al., 2007)	Governing principles underpinning the analysis, design and implementation of architecture, its middleware, services and processing logic	F	Various stakeholders	WSML	Discovery, mediation, selection and execution of Semantic Web services	Generic QoS attributes, SLA not considered	+

[25] M – e-marketplace, F – framework.

As may be concluded from the table, although tremendous research effort has been investigated in the area of open SWS market, until now, there is not any system that fully implements this idea.

The most interesting example and comprehensive solution, seems to be the Adaptive Services Grid project (ASG) (Kuropka and Weske, 2006). Its main aim was to develop a proof-of-concept prototype of an open development platform for adaptive and reliable matchmaking, discovery, creation, composition, enactment as well as negotiations and service profiling. Based on the semantic description of requested services by customers, ASG discovers appropriate services, develops composed services and then enacts them. ASG enables a dynamic selection of services. It leads to so called adaptive service provisioning. Instead of binding services to designed applications at the design-time, ASG proposes a more advanced and adaptive service delivery lifecycle as depicted in Figure 9. The platform is available as an open source solution.

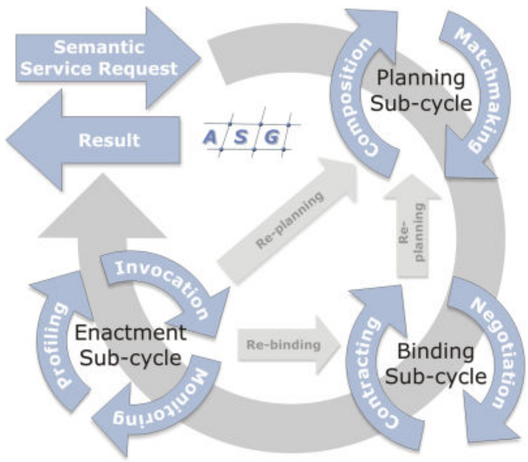


Figure 9 Adaptive Service Delivery Lifecycle (Fahringer et al., 2007)

Another interesting example is the WSMX (Web services Execution Environment) (Haller et al., 2005). WSMX is an execution environment enabling discovery, selection, mediation, invocation and interoperation of SWS. It operates on WSMO description of Web services. It was developed inter alia within DIP (DIP, 2007) and SUPER (Hepp et al., 2005) projects and is a topic of on-going research. In comparison to the ASG platform, it mainly focuses on the technical aspects of SWS and disregards almost entirely the business ones.

Another example is the Semantically-enabled Service Oriented Architecture (SESA) proposed by (Vitvar et al., 2007). The authors define both the architecture from the global perspective and then narrow it down to services, processes and technologies. Building on the Web services modelling ontology and taking into account governing principles of service orientation, semantic modelling and problem solving methods, the architecture provides support for total or practical automation of tasks such as discovery, mediation, selection and execution of Semantic Web services. In the global view, several architecture layers are identified including stakeholders, problem solving layer, service requesters as well as middleware and service providers.

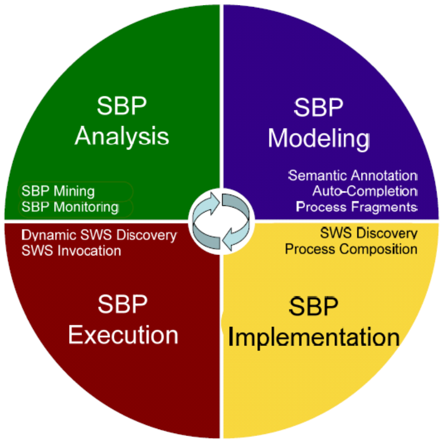


Figure 10 Semantic Business Process Management Lifecycle (Wetzstein et al., 2007)

The mentioned SUPER project (Hepp et al., 2005) defines a notion of Semantic Business Process Management (SBPM) aiming at increasing the level of automation within BPM and providing support within the process lifecycle for both business users as well as IT engineers. These aims may be fulfilled taking advantage of Semantic Web technologies (ontologies and reasoning mechanisms) and using SWS. Following (Wetzstein et al., 2007), the phases of Semantic Business Process (SBP) Management may be defined as follows: SBP Modelling, SBP Configuration, SBP Execution, and SBP Analysis as depicted in the following figure. The SUPER project aims at providing a tooling support for all of the mentioned phases.

Other systems quoted in the table focus on specific mechanisms. In addition, there are also research projects focusing only on particular types of interactions and trying to come up with an efficient solution. As the topics of discovery, filtering, composition and selection, negotiation etc. are on the must-have list of an open e-marketplace, the research conducted on these mechanisms needs to be briefly summarized.

In order to use SWS or compose new applications, first the appropriate services need to be discovered. A SWS discovery may be defined as an automated process of finding appropriate service (and in consequence service providers) for the needs of a service requester through semantic service matchmaking. The semantic matchmaking process operates on the similarity measure defined adequately to the available representation of a service and a user goal definition expressed using one of the created SWS description languages (OWL-S, WSMO etc.) and domain ontologies. In case of SWS, the similarity measure of two service attributes is the mapping measuring semantic distance between the conceptual annotations associated with the service attributes (Friesen, 2007).

There are a number of different semantic-based algorithms used to perform matchmaking between a request and an offer for each of the description initiatives already mentioned (Benatallah et al., 2005b, Haller et al., 2005, Jaeger and Tang, 2004, Lara et al., 2006, Paolucci et al., 2002, Patil et al., 2004, Sriharee, 2006, Srinivasan et al., 2004). The methods proposed for SWS discovery differ under a few aspects. From one side, they differ with respect to a kind of service and goal representations used as well as a type of matchmaking algorithm applied (e.g. subsumption reasoning, service ranking, behavioural methods or schema matching (Hao and Zhang, 2007)). They also differ when it comes to the precision, performance, level of expressiveness and flexibility as well as the complexity that the user has to deal with.

The discovery process aims at identification of one-to-one services that fulfil user requirements (ad hoc process). In case that the discovery process cannot identify a service that meets the client requirements, the composition process takes place. In general, there are four different dimensions for a SWS composition: degree of user involvement in a composition specification, whether the composition is based on templates, dynamicity (i.e., adaptation) of the composition, and again degree of user involvement in the adaptation of the composition (Arpinar et al., 2004). In the first dimension, a user can define fully a composition including its control and data-flow besides the individual services making the composite service. In contrast, a user is not involved in an automatic composition when the system itself defines control and data-flow. This is very challenging due to difficulties of mapping user needs to a collection of correlated services where their interim outputs can satisfy each other's input requirements and final deliverable meets the user demands (Arpinar et al., 2004). Besides that, in the user-defined or automatic composition techniques either actual WS instances or service templates can be used. In the latter, the individual SWS instances are searched and integrated automatically at execution time for a given plan (Chanderasekaran et al., 2003). In a dynamic composition, the composition itself can be adapted at run-time according to quality requirements. In addition, a composition may not be defined at design-time but can be assembled dynamically at execution time. Finally, some hybrid methods such as semi-automatic compositions and semi-automatic adaptations are also possible.

Within the typical process of automatic SWS composition, a composer tries to fulfil user's requirements by combining atomic services or their specifications. Usually, the composer takes the functionality of a process to be composed as an input and as outputs returns the process model describing the composite service. The process model contains a set of selected services or services goals along with the control and data flow between these services (Rao and Su, 2005).

Although the composition process is supported by a number of different initiatives for each of the SWS description languages, the achieved results still leave a space for improvement. The algorithms used to perform composition range from those using workflow techniques, to those taking advantage of the achievements from the AI area such as: situation calculus e.g. (McIlraith and Son, 2002), rules e.g. (Ponnekanti and Fox, 2002), theorem proving, HTN (Wu et al., 2003) or STRIPS based solutions (Rao et al., 2004). The algorithms used to perform SWS composition differ in terms of their precision and quality of the returned results (the correctness of the generated plan), efficiency, complexity, performance as well as the form (e.g. BPEL process description) and the scope of the returned result[26].

Services with similar and compatible functionality may be offered at different quality levels. Thus, a decision must be made to select appropriate SWS. A service selection takes place whenever more than one service with required functionality is found, both within discovery and composition process, and the automated decision needs to be made which component should be selected. As the service selection is in the focus of this thesis, a survey of the existing approaches is presented in the next chapter.

The idea that applications can be automatically composed out of reusable components seems to be very promising but to ensure the flexibility of this solution a further step i.e. negotiations of terms of use (Kretzschmar, 2007) should be considered. Negotiations describe the process of reaching an agreement between a service provider and a service requester based on economic criteria e.g. price and other non-functional parameters. Various attempts were undertaken to implement the negotiation phase for SWS, taking advantage mainly of the agent-based negotiations in e-commerce e.g., (Zhang and Qiu, 2005). The algorithms used to carry out the negotiation differ when it comes to the negotiation model and strategy they follow as well as performance based criteria.

Within the negotiation phase, the partners to perform certain transaction are selected and a contract for using the Semantic Web service is formulated. This contract takes a form of Service Level Agreement (SLA), defining the conditions and terms under which the provider and the consumers will collaborate (Estier et al., 2006). There exist a number of different SLA specification languages for WS such as e.g. WSLA (Dan et al., 2004), or WSOL (Tosic et al., 2002b) that may be applied for the needs of SWS. However, they are pure mark-up serialization languages which need specialized procedural interpreters, have a restricted expressiveness to describe complex decision and contractual logic in terms of rules and provide no capabilities to declaratively implement new functionalities (Paschke, 2006).

Another part of the SWS lifecycle that requires management and support is SWS monitoring as well as profiling. Once again, there are a few possible solutions offered by the research communities benefiting inter alia from the workflow execution and mining area (Kuropka and Weske, 2006).

---

[26] i.e. whether the generated plan is ready to be deployed on the execution engine and then executed or consists only of service specifications and service implementations still need to be added to the plan generated, the control flow is specified but the data flow still needs to be defined etc.

The conclusion from the studies of the mechanisms implemented is that in fact, each of the interactions is supported, in most cases, by a number of approaches operating on the SWS description (Kuster et al., 2007) (as summarized in Table 11). These approaches support various application scenarios and fulfil different requirements.

Table 11 Mechanisms overview (own study)

Mechanism	Status	Business aspects	Formalism used	Comments
Discovery	Many different algorithms operating on similarity function were proposed; different level of match are distinguished	Some initiatives to include QoS properties appeared	WSMO, OWL-S	A very important stage of the discovery is a service selection that has not been tailored so far to the business users needs
Composition	Various algorithms are proposed taking advantage of various techniques e.g. AI planning.	Some research has been initiated to introduce the NFP awareness into automatic composition algorithms	WSMO, OWL-S	The achieved results are still not satisfactory. An important part of composition should be taking into account business rules and constraints
Negotiation	Quite advanced multi agent based negotiations are used	The negotiation phase itself is business oriented	Various formalisms are used (e.g.WSLA, WSOL)	The result of negotiation is formalized in the form of SLA
Monitoring and profiling	Only few approaches to monitoring and profiling are proposed.	Mainly the network related quality aspects are considered	n/a	The information obtained may be formalized in the form of a service profile.

2.4 Summary

Within this chapter, basic definitions of SOA and WS technology have been provided and it has been indicated how they may be combined together to offer companies a competitive advantage.

Web services, being in fact interfaces to real services offered on the market, provide an appropriate abstraction that allows to offer and/or outsource the real services and consume them (by execution) over the Internet. Service oriented architectures enable a multitude of service providers to provide loosely coupled and interoperable services at different quality levels (Menasce et al., 2008). This provides a unique opportunity for businesses to dynamically select services that better meet their business needs in a cost-effective manner (Benatallah et al., 2003). The architecture that should provide the support for efficient discovery and application of appropriate and trustworthy services within business applications seems to be the SWS e-marketplace that is still under development, although there are a number of different initiatives in this direction.

SOA enables service compositions and facilitates reengineering of business processes, commonly defined as a structured set of activities designed to produce a special output. An important consideration in service composition is the selection of service providers in a way that meets the specific requirements of the resulting business process. The selection mechanisms are investigated within the next chapter.

3 Selection - criteria and mechanisms

Service selection is the problem of selecting the best offer made by service providers given a request. In order to select a Web service (or a Semantic Web service) from a set of similar services (i.e. service substitutes) a selection mechanism is needed.

This chapter presents the current efforts in the field of service selection. First, a service selection model is briefly discussed. Then, the overview of various aspects follows. Finally, the conclusions are given.

3.1 Selection model

As mentioned in the Motivation section, the selection is a decision for the best service or their configuration, most appropriate to implement certain task or a process. This decision requires:

- a choice - i.e. a set of services to select from, also called service candidates;
- criteria by which different choices are judged - i.e. non-functional properties and their values for each service in the set;
- a judging procedure - i.e. a set of instructions to evaluate each component and a strategy to select the most appropriate one.

Various approaches to (S)WS selection have been proposed. They differ in criteria and judging procedures applied. The existing approaches consider different number and type of non-functional properties, use various mechanisms to collect data based on which values of NFP for each of the service are computed, use different aggregation techniques to compute the value of composite service attributes based on the values of its atomic components and consider different scope of the selection etc. In addition, the selection may take place during design or runtime.

Nevertheless, on the high-level, each of the selection mechanisms includes two main elements that constitute the selection model, namely:

- information model that encompasses: information on artefacts based on which the selection is performed (i.e. selection criteria); methods used to gather the data and compute the values of selection criteria; aggregation methods utilised; a user profile and requirements taken into account;
- selection technique that encompasses: selection context (i.e. where it takes place at the design or runtime), scope of the selection and selection strategy (i.e. algorithms and mechanisms).

Within the following sections, elements of the selection mechanisms are briefly discussed.

3.2 Information model

Within the last few years of the increased (S)WS popularity, a number of different approaches to information model utilised within selection has been proposed. These approaches define different sets and understanding of non-functional properties, methods of their values computation as well as aggregation of values of service properties.

3.2.1 Non-functional properties as selection criteria

In the presence of multiple WS substitutes (i.e. services with overlapping or identical functionality), users discriminate substitutes based on NFP (see Chapter 2 for details).

Table 12 Selection criteria used by the selected approaches (own study)

Approach / Criteria	(Maximilien and Singh, 2004b)	(Liu et al., 2004)	(Day and Deters, 2004)	(Ran, 2003)	(Ludwig and Reyhani, 2007)	(Padovitz et al., 2003)	(Ardagna and Pernici, 2005)	(Galizia et al., 2007, Vu et al., 2005)	(Maamar et al., 2003)	(Zeng et al., 2004)	(Manikrao and Prabhakar, 2005)	(Shaikh et al., 2005)	(Sreenath and Singh, 2003)	(Menasce and Dubey, 2007)	(Dipanjan et al., 2005)	(Jaeger and Rojec-Goldmann, 2006)	(Lican et al., 2005)	(Agarwal and Lamparter, 2005)	(Schroepfer et al., 2007)	(Menasce et al., 2008)
Availability	-	-	+	+	+	+	+	-	-	+	-	-	-	+	+	+	-	+[27]	+	-
Reliability	-	-	+	+	+	+	-	-	-	+	-	-	-	-	+	-	-	-	+	-
Scalability	-	-	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Time-related properties[28]	-	+	+	+	+	+	+	-	+	+	+	-	-	+	+	+	-	+	+	+
Trust & reputation	+	+	-	-	+	-	+	+	-	+	-	+	+	-	-	+	-	+	+	-
Execution price	-	+	-	+	+	-	+	-	+	+	-	-	-	-	-	+	-	+	+	+
Transactions	-	+	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	+	+	-
Compensation rate	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Penalty rate	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Security	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	+	+	-
Clients' feedback	-	+	+	-	+	-	-	-	-	-	+	-	+	-	-	-	-	-	+	-
Business rules, SLA	+	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	+	-
Service load	-	-	-	-	-	+	-	-	-	-	-	-	-	-	+	-	+	-	+	-
Throughput	-	-	-	+	-	-	-	-	-	-	-	-	-	+	-	-	-	-	+	-
Location of host	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-

[27] In fact, the authors use the locative availability defined as a time frame a service is on-line, together with availability rate.
[28] For instance, execution time, response time, response latency.

As already stated in Chapter 2, there is no common QoS standard (nor NFP model) available for Web services (neither for SWS). In consequence, there is no consensus on the NFP measures that should be considered during selection, hence, various systems use different properties as depicted in Table 12.

As may be concluded from the presented table, the researchers focus mostly on generic criteria, mainly technical ones. The generic criteria are often considered as they may be applied to any service and values of these parameters can be evaluated relatively easily. The most popular ones are: availability, reliability as well as time related properties (e.g. response time).

However, with the advent of SOA and proliferation of WS in various applications, the need to consider also service objective quality and the consumer's subjective perception became apparent (Luo and Li, 2005). Therefore, various research projects extended the list of attributes and consider also e.g. trust and reputation as selection criteria.

Trust can have different meaning in different contexts. Following (Galizia et al., 2007), existing models may be classified as: policy-based approaches (policies are a set of rules that specify the conditions to disclose own resources); reputation-based (they make use of rating coming from other agents or a central engine, by heuristic evaluation) and finally trusted third party-based approaches (they use an external trusted entity to evaluate the trust level).

Other business properties (e.g. SLA or business rules) or additional information is hardly ever taken into account. For instance, only one approach has been identified that takes into account relationships such as e.g. partnerships, as part of the optimization process (Zhang and Li, 2004).

Various approaches to service selection take into account different NFP criteria but also various NFP categories are considered. Table 13 summarizes the selected NFP classification used by the selection mechanisms.

Distinguishing various categories is important as different categories of NFP usually require different sources of information as well as application of different methods in order to compute their values. For instance, if qualitative values are to be aggregated, they first need to be mapped to the appropriate numerical values (Liu et al., 2004). Depending on the versatility of the parameter's value either different sources of information are taken into account or different techniques are used to derive its value (Abramowicz et al., 2006d). When using user feedback, various techniques dealing with objectiveness of user evaluation of a service need to be applied (Vu et al., 2005). In turn, while computing the utility function (Abramowicz et al., 2005b) the nature of NFP becomes very important indicating whether the properties have positive or negative impact (Zeng et al., 2003) or whether we are interested in maximizing or minimizing its value (Yu and Lin, 2007). Finally, dependencies between properties i.e. division into independent and dependent parameters (Dumas et al., 2003) also need to be considered (e.g. the overall price of a service depends on the quality value: the bigger the quality value, the bigger the price is).

Table 13 Various NFP classifications (own study)

Category type	NFP category	Examples
Domain of the criteria (Liu et al., 2004)	Generic (encompass features that are common for every service)	Price, dependability, failure rate
	Domain-specific (also called business related)	Penalty rate
Dependency between NFP and environment (Liu et al., 2004)	Deterministic (values are known at the time that the service is invoked)	Price
	Non-deterministic (values are uncertain when the WS is being invoked)	Response time
User involvement (Maximilien and Singh, 2004a)	Objective	Availability, response time
	Subjective (parameters which are focused on the user's experience)	Clients' experience, rakings and ratings as well as the level of trust
Type of the criteria (Ran, 2003)	Run-time related (features that can be evaluated dynamically)	Execution time, latency time
	Transaction support related	Atomicity, durability
	Management and price related	Regulations, cost
	Security related	Access, privacy etc.
Provided information (Kaczmarek et al., 2008)	Cost related	Service, resource cost etc.
	Performance	Throughput, execution time
	Availability	n/a
	Reliability	n/a
	Security	Authentication, non-repudiation
	Owner	Name of service provider
Versatility (Abramowicz et al., 2006d)	Static (values of service properties that do not change over time)	Name of service
	Semi-static (values of service properties that may change (periodically) over time)	Price
	Dynamic (values of service properties that may be different in every execution of the service)	Response time, execution time
Character (Abramowicz et al., 2006d)	Qualitative	Security
	Quantitative (may be expressed as some measurable value)	Price, response time
Nature (Abramowicz et al., 2006c)	Larger the better	Reliability
	Smaller the better	Execution time, price
	Nominal the best	Security algorithm implemented

The most current selection approaches take advantage not only of the syntactically expressed NFP model but use the NFP ontology and Semantic Web technologies to perform the selection (Schroepfer et al., 2007). The developed ontology should be used to represent NFP as well as the requirements of consumers as e.g., done in (Maximilien and Singh, 2004a, Maximilien and Singh, 2004b).

Usually, the NFP ontology used during the selection contains the following key characteristics: concepts denoting non-functional parameters and relationships between parameters showing the direction and the strength of the qualities. In various approaches, these relationships are differently defined. E.g., within the framework used by (Maximilien and Singh, 2004a), the direction is either opposite, parallel, independent or unknown, whereas the strength is defined as weak, mild, strong or none.

Some researchers use less sophisticated ontologies. For instance, (Day and Deters, 2004) within their expert system performing selection, use ontology that represents only a simple hierarchy of NFP. This hierarchy is divided into two groups: criteria specific to the service itself (it includes the generic as well as domain oriented ones) and criteria specific to the request (e.g. time needed by the service to handle the request).

Some approaches also use ontologies to model a context of some specific criteria like e.g. security (Abramowicz et al., 2007a) or trust ontology (Galizia et al., 2007). However, as already mentioned in Chapter 2, for comprehensive overview and comparison of various NFP ontologies a reader is referred to e.g., (Zyskowski, 2010).

3.2.2 Ways to collect values of NFP

Various selection approaches use various sources to derive the values of selection criteria. The following sources of information used to compute or derive values of NFP may be distinguished: service provider's statement, SLA, clients' feedback, testing environment, monitoring and mining. Standard approaches are based on the prediction of services' performance from the quality advertised by providers as well as from feedback of users on the actual levels of quality delivered to them.

The simplest way to derive a value of NFP is to rely on service providers advertising it. Therefore, the most common approach to service selection depends on information published by a service provider (Liu et al., 2004). This requires users to rely on the accuracy of the values declared by providers. In general, this is not advisable as publication of NFP characteristic is neither an agreement nor an obligation to meet the published quality level. In addition, a provider may give untrue information in order to attract clients. In fact, this approach could be only applied in case of static properties of service which values cannot be computed or verified based on monitoring or testing facilities (Kowalkiewicz et al., 2005).

In order to get a service with a guaranteed quality a client can negotiate with a service provider and formalize the terms of provisioning in the form of already mentioned SLA (Kowalkiewicz et al., 2005). SLA is a special type of contract for quality requirements and guarantees. There exist many different formats of it. The most popular ones include such elements as: parties, service description (service operations, SLA parameters), obligations (Service Level Objectives (SLO) i.e. quality guarantees) and action guarantees that specify what happens if SLO is not met (Dan et al., 2004, Ludwig et al., 2003, Tosic et al., 2002a). SLA may also include the methods of how to measure the values of quality metrics (Ludwig et al., 2003).

As the values of some NFP change over time, their measurement for verification purposes is important (Abramowicz et al., 2005a). In (Ran, 2003), the author proposes a QoS model using a QoS certifier to verify published QoS criteria. It requires all providers to advertise their services with the QoS certifier. However, this approach does not take into account the dynamism of the environment and the fact that the values of a Web service change in time. The approach does not provide e.g. methods to update the NFP values automatically and it lacks the details regarding the verification process.

In (Sheth et al., 2002) the authors propose a QoS middleware infrastructure that requires a built-in tool to monitor metrics of NFP automatically. Such an approach requires the willingness of service providers to give up some of their autonomy. It may also require service providers to cover execution costs. Moreover, if the polling interval is set to too long, the QoS will not be up-to-date. If the polling interval is set to too short time, it might incur a high performance overhead. A similar approach emphasizing a service reputation, is proposed in (Maximilien and Singh, 2002 b).

Another approach obtains information from users themselves (Vu et al., 2005). When collecting quality information from the users feedback, each user is required to evaluate QoS (and at the same time QoR) of the consumed service. The main advantage of this approach is that QoS values can be computed based on the real user experience (up-to-date execution data). The main disadvantage is the fact that a user judgment is not objective, users use different definitions of quality, have different past experiences, etc.

When it comes to obtaining information from users, several approaches concerned with trust and reputation have been proposed for WS selection e.g., (Dumas et al., 2003, Kao et al., 2006, Liu et al., 2004, Maximilien and Munindar, 2004, Xu et al., 2007). In (Wang and Vassileva, 2007) those systems have been reviewed taking into account the following criteria: centralized vs. decentralized, agent vs. resource, and finally global vs. personalized. Most of these approaches depend on a central registry to collect and store feedback from consumers. The general idea of these methods is that consumers report the data acquired from executing a service (e.g. execution time, response time) and/or their ratings (overall satisfaction) of other metrics to the central registry. These ratings are further published and used to provide new customers with valuable information used to rank services. Based on the QoS information and a consumer's profile showing preferences over different QoS metrics (i.e. which metrics are important see 3.2.4), the registry calculate an overall rating for each service (Sherchan et al., 2006). Most systems assume that the customer ratings are non-malicious and fairly accurate e.g. (Xu et al., 2007), however, approaches to handle the malicious user behaviour have also been proposed (Vu et al., 2005). The most popular solutions are centralized agent-based systems (Maximilien and Singh, 2004b, Padovitz et al., 2003).

In other approaches e.g. (Casati et al., 2004, Kowalkiewicz et al., 2005) NFP values are solely collected through active monitoring. The monitoring can be performed by a user, service broker or platform, group of agents (Maximilien, 2005), a dedicated registry (Kuropka and Weske, 2006, Liu et al., 2004) or already mentioned QoS certifier (Ran, 2003). The data is collected from the actual consumption of a service and therefore is accurate and objective. One avoids the necessity to install rather expensive middleware to check constantly large number of service providers. However, there is a high overhead since quality level must be constantly checked for a large number of Web services. On the other hand, the approach that relies on a third party to rate or endorse a particular service provider is expensive and static in nature.

When the service related data collection is performed by e.g. workflow monitoring or user feedback, another important issue is how to compute the values of quality related parameters from the collected data. There are a few initiatives to solve this problem. One of them (Maximilien and Singh, 2004b) suggests performing analysis of past executions of atomic and composite services by using data mining and workflow log mining techniques. Similar approach, called service profiling, was proposed by (Abramowicz et al., 2006d). The main goal of service profiling is to create service profiles of atomic and composite services. A service profile may be defined as an up-to-date description of a selected subset of NFP of a service. It not only characterizes a service but also allows for services comparison based on aggregated values of NFP and in consequence, selecting the service most suited to the requirements of a user.

There exist selection systems that use a combination of the above-mentioned approaches to gather the information. For instance (Liu et al., 2004) present a system where QoS information can be either provided by providers, computed based on execution monitoring by the users or collected via requesters feedback and depending on the characteristics of each QoS criterion.

To summarize, it needs to be emphasized that the most important issue is to assign appropriate method to the given parameter taking into account its categorization/character. Table 14 presents the distinguished categories (see Table 13) and the possible computation method as well as source of information.

Table 14 NFP categories and possible sources of information (own study)

Category type	NFP category	Source of value
Area of application of the criteria	Generic	Service provider's statement, SLA, clients' feedback, testing environment, monitoring and mining
	Domain-specific	Service provider's statement, SLA, clients' feedback
Availability of NFP values	Deterministic	Service provider's statement, SLA
	Non-deterministic	Slients' feedback, testing environment, monitoring and mining
Source used to compute values	Objective	SLA, testing environment, monitoring and mining
	Subjective	Clients' feedback
Type of the criteria	Run-time related	Testing environment, monitoring and mining
	Transaction support related	Service provider's statement, SLA, clients' feedback,
	Management and price related	Service provider's statement, SLA, clients' feedback
	Security related	Service provider's statement, SLA, testing environment
Provided information	Cost related	Service provider's statement, SLA
	Performance	SLA, testing environment, monitoring and mining
	Availability	SLA, clients' feedback, testing environment, monitoring and mining
	Reliability	SLA, clients' feedback, testing environment, monitoring and mining
	Security	Service provider's statement, SLA, testing environment
	Owner	Service provider's statement, SLA
Versatility	Static	Service provider's statement, SLA
	Semi-static	Service provider's statement, SLA
	Dynamic	Clients' feedback, testing environment, monitoring and mining
Character	Qualitative	Service provider's statement, SLA, clients' feedback,
	Quantitative	Service provider's statement, SLA, clients' feedback, testing environment, monitoring and mining
Nature	Larger the better	In general values of those properties may be computed based on service provider's statement, SLA, clients' feedback, testing environment, monitoring and mining
	Smaller the better	
	Nominal the best	

3.2.3 NFP aggregation

When services are orchestrated in a service composition or a business process, then that service composition is again a service having certain non-functional properties. Two issues arise in this context:

- NFP aggregation: the overall quality of the composition is calculated by aggregating values of the properties of orchestrated services;

- NFP-based selection: if there are alternative services with different properties then an optimal selection considering the aggregated NFP of the process has to be found.

Thereby, NFP aggregation is a prerequisite for NFP-based service selection. Otherwise, one could not determine the result of choosing a particular service candidate. In addition, if there are requirements that a business process must meet, there has to be a method that checks whether assigned services will provide the required characteristics.

Atomic services in a composite service (i.e. a business process) may be connected using different structures. Therefore, such an aggregation must involve the structural arrangement of services in the composition. The most popular approach to NFP attributes aggregation, presented first in (Jaeger et al., 2004), is to map the structure of the composition to the fixed structural elements derived from workflow patterns (Aalst et al., 2003).

Workflow patterns describe the functional and structural characteristics of workflow management systems. In (Aalst, 2003) the author showed that they also apply to commonly known description languages for service compositions. The usually considered composition patterns are depicted in Table 15.

Table 15 Typical workflow patterns considered within service composition

Workflow pattern	Explanation
sequence of service executions	a sequence prescribes a specific order in which services have to be executed
loop	the execution of a service or a composition of services is repeated for a certain amount of times
XOR (XOR split followed by an XOR join)	in a parallel arrangement only one task is started, therefore, the synchronising operation only waits for the started task
AND (AND-split followed by an AND-join)	from a parallel arrangement all tasks are started and all tasks are required to finish for synchronisation
OR (OR-split followed by OR join)	in a parallel arrangement a subset of available tasks is started and all started tasks are required to finish for synchronisation;

In most cases, each of these patterns requires different aggregation functions to be used to different properties. Therefore, as presented in (Cardoso, 2002, Jaeger et al., 2004), first a description of a Web service composition (e.g. represented in BPEL) is transformed to a graph or sequence of composition patterns (i.e. workflow patterns presented in Table 15). Then, a recursive approach is used to aggregate properties on the level of identified composition patterns. This continues until only one task is left in the workflow and that task contains the NFP metrics corresponding to the workflow under analysis. The popular workflow patterns aggregation functions for the selected NFP are summarized in Table 16.

Table 16 Popular workflow patterns aggregation functions (own study)

Criteria	Sequence	AND	XOR	OR	Loop
Price	$\sum_{i=1}^{N} q_{price}(s_i)$	$\sum_{i=1}^{N} q_{price}(s_i)$	$\sum_{i=1}^{N} p_i q_{price}(s_i)$ or $\max q_{price}(s_i)$ is taken	$\sum_{i=1}^{N} p_i q_{price}(s_i)$ or the sum of all is taken	$k * q_{price}(s_i)$
Time-related	$\sum_{i=1}^{N} q_{dur}(s_i)$	$\sum_{i=1}^{N} q_{dur}(s^{CPA}{}_i)$ (only those being on the critical path)	$\sum_{i=1}^{N} p_i q_{dur}(s_i)$ or $\max q_{dur}(s_i)$ is taken	$\sum_{i=1}^{N} p_i q_{dur}(s_i)$ or $\max q_{dur}(s_i)$ is taken	$k * q_{dur}(s_i)$
Reliability, availability	$\prod_{i=1}^{N} (q_{rel}(s_i))$	$\prod_{i=1}^{N} (q_{rel}(s_i))$	$\prod_{i=1}^{N} (q_{rel}(s_i))$ or $\sum_{i=1}^{N} p_i * q_{reli}(s_i))$	$\prod_{i=1}^{N} (q_{rel}(s_i))$ or $\sum_{i=1}^{N} p_i * q_{rel}(s_i))$	$\prod_{i=1}^{k} (q_{rel}(s_i))$

The symbol s_i in the table denotes the i-th candidate service; q_{price}, q_{dur} and q_{rel} denote accordingly values of price, execution duration and reliability. Hence, $q_{price}(s_i)$ denotes the value of price of the i-th service. The symbol *N* denotes the total number of services within the given composition structure.

The symbol p_i in the table denotes the probability of execution of a given branch. It may be defined as a weight that may be attached to the link in the node as e.g. in (Menasce, 2002), and representing the transition probability of two subsequent tasks.

The loop structure should be defined with a maximum loop count *k,* denoting that it is not executed for more than *k* times. If the maximal number of iterations is unknown, some heuristic may be applied or it may be assumed that *k* equals 1.

In case of time-related attributes, their values are summed up. While summing up, the worst case execution time and the critical path concept are considered. The aggregation algorithm first identifies the critical path in the workflow (Zeng et al., 2003) and then, for that path, the individual values of the tasks are summed up.

In case of price-related aspects in parallel arrangements, every started task is relevant. For split cases (XOR, OR), the maximum price possible of all branches is relevant, or if already mentioned execution probability to each of the branches is assigned, then, the expected cost of the process may be counted (Abramowicz et al., 2006d).

The reliability (and availability) is understood as probability that the service is up and running. Based on the *Multiplication Law for Independent Events*, the reliability is computed as a product of reliability values of each atomic service in the composition.

Some researchers propose different techniques of aggregation. For instance (Zeng et al., 2003) while aggregating the values of reliability and availability also take into account the importance of the given task and compute the reliability value as:

$$\prod_{i=1}^{N} (q_{availability}(s_i)^{z_i}) \qquad (1)$$

where z_i denotes the importance of the task and equals 0 for tasks not belonging to a critical path and 1 otherwise.

In turn, other authors multiply reliability values only in case of a sequence or loop, in other cases the minimum value is taken (Agarwal and Lamparter, 2005).

In case of qualitative parameters, aggregation usually considers the task that is offering the weakest (lowest) value in the whole composition. Therefore, e.g. in case of security, the security level of the entire process is counted as a minimum-security level of all tasks being part of the process.

When it comes to subjective values as e.g. a user rating, usually the mean value is considered:

$$\frac{1}{N}\sum_{i=1}^{N} q_{reputation}(s_i) \qquad (2)$$

However, some authors, e.g., (Agarwal and Lamparter, 2005), postulate to take the minimum value instead (as in the case of qualitative attributes).

As indicated, a number of different aggregation functions can be defined depending on the property as well as the composition structure. Aggregated values are then used as an input to the selection techniques and become a basis for taking a decision.

3.2.4 Preferences and constraints

Selection criteria are a basis for service selection and are utilised by some algorithm the take a decision which services should be used. Taking into account user requirements allows for personalization of this decision taking process[29].

For the needs of information systems, user requirements are usually represented in a form of a user profile (sometimes also called context) (Abramowicz, 2008). A user profile is an object which structure depends strongly on its envisioned application and should be tailored to the specific needs of the system (Abramowicz, 2008).

In case of selection, one is mainly interested in non-functional requirements that apply to selection criteria considered. A non-functional requirement may be defined as

[29] The service selection does not necessarily need to be performed in order to optimize the choice from a user point of view. For instance in (Tsesmetzis et al., 2007) the authors focus on the modelling and simulation of QoS-aware Web service selection for provider profit maximization. The authors study within their work the problem of providers that receive concurrent requests of numerous customers for Web services demonstrating different bandwidth and price properties.

information that expresses a need to manage one or more non-functional characteristic e.g. a maximum value, a target or a threshold of an artefact to which the requirement is assigned. In general, user requirements may be divided into two groups (Kaczmarek et al., 2008): preferences and constraints. A constraint restricts the value of NFP, e.g. "execution duration value should be less than 5 seconds". In turn, user preferences are a collection of functions that evaluate how useful the environment is from the point of view of a user (Poladian et al., 2004). Therefore, in case of selection and several NFP, user preferences indicate the priority of optimization to be performed.

Most approaches proposed in the literature for the personalized selection concentrate on how to rank Web services according to users' preferences e.g. (Balke and Wagner, 2003, Liu et al., 2004, Zeng et al., 2004, Lican et al., 2005, Abramowicz et al., 2005b). The mentioned systems during selection apply the user-tailored optimization strategy. The optimization strategy defines whereas one is interested in the minimal or maximal value of a certain property (Casati et al., 2004) and which property is of the greatest importance (Gronmo and Jaeger, 2005). Most approaches assume that user preferences are expressed in the form of a vector of weights that may be assigned to each of the attributes. The weights are assigned a value between 0 and 1, where 1 indicates the highest importance. The user preferences expressed in the form of the weights are used within the evaluation function described in the context of local selection in the next section. It is also possible to assign a kind of ranking to the properties of interest e.g. from 1 to 100 and then normalize the assigned values.

When it comes to constraints, they may be divided into hard and soft constraints as well as quantitative and qualitative ones. The hard constraints are binding and need to be met whereas the soft constraints may be violated if there are no services that meet the defined constraint (Abramowicz et al., 2006c). The quantitative constraints restrict the value of quantitative properties e.g. time, execution cost etc. In turn, the non-quantitative ones restrict value of qualitative properties and therefore are more difficult to tackle (Vu et al., 2005). Most of the existing approaches focus on specification of quantitative constraints, disregarding the non-quantitative ones (Zeng et al., 2003). However, non-quantitative process constraints are critical for Web processes for business users (Vu et al., 2005).

The constraints may apply to various levels i.e. they may be expressed on the local level (i.e. level of a single task) or on the level of the entire composition i.e. global level. However, in case of local algorithms, as is discussed further in this chapter, global constraints are not considered during the selection process.

Both constraints and preferences constitute a user profile. A user profile may be modelled in various ways. Some approaches model a user profile using: an XML-based language to describe business constraints and preferences (Zhang and Li, 2004), policy-based languages and expressed using concepts of the ontology (Maximilien and Singh, 2004b), fuzzy if-then rules (Agarwal and Lamparter, 2005) or declarative rules (Balke and Wagner, 2003, Lamparter et al., 2007). The constraints and preferences may be also expressed using regular expressions and a context-free grammar (Kaczmarek et al., 2008) or modelled using a graphical language and taking advantage of the TCP (Trade of Enhanced Conditional Preferences[30]) used to express qualitative preferences and UCP (Utility Conditional Preferences) formalisms for the formal specification of quantitative values (Schroepfer et al., 2007).

The user profile for the needs of the selection may encompass also other information. For instance, it may also capture the information regarding the user category (e.g. business class, economy) and further personalize the content and apply privacy considerations as shown in (Dipanjan et al., 2005). The most extensive user profile has been used in (Yu and Molina, 2007) where different contexts and sub contexts influencing the decision made by the user are modelled.

As emphasised in (Zhang and Li, 2004), the business constraints and preferences constituting a user profile, tend to be informal, subjective and difficult to quantify, which poses great challenges in automating the business process selection and composition. Therefore, it is critical to formulate properly the descriptive and subjective requirements in quantifiable and objective machine-readable formats to enable dynamic business process composition and then selection. Not all of the existing approaches to service selection meet this requirement.

3.3 Selection techniques

A selection technique consists of a set of steps that should be followed. The research community proposed different selection techniques. They may be divided into several groups taking into account their nature and information they use about a service.

The first group consists of the selection techniques not relying on any information about services. It is used whenever no information about the service and its behaviour is available. In that case a random service selection is applied or the first service found is selected (Fedoruk and Deters, 2002).

The second group contains techniques that take into account only one criterion. An example may be a selection technique aiming at achieving system dependability taking into account only availability of service substitutes (i.e. if a service is not available it is not selected) or a selection technique aiming at achieving better response time, load balancing or trust-related properties as e.g. done in (Shaikh et al., 2005).

If only one criterion is relevant, then the effort is linear, a selection is trivial, and for each task the referring service candidate offering the best value is selected. However, if

[30] TCP-net (for Tradeoff-enhanced Conditional Preference-nets) model is a graphical model that encodes conditional relative importance statements, as well as the conditional preference statements

more than one service attribute need to be considered and the values of those attributes are oppositely correlated (e.g., the shorter execution duration, the more expensive service), then the evaluation function is needed. Therefore, the third group consists of selection techniques that use some kind of evaluation function (e.g., utility function) (Day and Deters, 2004) in order to determine the most appropriate service at run-time or design time. Within this dissertation, we focus on the third group of service selection techniques.

The selection approaches may use different evaluation functions, more or less personalized to the user needs, they may use different properties and take into account global or local constraints as well as consider different scope of optimization. Different types of approaches are discussed within next sections.

3.3.1 Design time versus run time optimization

The selection can be static or dynamic (i.e. take place at design or runtime). The static approach, e.g. (Jaeger and Rojec-Goldmann, 2006), requires all WS to be known at the design time of the process, whereas the dynamic (i.e. runtime) selection is more flexible as the selection is performed during the execution of a process, e.g. (Casati et al., 2000, Day and Deters, 2004, Lican et al., 2005, Ludwig and Reyhani, 2007, Manikrao and Prabhakar, 2005, Maximilien and Singh, 2004b, Padovitz et al., 2003).

The selection criteria i.e. quality values of services are subject to variability, mainly due to the high variability of the workload of Internet applications, which implies variability of service performance. The design-time service selection is criticized for being based on the statistical values of NFP attributes (Gao et al., 2006) and not considering actual values of service properties (being known only during process execution). As it uses a statistical measurement in the problem formulation (*e.g.*, the mean number of times a loop in the workflow is executed), it is not able to guarantee the strict enforcement of the agreed quality levels.

In addition, as pointed in (Zeng et al., 2003), the number of services providing a given functionality may be large and constantly changing. Consequently, approaches where the development of composite service requires the identification at design-time of the exact services to be composed may be inappropriate as the situation on the market may change before the process is executed.

Therefore, the runtime selection of component services during the execution of a composite service has been put forward as an approach to address this issue. The dynamic approach offers more flexibility and adaptability to processes that change continuously in the dynamic environment. It is based on actual values (Friesen, 2007) and thus is more reliable.

Run-time service selection requires usage of service templates within the process definition instead of service implementations. For instance, in (Schuster et al., 2000) authors propose usage of the concept of placeholder activity being an abstract activity replaced at runtime with a concrete activity type. In eFlow project (Casati et al., 2000) the definition of a service node of a process is used and it contains a search recipe represented in a query language. In turn, in (Benatallah et al., 2005a) a service community concept is used, whereas e.g. in ASG project (Fahringer et al., 2007) the composite service specification consists of a BPEL execution plan where the performed activities are not assigned to a specific WS but to SWS capabilities. Moreover, in a number of projects the notion of capabilities has been replaced with a SWS goal and included as a reference to IRI[31] within the process definition (Hoffmann et al., 2007).

In general, for the needs of the runtime service selection, the applications are expressed as processes built from abstract components. Then, the goal is to discover at run time the optimum mapping between each component and a WS, which implements the corresponding, abstract description. However, this generates an overhead during process execution what is not desirable in case of critical processes when time is a crucial factor. Therefore, optimization during the runtime needs to be efficient. In addition, the runtime selection is criticized for not being fully adjusted to business expectations as business users do want to know which service will be contracted prior to, and not during, execution of a process (Haniewicz et al., 2008).

As both approaches have advantages and disadvantages, some authors e.g. (Dipanjan et al., 2005) suggest to combine these two approaches or introduce the re-optimization step (Ardagna and Pernici, 2007, Kuropka and Weske, 2006). A re-optimization should be performed after the design time service selection only if a service invocation fails or, from a theoretical point of view, after the execution of each task since new Web services with better characteristics could become available.

3.3.2 Local and global optimization

A composite application is usually specified as a collection of service tasks combined according to a set of control-flow and data-flow dependencies[32] (Zeng et al., 2004). Different selection approaches may use different methods and techniques to describe composite applications (e.g. (Zeng et al., 2004) use state charts[33], in (Fahringer et al.,

[31] IRI – International Resource Identifier.

[32] The data perspective is not relevant for purpose of selection since the methods of allocating services to tasks only need to consider the control-flow dependencies and the available information on the component services. Data flow is of course relevant and important for execution of composite applications.

[33] Statecharts (Harel and Naamad, 1996) possess a formal semantics that is essential for analyzing composite service specifications. In addition, this is a well-known and well-supported behavior modeling notation and it offers most of the control-flow constructs found in existing process modelling languages and it is capable of expressing typical control-flow dependencies.

2007) BPEL is used). However, when it comes to the composite applications, an important aspect is not the language used to describe the composition but the scope of the performed optimization. The researchers distinguish between two basic approaches: local and global selection (Zeng et al., 2003).

Local selection is based on optimal task-level selection of services without taking into account other tasks involved in the service composition and without considering constraints spanning multiple tasks. This is because only constraints applied directly to a given task are considered. In turn, the global selection tries to find an optimal solution considering the aggregated NFP values of the composite application as constraints and preferences are assigned to a composite service as a whole and not only to individual tasks. In addition, it is possible to take into account the process structure during the optimization process.

Service selection by Local Optimization

This approach selects for each task the candidate that offers the highest score compared to other candidates. Within this approach, the selection of a Web service to execute a given task in a composite service specification is performed without taking into account other tasks involved in the composite application. This can be performed by using a greedy approach which selects the best candidate service suitable for the execution (Ardagna and Pernici, 2005). Although it is possible to take into account user preferences and constraints attached to individual tasks, it is impossible to consider a constraint which is applied to the whole composite service.

In general, in case of multiple quality attributes that should be considered, the selection system collects information on quality of each of WS that can execute the given task. Then, a quality vector for each of the candidate service is computed. Based on the quality vectors, the system selects one of the candidate services. Usually, in order to compare the quality vectors, an evaluation function is used.

An objective of an evaluation function is to identify the best service from the set of relevant services according to a user profile. Computing a simple distance function between values of non-functional properties of services, as done in (Taher et al., 2005), does not allow to take user preferences into account. Therefore, usually researchers take advantage of a Simple Additive Weighting (SAW) method (as e.g., in (Abramowicz et al., 2007b, Jaeger and Rojec-Goldmann, 2006, Vu et al., 2005)) introduced in the context of Multiple Criteria Decision Making (MCDM) (Hwang, 1981) or compute utility function[34] over the given parameters (Liu et al., 2004, Zeng et al., 2003).

In order to apply SAW, first all considered NFP attributes need to be unified (have the same character), then normalized. Finally, the weighted sum can be computed. The weights are applied to each NFP attribute and results from the users' preferences. If all

[34] Utility functions allow to assign a value to the usefulness of a system as a function of several attributes (Menasce and Dubey, 2007).

parameters are equally important to the user, the value of a score (also called the synthetic indicator (s_i) (Abramowicz et al., 2005b)) for each WS is counted as a mean of the parameters. If users express their preferences, the weighted mean is counted. The weights may be expressed as fractions which sum up to one, or may just assign values from 1 to m (m meaning the number of parameters taken into account) to the most and less important parameters. In the latter case they must be normalised to obtain the weights in the range <0;1> and summing up to one (Abramowicz et al., 2005b).

Once, the evaluation function values are known for each of the candidate services for a given task, the system chooses the service, which satisfies all user constraints for that task and which has the maximal score. If several services with maximal score exist, one of them is selected randomly. If no service satisfies the user constraints for a given task, the systems informs the user and usually propose to relax the applied constraints.

The goal of local service selection algorithm is therefore, to select individual services that meet the defined constraints and provide the best value for the user-defined utility function (expressing the preferences of a user). The most prominent approach such local selection is the initiative of (Liu et al., 2004).

An interesting element to local selection is introduced in (Xu et al., 2007). Namely, the authors introduce the concept of the random selection preventing the service with the highest score from always being selected and thus helping to balance the workload among the services that provide the same functionality and similar quality levels (i.e. one service is randomly selected whose score is greater than the user-specified threshold e.g. 0,9).

Service Selection as Global Optimization

In case of local optimization, the global quality of the composite application may be suboptimal (Zeng et al., 2004). For example, following (Jaeger and Rojec-Goldmann, 2006), let us consider the parallel arrangement of two tasks and let the optimisation goal be to form the quickest composition with the lowest cost. Using a quicker service is usually more costly meaning that cost and execution time form a trade-off couple. It may be the case that the quickest candidate for task A executes longer than any candidate for task B. The optimal assignment for task B is therefore the one that is cheapest as the execution time is not relevant in that case from the global perspective.

A selection from the local perspective is not able to discover this information, neither it can enforce constraints applied over more than one task. Therefore, the global planning approach to Web services selection that overcomes those limitations was proposed (Zeng et al., 2003).

Various algorithms were proposed for the global selection problem. As pointed in (Yu et al., 2007), the problem may be modelled in two ways as the combinatorial model or the graph model. The combinatorial model defines the problem as a multi-dimension multi-choice 0/1 knapsack problem (MMKP) (Yu et al., 2007) in case of sequential flow structure and Integer Logic Programming (ILP) in case of general flow structure e.g. (Ardagna and Pernici, 2007, Yu et al., 2007, Zeng et al., 2003, Zeng et al., 2004). The graph model in turn defines the problem as multi constraint optimal path problem (MCOP) as done e.g. in (Xiao and Boutaba, 2005).

In all models, a user-defined utility function of some parameters may be specified to optimize application-specific objectives. Following (Yu et al., 2007), in case that there are x attributes to be maximized and y attributes to be minimized, the utility function for a service s_{ij} may be defined as:

$$
\begin{aligned}
&F_{ij} = \sum_{\alpha=1}^{x} w_\alpha * \left(\frac{q_{ij}^\alpha - \mu^\alpha}{\sigma^\alpha}\right) + \sum_{\beta=1}^{y} w_\beta * \left(1 - \frac{q_{ij}^\beta - \mu^\beta}{\sigma^\beta}\right)\\
&where \qquad (3)\\
&\sum_{\alpha=1}^{x} w_\alpha + \sum_{\beta=1}^{y} w_\beta = 1
\end{aligned}
$$

And w_α and w_β are the weights for each attribute, whereas μ and σ are the average and standard deviation of the attributes values for all candidates in a service class.

Knapsack problem is about taking items into a knapsack with a limited capability, which does not allow taking all items. Thus, a selection must be performed to identify the optimal subset. In case of WS, we have various criteria e.g. (price, availability) subject to optimisation and a candidate service cannot be split to meet the constraint (Jaeger et al., 2005). The main difference of the selection of services is that a knapsack solution results in as few items as possible to keep the constraint, while for the selection a fixed number of candidates is needed. Following (Yu et al., 2007), the service selection problem is mapped to MMKP in case of sequential flow as follows:

- each service class is mapped to an object group in MMKP,
- each atomic service in a service class is mapped to an object in a group in MMPK,
- the QoS attributes of each candidate are mapped to the resources required by the object in MMPK,
- the utility a candidate produces is mapped to the profit of the object,
- a user's QoS constraints are considered as the resources available in the knapsack.

The MMKP problem has been shown to be NP-complete (Martello and Toth, 1987). The MMKP problem may be solved by finding optimal results or by using heuristic algorithms to reduce the time complexity. Most algorithms for finding optimal solutions for MMKP are based on the Branch-and-bound method. (Khan, 1998) presents such a BBLP algorithm for MMKP that finds the optimal solution by iterative generation of a search tree. The computation time for BBLP grows exponentially with the size of the problem. This may not be adequate for dynamic service selection. Heuristic algorithms in this case may be utilised as e.g. the WS HEU algorithm proposed by (Yu et al., 2007) and extending the HEU algorithm (Khan et al., 2002).

However, many real-world processes have services that are not in strictly sequential order. (Zeng et al., 2004) present a global planning approach to selection using integer programming and taking into account workflow patterns. In order to apply 0-1 integer programming, first the execution routes should be defined. An execution route is a route within a process from a start to end task, which includes only one branch in each conditional operation but all branches in parallel operations. Each execution path has then a probability assigned which is the product of all probabilities for all conditional branches selected in the route. If there are parallel operations within an execution route then sequential paths are created such as they include only one branch in conditional operations and only one branch in parallel operations. Then, the integer programming problem may be mapped as follows:

- variables definition – the selection variables in the selection problem are x_{ij} ($0 \leq i \leq N$, $0 \leq j \leq L$). If service s_{ij} is selected then $x_{ij} = 1$ otherwise $x_{ij} = 0$
- constraints definition i.e. exactly one service is selected for each service class; start-to-end constraints for execution routes;
- objective function definition maximizing the utility of the process.

Once, the selection problem is modelled as an ILP problem, well-known algorithms to find the optimal service selection can be used.

The approach presented by (Zeng et al., 2004) was first to tackle the global selection problem using the integer logic programming. Their approach has, however, some limitations as pinpointed in (Ardagna and Pernici, 2007). Authors in (Zeng et al., 2004) separately optimize each execution path and obtain the global plan by composing separate solutions according to the frequency of execution. Therefore, the fulfilment of availability and response time constraints is guaranteed only for the critical path. Furthermore, the assignment of services is obtained as the merge of assignments of services to each execution path. If a task belongs to multiple execution paths, then the task is executed by the service identified for the most frequently executed execution path. Note that, in this way, the fulfilment of global constraints cannot be guaranteed since the Web service composition problem is not separable (Ardagna and Pernici, 2005). Therefore, Ardagna and Pernici (Ardagna and Pernici, 2005, 2007) propose to model the service composition as a mixed integer linear problem where both local and global constraints are taken into account and tackle some of the limitations of the previous approaches.

In addition, some approaches tackle the problem of multiple process instances e.g. (Ardagna and Pernici, 2007) (i.e. if a very large number of requesters are assigned to the same best service, critical load conditions could be reached and the quality of service degrades) and perform the optimization per flow rather than per request basis (Cardellini et al., 2007).

The experiments performed in (Ardagna and Pernici, 2005) show that on average global optimization allows improving local optimization results by 20-30%. However, the problem with the global optimization is that an evaluation of all assignments results in an exponentially rising computation effort with the growth of the number of candidate services. Therefore, the straightforward evaluation of all combinations is unfeasible for a large number of candidates and the exact solution has an exponential complexity (Berbner et al., 2006, Yu et al., 2007). In order to improve the selection performance, heuristic algorithms are utilized that find near-optimal solutions in polynomial time e.g. (Ardagna and Pernici, 2007, Yu et al., 2007).

If we consider the composite structure as a graph model (more precisely Directed Acyclic Graph (DAG)) in which every edge has a set of quality attributes and a utility value, then the selection problem may be defined, as done in (Xiao and Boutaba, 2005), as finding a path from source to sink that produces the highest utility subject to the multiple constraints specified by the user. This is the well-known multi-constraint optimal path problem (MCOP) in the graph theory. In (Yu et al., 2007) authors, based on the algorithm of single-source shortest paths (MCSP) in directed acyclic graphs, propose the algorithm and its relaxed version that keeps only K paths on each node. After appropriate transformation, the MCSP as well as MCSP-K general algorithms may be applied also for general flow structures as showed in (Yu et al., 2007).

A comprehensive comparison between various selection algorithms may be found in (Yu et al., 2007) and is summarized in Table 17.

Table 17 Comparison of selected global algorithms (Yu et al., 2007)

Criteria	**BBLP**	**WS_HEU**	**MCSP**	**MCSP-K**
Running Time	very slow	fast	slow	fast
Memory usage	low	low	high	low
Optimality	optimal	near-optimal	optimal	near-optimal
Algorithm usage	very small size problem	large size problem	small size problem	large size problem

Heuristic algorithms to find close-to-optimal solutions in polynomial time are more suitable for making runtime decisions. However, the success of all algorithms (ILP and heuristic algorithms) eventually depends on the correctness and the precision of the data on service quality supplied. If accurate data is available, algorithms may produce excellent service selections. Otherwise, the results may deviate from the best ones.

When the problem size is small, we can use ILP to solve the problem, as proposed in (Zeng et al., 2004) and (Agarwal et al., 2005). However, high complexity of ILP does not work well for large systems with many tasks in a business process and many candidates to select from. Application of MMKP is not recommended as it does not allow to take into account process structure.

3.3.3 Other techniques

Apart from the most popular local approach (also called greedy selection) as well as the mentioned ILP and MMKP algorithms, there exists a few other selection techniques.

For example (Jaeger and Rojec-Goldmann, 2006) use a backtracking-based algorithm for the selection, called Discarding subsets. The algorithm uses a search tree which consists of nodes each representing a possible pair of candidate and a task. Each possible assignment for the composition is represented by a path starting from the root and ending at a leaf. The algorithm cuts, using two estimation heuristics, subtree representing unfavourable combinations to save computation efforts.

Another approach is a pattern-based selection (Gronmo and Jaeger, 2005) that constitutes a compromise between local and global selection. The pattern-based selection algorithm determines the best assignment considering each composition pattern in isolation. The algorithm performs four steps: the algorithm walks recursively into the structure and identifies pattern elements that do not contain any sub-patterns, then for all tasks within such an element, all sets of candidate assignments are evaluated and the combination that delivers the best score is selected. Then, in the third step, if the optimal solution for a particular pattern is determined, the algorithm walks one level upwards to evaluate the assignment within the new pattern. The aggregated NFP value of contained sub-patterns is taken as a fixed value. Finally, the pattern wise optimization and aggregation is performed until the whole composition is covered and one aggregated NFP value is returned.

Other approaches propose to tackle the selection problem by implementing genetic algorithms e.g. (Canfora et al., 2005, Claro et al., 2005). The work of (Canfora et al., 2005) is based on the reduction formulas presented in (Cardoso, 2002) and considers also global re-optimization. However, only suboptimal solutions are identified and global constraints are guaranteed only statistically. In turn, in (Claro et al., 2005), the multi objective evolutionary approach Non-dominated Sorting Genetic Algorithm is implemented. It identifies a set of Pareto optimal solutions, however, without considering a ranking among different quality dimensions. In the proposed approach, every identified solution is characterized by the fact that no other configuration exists such that a quality dimension is improved without worsening other dimensions.

The advantage of applying genetic algorithms is that they allow considering also non-linear composition rules for composite applications. However, they are not as efficient in computational terms as e.g. integer programming approach. In addition, in the mentioned approaches (Canfora et al., 2005, Claro et al., 2005) non-feasible solutions are also generated, thus, wasting some of the execution time. Moreover, sometimes no solution can be identified even when the problem is feasible, especially in cases in which the global constraints are stringent.

Other examples encompass application of a transiently chaotic neural network algorithm to perform service selection (Yang et al., 2005) and implementation of immune algorithm for selecting optimum Web services, which is based on the longest path method of weighted multistage graph (Yan et al., 2006).

3.4 Selection approaches and systems – summary

The service selection starts from an already defined set of potential service candidates which have been identified to be relevant for a defined goal within the discovery phase (Friesen, 2007). Within the selection process a specific configuration of service implementations is chosen based on non-functional properties and user requirements (preferences and constraints) using some selection technique.

Within this chapter, the selection model and exemplary approaches have been discussed. It has been shown that there exist many different initiatives that differ under the information model and selection techniques used. The current selection approaches lack uniform representation to capture business requirements and preferences. Most of the approaches take into account only price and some generic criteria and adopt at first local selection strategy approaches. Therefore, these approaches fail to meet the business users expectations as is shown within next chapter.

PART II: CONCEPTUAL MODEL

This part provides the conceptual model of the two-step business conditions aware selection of Semantic Web services for the needs of business processes. It describes the central concepts underlying the approach and provides answers to the remaining research questions posed in the introduction of this thesis.

To be more specific, first, it provides the overview of the requirements of the business users as well as expectations of the potential SWS e-marketplace participants towards the selection mechanism. The approaches presented in Chapter 3 are evaluated against the defined requirements. Then, the general model of the selection is presented, including the envisioned structure and interaction scenarios. Finally, the information model as well as selection algorithm are discussed in details. The explanations and models are based on the background information presented in the previous part and are tied together into a self-contained approach.

4 Two step business conditions aware selection model

This chapter provides an overview of the two-step business conditions aware selection mechanism. The most important elements of the designed framework are explained as well as all assumptions that are followed are listed and discussed.

This chapter is structured as follows. First, the requirements analysis is performed and the defined requirements are shortly described. Then, the overview of the business oriented selection mechanism is presented. Next, the details of the information model as well as the selection mechanism itself are given. Finally, the conclusions follow. All items presented in this chapter, unless stated otherwise, constitute a novel contribution developed within this thesis.

4.1 Requirements analysis

Definition of requirements is crucial when developing an information system (Maciaszek, 2007), therefore, this section provides informal requirements analysis of business users' (and potential SWS e-marketplace participants) expectations with the special focus on the requirements on the selection mechanism. Thus, the first goal of the thesis is addressed. This section is structured as follows. First, sources used to derive the requirements are discussed. Within the next subsection, the requirements are formulated. Finally, the evaluation of the currently existing approaches described in the previous chapter follows.

4.1.1 Sources used to derive the requirements

Various examples of usage of SWS are presented in research (Abramowicz et al., 2006c, Lord et al., 2005, Paolucci et al., 2003a) and white papers (Fahringer et al., 2007), as well as in research projects deliverables (DIP, 2007, Fahringer et al., 2007, Kuropka et al., 2008). The examples encompass a number of different applications of SWS systems within organizations, but most of them are in fact just prototype demonstration systems taking advantage of the SWS technology[35]. There are many scenarios arguing the value and the need of using SWS to perform service composition, like: travel booking (Stollberg and Lara, 2004), attraction booking (Kuropka and Weske, 2006), buddy scenario (Noll, 2004), health care service (Tilsner et al., 2005) or

[35] Some examples are: B2B application in which a business that assembles computers automatically finds partners providing parts and automatically transacts with them (Paolucci et al., 2003a), interoperable E-Tourism Marketplace (Fodor and Werthner, 2005), an e-commerce application that helps a user to organise a trip to a meeting automatically interacting with different Web services and the calendar of the user stored in MS Outlook (Paolucci et al., 2003a).

dynamic supply chain (Verma et al., 2004). However, there is no record of an enterprise-strength successful implementation of SWS. The situation regarding the availability of the B2B SWS e-marketplaces has been discussed in Chapter 2.

Therefore, taking into account the immaturity of SWS applications and the still emerging SWS e-marketplace (or services ecosystems), in order to derive the requirements on the selection mechanism various sources needed to be used. The requirements formulated within this section are based on the careful studies of the relevant literature from the area in question, international projects on application of SWS, interactions with use case partners coming from four European Projects focusing on SWS as well as from an on-line survey conducted. The performed studies took the form of expert group studies with the strictly designed group selection procedure.

The conducted literature studies and market observation encompass the analysis of the available products, industrial blogs[36] as well as studying of reports relevant to the topic in question, published by various organizations e.g. Gartner Group[37]. The literature studies encompassed also research papers published as journal publications as well as considered proceedings of top conferences from the relevant field [38].

The analysis of various research projects, the author participated in, on the SWS technology has been performed[39]. Also outcomes and use cases of other projects have been considered and an attempt to obtain feedback on achieved outcomes from use case partners has been undertaken. The additional input to the requirements formulation has been taken from the personal communication with the business partners (encompassing both service providers as well as service consumers) from the mentioned projects. Also informal interviews with representatives of SAP, IBM and IDS-Sheer were performed for the needs of this dissertation. Finally, the on-line survey was conducted together with the group of researchers from the Department of Information Systems, Poznan University of Economics, targeted at polish companies, in order to get to know their opinions on the WS and SWS technology[40].

Based on the gathered information, requirements regarding the selection mechanism may be formulated. The list and description of the requirements follows.

4.1.2 Requirements

The identified requirements are divided into two groups: requirements that will influence the information model (requirement 1-6) and those that concern the selection

[36] E.g. Semantic Business Process Management Blog http://www.arisblog.com/2008/07/14/semantic-business-process-management/; Application Platform Strategies Blog http://apsblog.burtongroup.com/.
[37] http://www.gartner.com/.
[38] The top conferences are e.g. as follows: WWW, VLDB, ESWC, SOC.
[39] Those projects are Knowledge Web, Adaptive Services Grid, SUPER and Service Web 3.0.
[40] The result of the performed survey are discussed more in detail in technical report: (Kaczmarek et al., 2009).

technique per se (requirements 7-10). They inform both about what users expect from the selection mechanism as well as which aspects should be considered.

The requirements relevant to the information model could be summarized as the need to incorporate business-oriented view (not only technical one) within the selection process to a far more extent that it is currently done. This manifests itself in a need of correspondence between the information model used by the selection mechanism and business perspective resulting in describing the artefacts using the proper terminology. The requirements relevant to the selection techniques indicate the need for personalized and user-tailored selection procedure. They also emphasise that the selection procedure should not ignore the business reality i.e. service provisioning conditions and business transactions features.

Each of the requirements is shortly discussed within following sections.

Flexible, extensible and commonly accepted NFP model

A selection system should support a rich and flexible set of selection criteria. As pointed in (Lin et al., 2008) and postulated by each use case partner, consumers not only expect a service to meet certain functional aspects but they also demand good quality of services such as service reliability, security, trust and execution cost. In fact, quality of service along with other non-functional properties is one of the most important factors for user's choice of a Web service. Therefore, the appropriate and extensible set of NFP properties should be defined and should constitute a NFP model customizable for use in any domain. Such a standardized but extensible model should be used not only by the selection mechanism but become a basis to form common SLA and monitoring mechanisms as well as other elements that should be a part of every mature marketplace.

Fair and open computation of NFP values

The selection model should rely on up-to-date and accurate values of selection criteria. Therefore, the selection mechanism should not be based only on information provided by service owners, but also on information provided by the users independently of the providers (Yeom and Min, 2005) as well as by testing and monitoring facilities.

The users postulate that it is imperative to devise techniques to publish less subjective quality values to assist service consumers in selecting services according to the desired level of quality. The approach followed must allow information needed about a service to be supplied as and when required without placing an unacceptable overhead on service providers and consumers(Walkerdine et al., 2007).

Inclusion of business properties

If one wants to understand the nature of a Web service within the SOA paradigm and the requirements of clients addressed toward the WS provisioning, one should not only pay attention to the IT area, technical procedures and generic non-functional properties, but also should take into account the business side of the issue. It seems that WS and SWS mechanisms focus too much on the IT side of a service without paying proper attention to business aspects (Hidalga et al., 2006). The approaches cannot be centred on the collection of networking level criteria such as execution time reliability or availability. More business oriented properties and criteria should be also included (Verma et al., 2005b). Therefore, the information model should support a user in inclusion of business-oriented properties of choice. In addition, business rules and policies as well as SLA also could be taken into account.

A very interesting issue underlined by business users is a need to include the service provisioning conditions and dependencies between services. A very good example is a need to address the statefull services that need to be used together. Another example is that none of the mechanisms considers an existence of so called service bundles. A service bundle is a set of services that if used together is offered at a discount price or with other benefits.

The inclusion of business aspects will allow to ensure that the decision about which configuration of services to use is not made economics-independent (Boehm and Sullivan, 2000) and with focus only on functional or network-related aspects.

Facilitation of transition from the business to the more technical layer

Semantic Web and particularly Semantic Web services promised to enable users to perform complex tasks without requiring an understanding of the underlying technology (Born et al., 2007). However, right now, it seems that this promise is not fulfilled. Clients are required to provide a semantic description of their goals/problems and attach preferences to the technical criteria they do not understand.

It is said that "business needs and not technology will drive the use of WS in enterprise" (Mimoso, 2004). In order to ensure the successful application of SWS and SOA based solutions it is expected by the industry (business users) that different layers of abstraction should be used depending on the user and its role in the organization (Brelage and Domingue, 2008). For instance, within the Business Process Management domain, as postulated by the SUPER research project mentioned in Chapter 2, four layers of abstraction may be distinguished, as depicted in Figure 11.

Similar layers should be applied in the case of SWS e-marketplace. Participants of each of the abstraction layers have different information needs and operate on different artefacts. The SWS e-marketplace is targeted at business users. Therefore, their expectations and their abstraction layers should be taken into account by the selection mechanisms and only then mapped to the more technical layer required by the Semantic Web services.

Figure 11 Abstraction layers in Business Process Management

Flexibility and fuzziness in the specification of selection requirements

SWS e-marketplace participants have a number of different expectations (Papazoglou and Heuvel, 2007). In the context of business-to-business services, business clients' expectations are driven by the expectations of their own customers. For example the need to keep the system up and running all the time is not just the company own expectation but it is derived from the pressure of customers (Zeithaml, 2003). Other issues are also standards set by other service providers in the same domain.

The business constraints and preferences tend to be informal, subjective and difficult to quantify, which poses great challenges in automating the business process selection and composition while incorporating business requirements. Therefore, it is critical to properly formulate the descriptive and subjective requirements in quantifiable and objective machine-readable formats. The business users expect to be able to define both quantitative and qualitative constraints and preferences and at the desired level of abstraction. The developed mechanisms should be flexible and allow for specifying various requirements to various processes and for various clients.

The system should allow users to specify business constraints and preferences in a user-friendly manner.

Context

Different preferences on the services are applied by users based on users' current context, location, activities etc (Lamparter et al., 2007). The selection mechanism needs to be able to model the context and then take it into account.

The context may be understood in many ways. It may e.g. encompass information whether the selected service is to become a part of the short or long-term relationship (Benatallah et al., 2002). In order to get to know the preferences regarding the process, it may be also necessary to consider division of processes into use-case dependent groups, as e.g. done in (Friesen and Namiri, 2006) where the following scenarios were distinguished:

- a process takes place sporadically and has a low value from the company point of view. The default preferences should be used;
- a process takes place sporadically but has a very high value. The results (i.e. quality of a process) should be approved by a user;
- frequently occurring process over some period of time (appropriate balance between the price and quality is expected);
- a business user sets constraints for a process model. They may differ in time.

It is also very important to take into account other business context e.g., partners the organization is collaborating with, the context of the process the selection is performed on, as well as the context of the entire organization.

Personalized, global and business properties aware service selection

Personalization has long ago been discovered as a crucial factor. The quality of the composite service resulting from the composition process is a determinant factor to ensure customer satisfaction (Zeng et al., 2004). Different users may have different requirements and preferences regarding the values of NFP. Thus, a personalized approach to service selection is a must (Zeng et al., 2004).

Two approaches allow specifying constraints at local and global level respectively. A local constraint allows selecting a Web service according to a desired characteristic. Global constraints pose constraints over the whole composition. An end-user is mainly interested in global constraints (Ardagna and Pernici, 2005).

Trust and contract-aware selection mechanisms

As pointed in (Petrie and Bussler, 2008), although a huge potential exists for open source services and technologies for combining them and creating business processes, perhaps on the fly, business people want to use trusted services and technologies from recognizable brands.

An organisation is not likely to use services of other organisation found by an automated service discovery. Usually business relations need a contract or similar explicit agreements. This problem relates to reputation, trust and contracting issues in this field. Survey conducted by Merrill Lynch found that companies conduct 90% of their business with established trading partners and only 10 percent on the "spot" market, or via auctions (King, 2002). The second study conducted by BroadVision with McKinsey discovered that only about 30 percent of B2B revenues are directly related to cost savings or efficiency gained through transactions. The conclusion is as follows, a service provider and consumer need a contract in place just as with any other non-technical customer-provider arrangements. However, the appropriate mechanism is required to deal with the dynamism and take into account that new collaborators appear all the time.

In addition, the selection mechanism should also take into account the dependencies that may appear between providers and their services. For instance, that one service needs to be used with another or that they constitute a bundle, as defined earlier.

Dynamic binding and changes support (re-selection)
One of the original goals of SOA was dynamic binding of Web services to existing business processes (as pointed in (Verma et al., 2005b)). The business use cases such as those pointed in (Verma et al., 2005a), show that businesses are trying to build infrastructures, which will allow them to integrate the most optimal partners in their business processes.

As Web services operate in a Web i.e. highly variable environment, their NFP characteristic may evolve relatively frequently, either because of internal changes or because of changes in their environment (Zeng et al., 2004) (e.g. higher systems loads). Therefore, during execution of a composite process, the component service may change its properties, become unavailable as well as new services may emerge. From this point view, the approaches where WS are statically composed are inappropriate (Zeng et al., 2004), thus, a dynamic composition should be considered. However, one cannot forget that the dynamic composition approach may not result in selecting an unknown or unacceptable business partner.

Precision, scalability and computation efficiency of the mechanisms
A lesser requirement, yet nonetheless critical for resource-constrained environments, is that the mechanisms should be designed for communication efficiency and computational tractability. Precision entails that the mechanism should guarantee at least sub-optimal solution. Therefore, it needs also to support at least basic workflow patterns.

The scalability means that it should be possible to use the methods independently on the tasks number or WS. All of the defined rules should be applicable in various cases.

4.1.3 Evaluation of current approaches using the defined requirements

Table 18 summarizes the fulfilment of requirements defined within the previous section by the existing mechanisms and approaches to service selection.

The conclusions is that the defined requirements are only partially supported by the existing mechanisms. This hampers the adoption of the SWS-based mechanisms. This is mainly due to the fact that for the first generation of Web services the target user was a programmer. However, right now the target users of the e-marketplaces (and also B2B SWS e-marketplace) are domain experts, consultants and business specialists implementing business processes through service composition (Hidalga et al., 2006).

The current description stack and interactions are still tailored to the needs of the developers and not business people.

Table 18 Evaluation of current approaches using the defined requirements (own study)

Requirements	Level of fulfilment	Comments
Flexible, extensible and commonly accepted NFP model	Lack of commonly acceptable model	There is a lack of accepted NFP model. In addition, most of the existing ones are not extensible and offer insufficient information on quality of WS (Yeom and Min, 2005).
Fair and open computation of NFPs values	Supported	The appropriate mechanisms have been put in place to gather the reliable information. For instance service profiling developed within the ASG project.
Inclusion of business properties	Partially supported.	Some initiatives to consider business rules and SLA have been undertaken on the SWS e-marketplaces. However, no selection mechanism considers KPI or other business oriented metric
Facilitation of transition from the business to the more technical layer	Not supported	It is not addressed by most available solutions as they are targeted mainly at the last or second-to-last abstraction layer.
Flexibility and fuzziness in the specification of requirements	Not supported	The current mechanisms lack flexibility and do not offer the support for various levels of abstraction. The decomposition is not supported.
Context	Not supported	The inclusion of business context and different types of relations that may exists is not included in the existing approaches
Personalized, business properties based and global constraints aware service selection	Partially supported	Most of the selection mechanisms are personalized and NFP based. However, no profile suited to business needs has been defined so far. The specification of business constraints and preferences is not adjusted to the business needs.
Trust and contract-aware mechanisms	Partially supported	Only few approaches take into account the existence of the SLA during the selection.
Dynamic binding and change support	Supported	The dynamic biding and runtime service selection is possible
Precision, scalability and computation efficiency of the mechanisms	Supported	The existing mechanisms either use the algorithms ensuring finding the optimal solution but with exponential complexity or usage of heuristics is envisioned with polynomial complexity.

In addition, the lack of business perspective in the mechanisms often results from the fact that the people designing them are not involved or do not understand enterprise-level value creation objectives. In fact the connections between functional and non-functional (value-creation) are rarely understood. Therefore, in many cases the technical criteria (not business ones) are taken into account and thus the selection is not optimal in business terms.

The last issue is connected with the fact that some SWS initiatives in the pursuit to automate everything what may be automated, fail to take into account the exact business perspective. Business people do expect a helping tool that would support them in quick reconfiguration of existing services and processes, and help them dealing with e.g. cancellations and unexpected events. Also, business people rarely want dynamic contracts automating all business to business interactions as most interactions require something more than just following a set of algorithms and rules. Therefore, the business users emphasize that one cannot automate the interactions at any price but the appropriate balance between business and technical side should be respected.

Within the next sections, the proposed selection model that is to solve some of the indicated pinpoints is discussed.

4.2 Proposed model - assumptions and general scenarios

According to (Hars, 1994), a model is an immaterial representation of a relevant part of the real world. The two-step business-oriented selection model presented within this section has been tailored to the expectations of B2B SWS e-marketplace participants. In order to discuss the created model, a few elements need to be defined as well as assumptions taken need to be explained.

First, it is assumed that there exist a number of SWS, SWS providers as well as business users interested in service outsourcing and collaboration with other parties.

It is assumed that the selection model presented is a part of a fully-fledged SWS e-marketplace[41] that offers services (see Figure 12). The e-marketplace may take form of an open market, but also an IT service park for collaborating partners (Petrie and Bussler, 2008) or internal repository of services used by multi-branch organization.

In addition, it is assumed that the marketplace is equipped with a user-friendly interface that allows a user to share with the e-marketplace its process models and to perform service composition[42]. This interface may be also used to facilitate the creation of a user profile used within the system[43]. It is assumed that the participants trust the e-marketplace and are willing to share information about their processes, business rules and all requirements in general with the developed marketplace mechanisms.

In addition, it is assumed that there is no need for ontology mapping as all market participants use the same ontology. However, if different domain ontologies would be

[41] As discussed in chapter 2 the fully-fledged SWS e-marketplace is still under development.
[42] The mentioned interface may be implemented in a form similar to BPMO studio developed within the SUPER project (Dimitrov et al., 2007).
[43] An example of such an interface may be the F-WebS system (Abramowicz et al., 2006c) created for the needs of the master thesis.

used to annotate services, the mediation techniques such as e.g. the ones presented in (Bruijn et al., 2006), may be utilised.

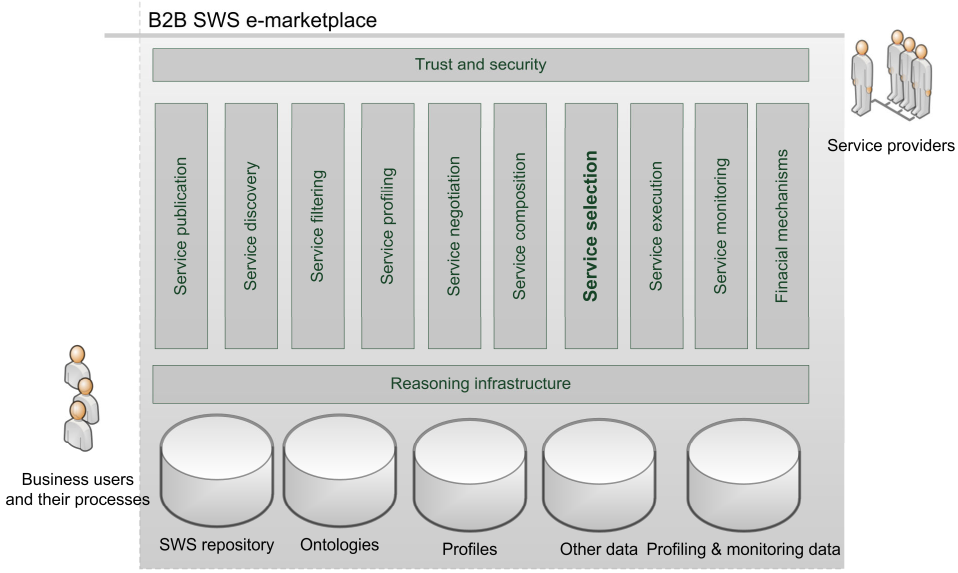


Figure 12 SWS e-marketplace and the selection mechanism

It is assumed that SWS are represented using one of the existing semantic representation language. In Chapter 2, it has been highlighted that there exist a number of different, often competing, mechanisms that can be used to describe and represent a service (also a semantic one). Therefore, within this dissertation a fairly abstract view of a service is followed and for the needs of selection it is considered to be assigned to a certain capability (i.e. to have a certain goal) and to be fully described by a set of properties: $A_1, \ldots, A_n$. These properties are defined within the information model used by the SWS e-marketplace and its participants and encompass service non-functional aspects as well as business properties. Such a general description of SWS allows abstracting from various existing Web service description frameworks, while simultaneously allowing application of existing algorithms.

In addition, it is assumed that to each capability there are a number of different services assigned. Therefore, a service capability (i.e. a service goal) is deemed to be a service specification whereas a concrete service is a service implementation that has some specific values of non-functional attributes assigned (e.g. provider, quality level). The result of composition is a process model i.e. a template for a class of similar business processes performed within an enterprise (Leymann and Altenhuber, 1994) created by appropriate algorithms[44] (e.g. as done in ASG, SUPER or other projects). Each individual process is an instance of a process model, and it represents a concrete, specific execution of a variant prescribed by the process model. The fundamental

[44] As e.g. composition algorithm developed within the SUPER project that has polynomial complexity (Hoffmann et al., 2007).

building block of the process model is a task. A task represents a business action that has a service capability (i.e. a goal) assigned and therefore is implemented by one or a set of Semantic Web services registered on the e-marketplace.

In addition, it is also assumed that the SWS capability matching is implemented using state of the art algorithms achieving a good performance. One of the main presuppositions of the selection is that the matchmaking process identifies functionally suitable candidates and that this step may result in more than one candidate for a particular task. Otherwise, if the matchmaking has determined only one candidate for each task in the process, then the selection is trivial and the discovered service is assigned to a process model.

It is also assumed, for the needs of simplicity of discussion and following other researchers in this field e.g., (Ardagna and Pernici, 2007, Kuropka et al., 2008) that one service encompasses only one operation. This assumption is not too restrictive. If a service has more than one operation, within the e-marketplace each operation is registered as a separate entity (having its own capability assigned).

Therefore, informally the selection problem on the considered e-marketplace may be formulated as follows:

> *The problem of automated selection of optimal*[45] *configuration of offers (i.e. semantic service implementations) made by service providers and registered at the marketplace given a request consisting of a process model, selection criteria and business requirements.*

The following selection system, tailored to meet expectations and requirements of a user, operating on the above described B2B SWS e-marketplace is proposed.

The selection is divided into two steps and takes place both at the design time and the runtime. In addition, the re-optimization stage triggered by the occurrence of the relevant change in the environment is introduced. The first step of the selection process aims at identification of all providers and configuration of their services that meet the binding constraints and slow-changing context defined by users as well as business conditions (e.g. trusted parties, competitors). It allows efficiently narrowing the set of possible services that may be used to execute the process. Then, the second step focuses on the identification of the best configuration taking into account local and global constraints, service dependencies as well as additional rules defined by a user. Thus, the selection of multiple services occurs through an initial filtering of a set of feasible solution for each task depending on business agreements and end users needs. Then, the subsequent, context-based selection of the most appropriate solution takes place. The two-step algorithm may be utilized on the e-marketplace in the following way.

[45] Or sub-optimal depending on the algorithms used.

The first stage takes place at the design time, aiming at selection of all providers and configuration of their services that meet the binding constraints and slow-changing context defined by users. The most important are requirements applied to static or semi-static properties. The result of this stage is two-fold. First, the set of possibilities to each task meeting the requirements is ranked according to the utility function derived from the user preferences. Second, the global optimization takes place and the optimal service configuration considering the process structure and requirements is identified.

Then, during the execution (run-time) stage the system checks whether the attributes' values of a service or of a user profile have changed. If the change is minimal or irrelevant, the services of previously computed optimal configuration are chosen and executed to perform the task. If, however, a relevant change has occurred, the re-optimization (i.e. re-selection) takes place.

In addition, in order to deal with the dynamism of the environment (e.g. new service providers and new services), the re-optimization not directly connected with the execution of the process itself has been introduced. It is connected with events like: registering new service, changed characteristic of a service, change in the contract or a user ceasing to trust a certain business partner.

The two-step selection sequence is depicted in the following sequence diagram.

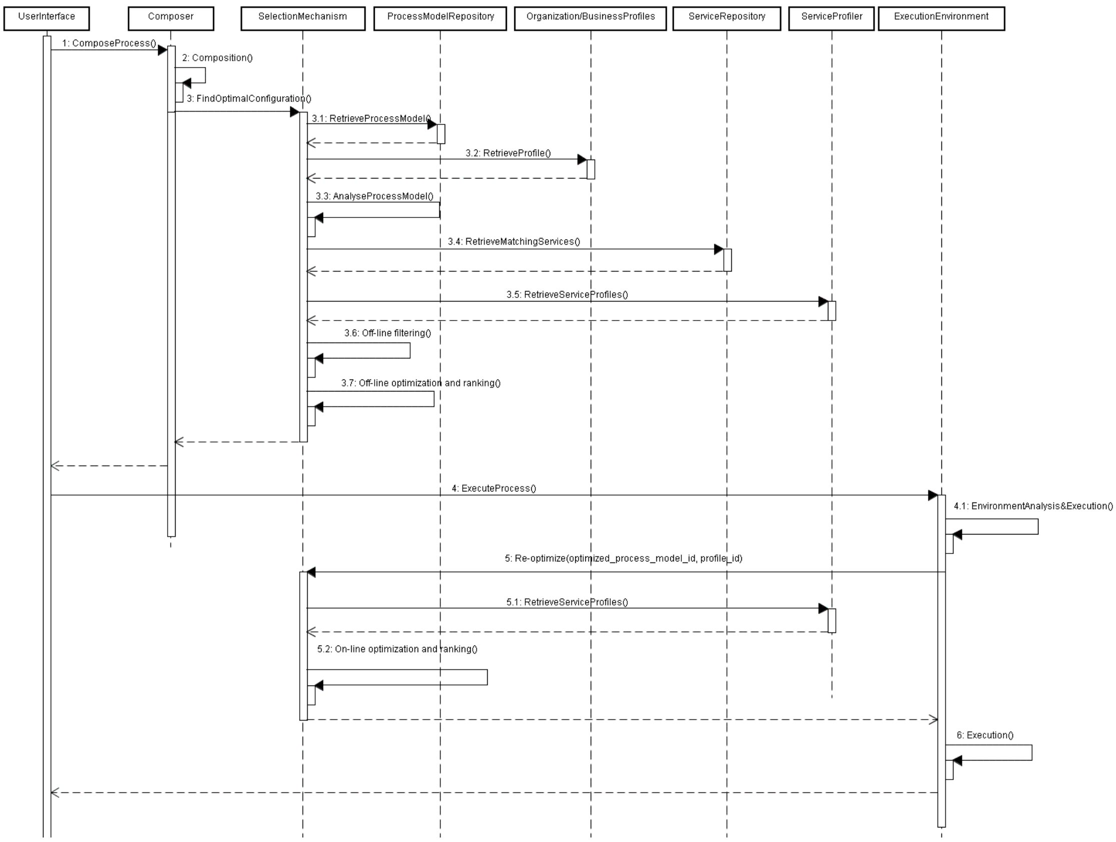


Figure 13 Sequence diagram – two-step selection (see Figure 42 for enlarged version)

A selection mechanism receives a request from a composer to optimize the composed process. The design service selection takes place. First the process model to be optimized need to be retrieved from the process model repository along with the set of profiles (i.e. a set of requirements) needed in order to derive the selection criteria of interest. The process definition is expressed using the information model provided by the e-marketplace. The process model provides not only information on the process itself but also it encompasses the information about specific requirements attached to the process, process fragment or a single task as is explained further.

A set of alternative feasible services per task is provided through a service discovery mechanism provided in the interface to the repository. Those services may come from various providers or include different quality classes of the same WS. Finally, the information on the services is retrieved from the service profiler mechanism. A service profiler module stores performance information from previous WS invocations as well as it is able to create dynamically service profiles on demand. Once, all the mentioned information is retrieved, the design-service selection may start. The result of this phase is the optimal configuration of services to perform the process along with the ordered list of services to each task that may be used. The optimal assignment as well as the ranking may be agreed upon by a business user who is presented with the final result.

The design-time selection focuses mainly on the slow-changing context (it may be associated with business context and rules as well as existing agreements between service providers and business users). Appropriate profiles are used to capture such information and utilize it to personalize the content, apply preferences and privacy considerations while evaluating and selecting a service. This information is used to generate the rules that specify evaluation and selection criteria for competing service providers. For instance, in case of price sensitivity, the design-time filtering will try to filter out costly services while selecting a service.

Once the process has been approved by a user and invoked, the execution engine analyzes the current context and tries to identify whether the relevant change has occurred in the environment. If the relevant change is spotted by the system, then the dynamic service selection (on-line service selection) occurs. Here, appropriate services (and their configuration, if appropriate) are selected based on the current state of the environment. It means that if the characteristic of the selected service implementation has changed and does not meet the customer's needs any longer or becomes untrustworthy or a better implementation is found, then the service linked to a given task needs to be replaced. So, if the change in the list of available services implementing the capabilities is discovered, the optimization is performed. Therefore, at runtime, if the service still meets all the runtime requirements then, the service is selected and executed. The said performance information comprises a long-term component that considers multiple instances of previous web service invocations (selection at the design-time is therefore based on the long-term profile) whereas within the second phase a short term profile (context dependent) is taken into account.

The big advantage of this approach is that the actual service selection logic includes both static (off-line) optimization component and a dynamic self-managed context-sensitive selection mechanism.

In addition, in order to deal with the dynamism of the environment i.e. new service providers, new services as well as process appearing all the time, the re-optimization not directly connected with the execution of the process itself (in opposition to the scenario described above where the re-optimization is performed at the runtime) has been introduced. It is connected with the registering new WS or changed characteristic of a service, change in the contract or a user stopping trusting a certain business partner etc.

The following sequence diagram presents the idea. As emphasised by (Zhang and Li, 2004) when creating or updating a business process composed of Web services, an important aspect is to meet the new and evolving business requirements. Thus, each new or changed service is treated as a new stream of information coming to the e-marketplace and thus standard filtering procedure may be utilised. If the service has a capability assigned that is used within some process of a business user, then the new service is compared to the already existing ones and its overall score (utility function) is counted. If the service is potentially interesting for the user, the re-optimization takes place and a user is informed about this. Similar process may take place if there is a change in a user profile and the user indicates that all processes should be re-optimized.

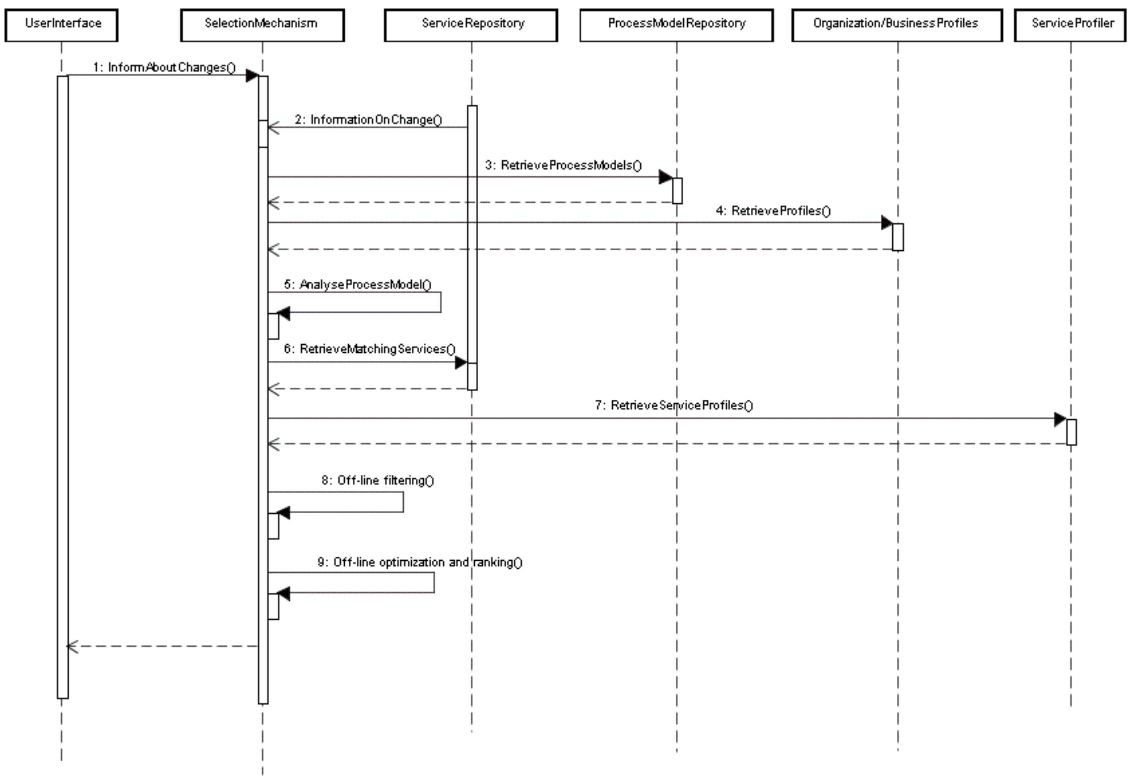


Figure 14 Sequence diagram - re-optimization (see Figure 41 for enlarged version)

Therefore, the re-optimization may be triggered in the following cases:

- the current values of non-functional property differ from the corresponding prediction included in the service profile generated by service profiling component more than by a given threshold. For examples of such thresholds see (Ardagna and Pernici, 2007),
- a Web service invocation fails,
- a change in a user profile or preferences (change in the objective function),
- optimization is performed statistically i.e. by evaluating branch conditions and probability distribution of number of loops iterations. Re-optimization is triggered after the evaluation of branch conditions resulting in different probability distribution.
- new relevant services are registered to the market.

Thus, in order to fulfil the above-described scenario and ensure two-step business oriented selection the following issues need to be addressed:

- provide a representation of user needs and requirements that is expressive, efficient and automatically processable;
- provide a representation of non-functional and business-oriented properties on the appropriate level and support and facilitate the transition between various layers. In addition, this model should be commonly accepted and relevant to all domains;
- define a set of rules (R) that would allow to identify/define the set of relevant configurations;
- define the scoring function (F) for each of the attributes from the model;
- apply an efficient algorithm for computing a near-optimal or optimal (depending on user preferences) configuration of services and quality levels, given a specification of the user preferences and the current state of the environment. This algorithm need to take into account the process structure (workflow patterns);
- provide a mechanism to minimize disruption to a user resulting from changes in the environment or in a user intent (re-optimization).

The following sections provide an in-depth description of the two-step business oriented selection algorithm starting from the information model it requires.

4.3 Information model for business-oriented selection

This section presents the information model developed for the needs of business-oriented selection. Thus, it addresses the second research goal defined. The information model encompasses all data necessary from the point of view of the selection mechanism in order to perform its task (according to scenario presented within the previous section) and fulfil, at least partially, identified requirements of business users (4.1.2 Requirements).

Dealing with service selection on the B2B e-marketplace, one deals with artefacts at different levels of granularity. A business user models a process that consists of various tasks to which capabilities are assigned during composition and then service implementations during the selection stage. Each of these artefacts uses various resources to provide the desired result. The above-mentioned granularity results in the distinction of three levels: process, service and resource layer. Additionally, the process level may be divided into a process sublevel, process fragment sublevel as well as task sublevel, as done in (Kaczmarek et al., 2008).

In addition, in the context of the B2B SWS e-marketplace another perspective becomes important, namely a user and organization specific requirements that may be placed above the process level. Thus, the following hierarchy depicted in Figure 15 may be distinguished encompassing additionally a business, process and technical view. A business view is further on decomposed to an organization as well as business user context. The process view encompasses process, process fragment as well as task layers whereas the technical view focuses on services and resources.

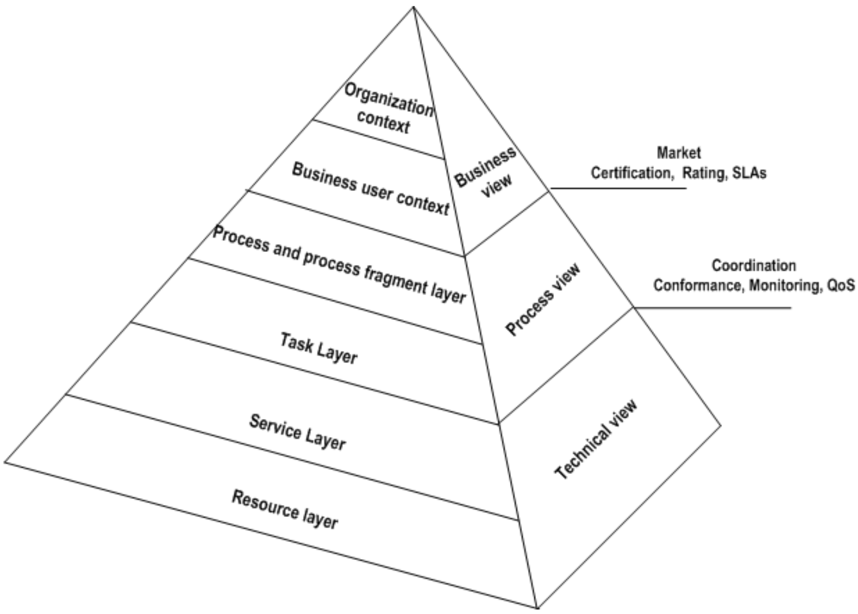


Figure 15 Information model – different perspectives

A business view encompasses strategic business goals and policies modelled on the level of the entire organization (they constitute an organization context). The given organization may have various departments and many employees. The organization context is therefore shared by a number of business users. Each of these users, in correlation with their role in the organization, may have some general requirements like e.g. "I want my parcels to be delivered only by X company and I always want to choose the most reliable service" (constituting a business user context). In turn, each business user models or owns some process model and may have some requirements connected with the specific process model (in addition to the general requirements that are expressed in a business user profile). The process model in turn, decomposes into process fragments and tasks and therefore, each requirement from the higher levels needs to be decomposed in order to be applied within the lower level e.g. to the tasks. The process model specific requirements (those that apply to a process, process fragments as well as tasks) constitute the process model profile. In turn, services are assigned to a process model. Each service has its own characteristic (service profile) that is considered while performing the selection.

The matter of utmost importance, neglected by most approaches discussed in previous sections, is that business users in general are interested in defining constraints and listing requirements on the process level of abstraction. A business user is able to particularize whether his demands are directed towards specific process fragments, tasks or the whole process itself, however, usually he is not interested in applying the requirements on the level of service or even more so on the level of resources. Therefore, the requirements (information) from the higher levels need to be automatically propagated down to the lower levels in order to allow for appropriate service selection[46].

The description of the information model that encompasses all of the mentioned perspectives and contexts and facilitates transition between various views is described in the following subsections.

4.3.1 Knowledge representation technique and modelling procedure

An ontology is a description of entities and their properties as well as relations (Grueninger and Fox, 1995). In the domain of SWS, selecting the ontology as a knowledge representation technique to be used to create the information model and also to overcome the ontological conflicts that may occur between e-marketplace actors, seems to be an unquestionable decision.

The developed information model expressed using ontology ensures that providers use a standard mean to express attributes of their services and simultaneously ensures that business users (i.e. consumers) use the right mean to express their preferences and therefore facilitates matching between information provided by providers and consumers. By modelling certain attributes as classes in a Semantic Web language, they can be classified into attribute hierarchies and model additional relations that exist between various concepts. Moreover, the ontological representation of the information is expressive, machine-understandable and can be automatically processed. One of the advantages of using semantic description language is that one can use logical reasoning, in particular class subsumption, to bridge different levels of abstraction (as mentioned in the following sections). In addition, the ontology is a concise and complete representation technique, flexible and easy to understand. Although, the reasoning is

[46] However, it may happen that a need arises for a business user to examine also technical details of the service constituting tasks being crafted and this should also be made possible by the model.

deemed to be not so efficient, the state-of-the-art reasoners are proved to be both efficient and effective (Bishop and Fischer, 2008).

There exist a few methodologies used to design an ontology. They all consider the following steps: definition of the ontology purpose, conceptualization, validation, and finally coding. The conceptualization is the longest step and requires the definition of the scope of the ontology, definition of its concepts, description of each one etc.

The ontology model presented in this section has been designed following the above-mentioned steps. The purpose is to define an information model describing the knowledge relevant to the two-step business conditions aware service selection on the B2B SWS e-marketplace as described earlier. The conceptualization step has been based on extensive study of the literature and the experience of the author gained from the participation in SWS related projects (as already mentioned) as well as taking into account the identified requirements. In addition, the developed model has been created following the enumerated guidelines:

- guideline of correctness – the information model should be both semantically as well as syntactically correct. The model is semantically correct, if it reflects the real business needs, the syntactic correctness demands the appropriate use of the modelling technique;
- guideline of relevance – an information model should not describe elements that are not relevant. Only the important criteria should be defined.
- guideline of clarity – it requires that a model should be intuitively readable, well-structured and illustrative. This ensures that the model can be used in the future.
- guideline of economic efficiency – the modelling effort should not exceed a specific cost-benefit ratio which means that the effort of creating the model should not be higher than the benefit of the model itself.

The above-mentioned guidelines were followed within the conceptualization process as well as while defining the concrete requirements.

4.3.2 Requirements and informal competency questions

The requirements described below are on one side particularisation of the requirements listed in 4.1.2 and additional analysis and interviews performed with use case partners from European projects, and on the other, they result implicitly from the developed two-step selection mechanisms and followed assumptions.

There exist a number of requirements that should be met by the information model. At the highest level, they may be divided into two groups: functional and non-functional ones. Functional requirements strictly relate to the interactions and processes in which the model could be used. Non-functional requirements address the issues of inter alia applicability, quality and usefulness of the proposed model.

When it comes to the functional requirements, primarily the ontology in question should support selection of services taking into account non-functional and business aspects of various artefacts. Although the main aim is to use the ontology within service selection, it may be also applicable to other interactions like filtering, composition or negotiation. The model proposed should meet at least the following functional requirements:

- support various perspectives that may be applied to the model i.e. business view as well as more technical one (as explained in previous section);
- appropriate coverage of business oriented properties – it needs to include the properties that would take into account properties of interest to a business user at the level of the entire process. One of the most important elements is that the proposed enhancements should allow thinking/acting more strategically. This requires the mechanisms to step away from purely technical criteria that are not linked to enterprise level value outcomes;
- appropriate coverage of non-functional parameters - the ontology of non-functional properties must be adapted to be a framework for all important service and process related parameters;
- flexibility in specification of requirements on various artefacts;
- SLA compliance - the ontology of properties should allow for mapping to the most popular concepts and expressions used in the standards of SLA languages and templates;
- it should support the description of process model and all its elements allowing to express various requirements to the entire process or its elements and to link services to tasks. In addition, the description should support the formulation of the selection problem as ILP problem;
- the information model should reflect dependencies existing between services as well as between various business actors;
- reflect the relation between technical-decisions and enterprise-level value maximization - i.e. the mechanisms of the SWS e-marketplace should take into account and understand and as well as reason effectively about the mentioned relations. Therefore, specific properties of services have to be linked adequately to business value (to properties of processes and the entire enterprise portfolio);
- provision of measurement methods - if ontology is to be used in the interactions between services, it needs to provide guidelines for computation of quality parameters. These methods should reflect the nature of parameters.

Having briefly described the functional requirements, the non-functional ones are considered. Putting the SWS selection vision into practice requires ontology reflecting various spheres and operations meeting the following requirements:

- consistency – consistent means that the modelled concepts are based on compatible paradigms, have a compatible degree of detail, and include at least partial sets of alignment relations which allow data interoperability;
- size – the ontology should not be too complex. It should be maximally lightweight. Only the most important attributes need to be included in the ontology;
- extensibility - the ontology should be extensible. When the need for an addition of some concepts to the ontology arises, it should be possible to expand the ontology;
- modularity - the attributes should be logically grouped, dependencies between them should come from the hierarchy between attributes;
- reusability - the ontology must be reusable. It should be possible to describe the concepts and later use these descriptions in different interactions;
- operational - operational in this context means that the ontology specification is available in a single, current ontology formalism for which scalable repositories, reasoning support, APIs, and tools are available.

By fulfilling these requirements, the ontology has a great opportunity to become useful and flexible tool to be used in real-world cases.

Given a selection scenario, problem model, as well as requirements described above, a set of queries have been defined that place demands on an underlying ontology. These queries may be considered as the expressiveness requirements that are in the form of the questions (Uschold and Grueninger, 1996). An ontology should be able to represent these questions using its terminology, and be able to characterize the answers to these questions using the axioms and definitions (Uschold and Grueninger, 1996). The following informal competency questions have been formulated:

- What is the value of a characteristic X of a given service?
- What are the constraints connected with the task A?
- What preferences has a user towards the given process model?
- What is the type of the property?
- Which elements does the process consist of?
- Which workflow structures are present?
- Which service may realize the goal of a certain task?
- Who is the provider of the service?
- What are the requirements of the organization towards its processes?
- What kind of collaboration exists between partners?
- Which properties are defined within the SLA?
- What is the unit of the property?
- How the given parameter contributes to the overall quality of service?
- Which level the given property belongs to?
- Which aggregation method should be utilised to the given property?

The extended list of competency questions formulated may be found in Listing 8.3.

The definition of the competency questions, as well as elicited requirements, resulted in a set of concepts that are included in the information model.

Thus, the information model encompasses the following: concepts that may be assigned to various layers; various profiles (organizational, business users, process, service); requirements; decomposition and aggregation rules. The informal high-level view of the information model is presented in the following figure.

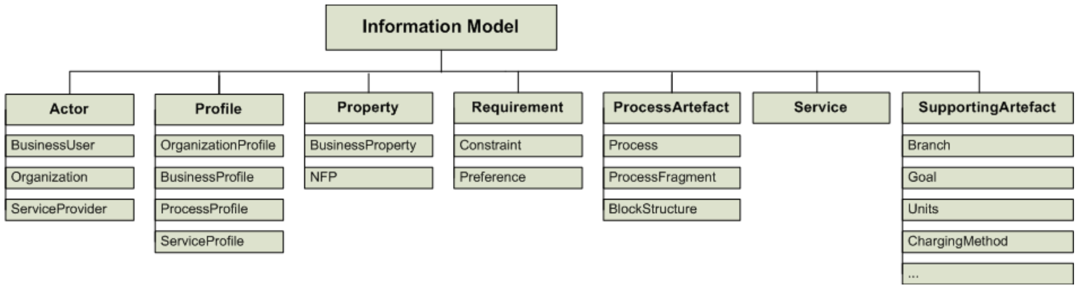


Figure 16 Information model overview

The following sections describe each of the elements of the proposed information model.

4.3.3 Properties representation

SWS e-marketplace participants need objective criteria to distinguish service substitutes. Following (Liu et al., 2004), it is not practical (or even possible) to come up with a standard model that can be used in all domains. Instead, the created model should be extensible in order to meet the needs of each market actor. Therefore, within this dissertation a general model that constitutes a kind of an upper ontology is proposed. Thus, it easy to further extend the model and apply it in any domain. For the illustration purposes, the model is defined with a limited number of criteria. However, adding new properties to the model will not alter the developed method or selection procedure (as is shown in Chapter 5).

The main concept is *Property* (see Figure 17). On the highest level properties have been divided into business oriented properties and non-functional properties. In addition, a group of process characteristics attributes has been distinguished in order to allow for easy categorization of aggregation as indicated later on in this subchapter.

Each property has a few attributes: *hasName* allowing for adding human readable property description, *value* allowing to assign a concrete value to the property and *relatesToLevel*, allowing to indicate which level the property in question relates to. The *Level* concept has been modelled with instances *OrganizationLevel*, *UserLevel*, *ServiceLevel* as well a subconcept *ProcessLevel* with instances *processSubLevel*, *taskSubLevel*, *processFragmentSublevel* have been added. The information to which level or sublevel the given property instance relates to allows for reasoning on appropriate decomposition or aggregation of values. The set of appropriate axioms makes sure that if the given property already defined within the information model is relevant only to a given level or sublevel, a user cannot attach it to different one, e.g.:

```
axiom allProcessAndTaskAndServiceLevelsSupported
     definedBy
?x memberOf {Cost, Availability, Owner, DataConfidentiality,
DataEncription, Authentication, Authorization, NonRepudiation} and
?x[relatesToLevel hasValue {processSubLevel,taskSubLevel, ServiceLevel} ].
```

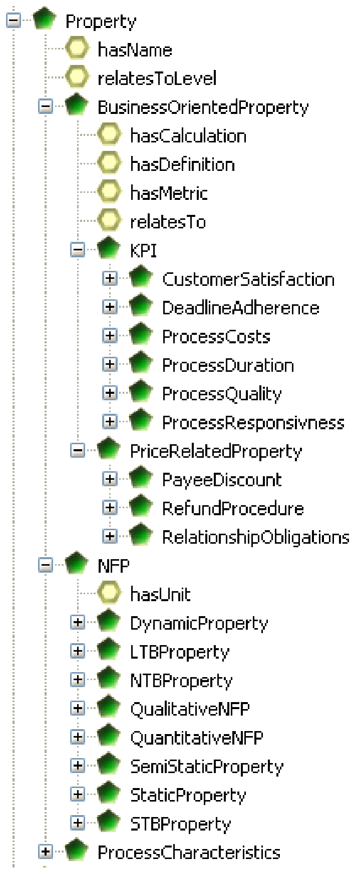


Figure 17 Information model – properties structure

When it comes to business oriented properties, a number of different information has been considered. As explained in (Fantini et al., 2007) within the strategy layer business people specify critical success factors (CSF) that define business goals for the next mid-long period. However, the CSF are too general to be applied automatically to the lower levels. Therefore, the rather strategic high-level CSF should be translated to mid-level operational performance measures, called key performance indicators (KPI) that are of interest for a business user. KPIs are monitored during the evaluation period and are used to control the achievement of CSF. A popular instrument for defining and managing CSF and KPI is the Balanced Scorecard introduced by Kaplan and Norton (Kaplan and Norton, 1996).

KPI in the context of processes are used for evaluating process effectiveness and process efficiency. (Schmelzer and Sesselmann, 2006) define five main categories which should be assessed for business processes: process duration, deadline adherence, process quality and process costs as indicators for process efficiency, and customer satisfaction as indicator for process effectiveness. These five categories provide only a framework for business analysts for defining KPI for each concrete business process in a certain business domain. For every business process, concrete KPI may be specified for each of these categories, and therefore the mentioned KPIs were modelled as concepts and not instances. In addition, for some domains there exist predefined KPI catalogues. An example is SCOR for the supply-chain domain. Examples for KPIs in the supply-chain domain are "the average time for processing a purchase order" (process duration) and "percentage of purchase orders that were processed successfully and in time" (deadline adherence). The mentioned concepts modelled in other ontology may easily import the model proposed in this dissertation, and thus, become the subconcepts of the *KPI* concept and therefore be easily used by the selection.

A KPI consists of a metric and a target value. For the business goal defined in natural langue as: "Increase customer satisfaction by reducing process duration" one could define a KPI "Average duration for processing a loan request < 3 days". When considering above-mentioned case the target value of "3 days" formulates a constraint on the duration of this particular business process. Stepping from the process level, one could additionally assign duration constraints on the task sublevel. If loan approval process is to be set as an example one can illustrate new constraint on the task sublevel as "average duration of risk assessment task < 1 day".

KPI will be used by business users to express their preferences on the process level. Therefore, the appropriate linkage between the KPIs and the lower level information they apply to, need to be modelled. This is done by using additional attributes *relatesTo* that indicates which property or properties on lower levels are affected or should be used to compute the value of KPI. The aggregation (computation) method is indicated by treating KPI concepts as subconcepts of Process Characteristics attributes.

The business-oriented properties focus not only on the KPIs by also on the price related properties that encompass not only the price itself, but also focus on other information such as (following (O'Sullivan et al., 2005)): conditions, refund procedure, negotiability, price customisation, relationship obligations, payee discount. Therefore, some additional concepts that can be extended by other ontologies were added.

The *NFP* concept stands for any non-functional property including the quality parameters that may be relevant to either a service, task or process. In Chapter 3, various NFP categories were discussed as well as various approaches and suggestions were given on which NFP should be considered in Web service composition. Their contribution has been taken up to determine the relevant categories for this work. For the needs of the selection mechanism, the properties are categorized taking into account:

- type – defining the type of value of the parameter i.e. dividing them into quantitative and qualitative ones. Quantitative parameters are measured on an ordinal, interval, or ratio scale, whereas qualitative variables are measured on nominal scale;
- volatility - represents static or dynamic nature of non-functional property;
- impact on perceived quality - it represents the way the discussed non-functional parameters affect the user perceived quality of service. Therefore, the division into Larger-the-better (LTB), smaller-the-better (STB) and Nominal-the-best (NTB) has been applied.

In addition, in order to make sure that adding new criteria does not change the underlying computational model used to evaluate SWS, the already mentioned categorization has been introduced: *AverageAttributes*, *CriticalPathAttributes*, *MinimalMaximalAttributes*, *ProbabilityBasedAttributes*, *SumAttributes* and *UserAssessmentBasedAttributes*. To each category appropriate methods of aggregation have been defined (see next section for details).

Within the NFP ontology, parameters have at least the following attributes: name (obligatory property), value and unit informing about the physical unit of the measure e.g. milliseconds for the response time.

Currently, the properties enumerated in Table 19 are included in the model. They constitute a blend of technical and more business oriented ones. These criteria may be easily extended with the domain-specific ones[47]. Each of the enumerated property has to be assigned to appropriate categories e.g.:

- duration is a subconcept of: *Performance* (subConcept of *QuantitativeProperty*), *STBProperty*, *DynamicProperty*, *CriticalPathAttribute*. In addition, it is also a subconcept of *TemporalDuration* concept coming from an external *TemporalOntology;*
- security is a subconcept of: *Qualitative*, *NTBProperty*, *StaticProperty*, *MinimalMaximalAttribute*.

Such a matrix like construction allows for reasoning and categorization of the properties as well as makes sure that the model is extensible because methods and decomposition or aggregation rules are associated mainly with categories and not specific properties. Therefore, each new property added or new ontology, need to fit into this categorization.

[47] It is shown during validation procedure in chapter 5.

Table 19 Non-functional properties within the proposed information model

Property	Definition and comments
Availability	Availability of a service is the probability that the service is accessible. It is expressed as percentage.
Compensation rate	Compensation rate of a service indicates percentage of the original execution price that will be refunded when the provider cannot honour the committed service or deliver the ordered commodity.
Cost	Cost understood as the amount of money that a requester has to pay to the provider to use a service.
Duration	Duration measures the expected delay in seconds between the moment when a request is sent and the moment when the service is rendered. It is the sum of the processing time and the transmission time.
Payment model	Captures the manner in which a requester can fulfil they payment obligations (O'Sullivan et al., 2002). It may be defined using the following attributes: hasCurrency - e.g. Euro and etc.; hasPaymentMethod - e.g. credit card; hasChargingFrequency - e.g. per Invocation.
Payment method	Independent of the payment model, it describes the payment method used - e.g. credit card.
Charging frequency	Independent of the payment model it describes the charging frequency of service e.g. per invocation.
Penalty rate	Indicates what percentage of the original price requesters need to pay to the provider when their want to cancel the committed service after the time out period for transaction to roll back is expired.
Reliability (Zeng et al., 2003)	It is the probability that a request is correctly responded within a maximum expected time frame. Reliability is a technical measure related to hardware and or software configuration of services and the network connections between the requesters and providers.
Reputation (user rating)	The reputation of a service is a measure of its trustworthiness. It mainly depends on end user's experiences of using the service. Different end users may have different opinions on the same service. Reputation value is defined as the average ranking given to the service by end users.
Security	The general concept encompassing: data confidentiality, data encryption, authentication, authorization and non-repudiation.
Throughput	It is the number of completed service requests over a time period.
Transaction support	Transaction support is used for maintaining data consistency. From the perspective of a requester, whether a service provides an undo procedure to rollback the service execution in certain period without any charges is an important factor that affects his/her choice. Transactional property is evaluated by two dimensions: whether undo procedure is supported and the time constraint on undo procedure.

4.3.4 Process structure, service and supporting artefacts

The developed information model needs also to allow expressing the artefacts such as a process and its elements as well as services. It has been decided to model the process using the block based approach. However, if the graph based representation would be required, the structure similar to the one in Business Process Modelling Ontology (BPMO) (Zhixian et al., 2007) could be utilised.

Following the block based approach resulted in the following elements belonging to the branch *ProcessArtefact* (see Figure 18):

- process (with attributes allowing to define multiple constraints and preferences, assign a process owner as well as a process profile);
- process fragment (with attributes allowing to define multiple constraints and assign process fragment to a process);
- task (with attributes allowing to define multiple constraints, assign a SWS goal to a task, as well as indicate which property is critical for the given task (e.g. roll-back or security)).

It has been decided to represent each process as a sequence of blocks. Therefore, the following concepts have been modelled: *Loop*, *XOR*, *OR*, *Parallel*, *Sequence*, *Task*. The mentioned elements allow representing the process along with its control flow. In addition, to each of these elements multiple constraints can be added. Moreover, the loops structures allow for associating the maximum number of iterations.

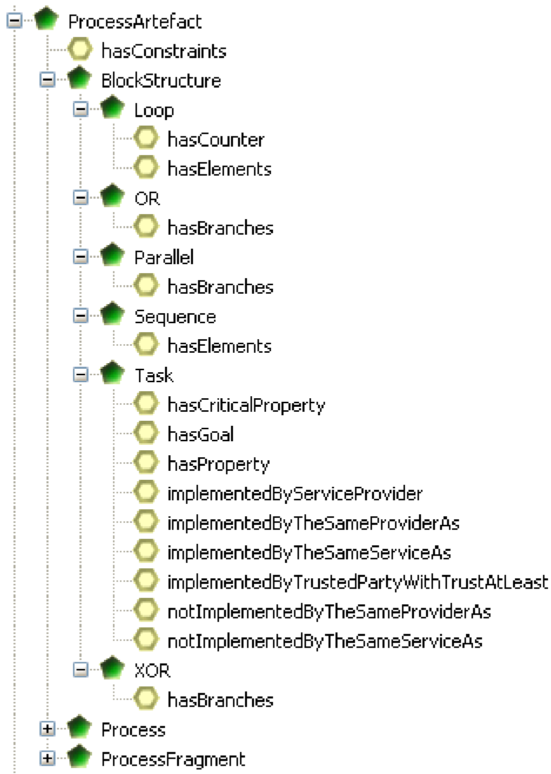


Figure 18 Block structures allowing for process representation

If the process model cannot be represented as structured elements i.e. as block based structure, then the process model structure needs to be transformed, as e.g. shown in (Aalst et al., 2003). However, the detailed discussion of this issue is out of scope of this dissertation.

In addition to the above-mentioned process artefacts, a concept *Service* also has been modelled. It has attributes allowing to define multiple constraints, assign a SWS goal as well as a service profile.

A service may encompass more than one operation, but in case of SWS each of the operations becomes the separate service. In order to deal with this scenario and the case of statefull services, as emphasised by (Ardagna and Pernici, 2007), or with situations like e.g. the equipment bought from company X must be delivered by services of company Y, an additional attribute *needsToBeUsedTogetherWith* has been modelled.

Another situation that often occurs on the market is that providers offer their services in a bundle i.e. if two services are used (or bought) together, the price is different or there are some other benefits. Therefore, an additional attribute has been added to the service: *belongsToBundle*. It allows connecting it with a concept *Bundle* being also a part of the information model. Additional information is added to the bundle concept that represents the price of the bundle as well as additional attribute allowing to store information about the special benefits. However, within this work, the focus is assigned only to the price related discounts.

In addition, the user may point out that task A and B need to be implemented by the same service provider (or service) without specifying which service provider (or service) it should be. Or, that the task A and B cannot be fulfilled by the same provider (or service), because of some business rule. It is addressed by using attributes: *implementedByTheSameProviderAs* or *implementedByTheSameService* as well as *notImplementedByTheSameProvider* or *notImplementedByTheSameService*.

In addition, a number of supporting artefacts needed to be defined, such as inter alia (see Figure 19):

- branch for the needs of representing process structure with the sub concept conditional branch allowing for associating the expected frequency of execution of conditional branch,
- bundle – allowing to store information about services, service providers as well as offered discount,
- SLA with pointer to a service, service provider and client as well as allowing to attach multiple SLO – in order to allow to express the business and SLA related aspects of a service. The SLA concept is generic in the sense that it may become a super concept for any SLA ontology modelled for the needs of SWS,
- horizon – allowing to indicate whether the process is to be run in long or short term relationship,
- goal – representing the functionality offered by a service or a task,
- unit – allowing to represent various units of measures e.g. percentage, milliseconds, seconds etc.

- trusted party – allowing to add a trust level to a specific organizations (see 4.3.7 for details).

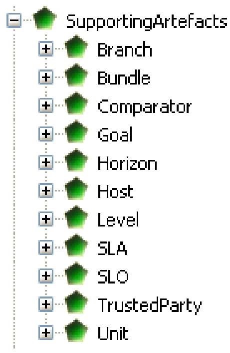


Figure 19 Supporting artefacts

4.3.5 Aggregation techniques

The above-mentioned properties may be applied either to services or to more complex structures e.g. tasks, process fragments or processes. Only in case of tasks, service attributes values correspond directly to attributes value of a task. In case of process fragments and processes, attributes values need to be aggregated.

Therefore, once a process model is transformed into a sequence of composition patterns, then, a recursive approach is used to aggregate properties on the level of identified composition patterns. In addition, probabilities of flow in conditional structures are considered during computation.

Table 20 provides the aggregation function for the computation of the quality of process taking into account categorization of attributes (see Figure 20), introduced in the information model presented in the previous sections as well as workflow structures.

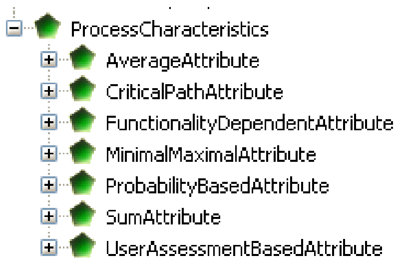


Figure 20 Process Characteristics categories

As indicated in the table, instead of defining aggregation method to each NFP as done in other approaches (see Chapter 3), the aggregation methods are assigned to process characteristics, which in turn are linked to properties.

The symbol p_i in the table denotes the probability of execution of a given branch, k denotes a maximum loop count.

In case of sum attributes, their values are summed up, as every started task is relevant. For parallel split cases (XOR, OR), the maximum value possible is relevant or, the expected value of the attribute is counted depending whether the values of probability of branches execution are available.

Table 20 Aggregation techniques (own study)

Criteria	Sequence	AND	XOR	OR	Loop
Sum Attributes (sa)	$\sum_{i=1}^{N} q_{price}(s_i)$	$\sum_{i=1}^{N} q_{price}(s_i)$	$\sum_{i=1}^{N} p_i\, q_{price}(s_i)$ or $\max q_{price}(s_i)$ if no probability is given	$\sum_{i=1}^{N} p_i\, q_{price}(s_i)$ or the sum of all if prob. is not given	$k * q_{price}(s_i)$
Critical Path Attributes (cpa)	$\sum_{i=1}^{N} q_{duration}(s_i)$	$\sum_{i=1}^{N} q_{duration}(s^{CPA}{}_i)$ (only those being on the critical path are counted)	$\sum_{i=1}^{N} p_i\, q_{duration}(s_i)$ or $\max q_{duration}(s_i)$ is taken if no probability is given		$k * q_{duration}(s_i)$
Probability Based Attributes (pba)	$\prod_{i=1}^{N}(q_{reliabiltiy}(s_i))$	$\prod_{i=1}^{N}(q_{reliabiltiy}(s_i))$			$\prod_{i=1}^{k}(q_{reliabiltiy}(s_i))$
Average Attributes (aba)	$\frac{1}{N}\sum_{i=1}^{N} q_{rep}(s_i)$	$\frac{1}{N}\sum_{i=1}^{N} q_{rep}(s_i)$	$\sum_{i=1}^{N} p_i\, q_{REP}(s_i)$ i.e. expected value is taken		$q_{rep}(s_i)$
Minimal Maximal Attributes (mma)	*Min* or *Max* depending on property character				
Functionali-ty depen-dent attri-butes (fda)	$MIN\{\forall_{i=1..n} f(s_i)^{z_i}\}$ z_i is equal to 1 if service s_i is a critical service in the process when it comes to the considered functionality, or 0 otherwise				

In case of critical path attributes, the critical path concept is considered. Therefore, the aggregation algorithm first identifies the critical path that takes the longest time in the workflow and then, for that path, the individual values of the tasks are summed up.

In case of the probability-based attributes, a product of probability values of each atomic service in the composition is counted.

In case of the Minimal Maximal Attributes, aggregation considers the service that is offering the weakest (lowest) value relevant for the whole composition. Whether the minimum or maximum is taken depends on the attribute character. In case of LTB properties minimum value is taken, in case of STB maximum.

When it comes to average attributes values, the average value is considered.

In addition, another group of properties has been distinguished, i.e. functionality dependent attributes. The motivation is as follows: we cannot say, that the process does not offer roll back because one service, e.g. reading some data from database, cannot be undone. On the other hand, even if all services offer roll-back apart of one making reservation, then the entire process does not offer roll-back support. Therefore, the minimum is taken of $f(s_i)^{zi}$ where function z_i is equal to 1 if service s_i is a critical service when it comes to the functionality in question, or 0 otherwise. If z_i=0, i.e. service s_i is not a critical service, then $f(s_i)^{zi}$ equals 1 and hence the given functionality or its lack of service s_i will not affect the value of this functionality for the entire process.

One more group has been distinguished in the process characteristics class and not included in the table, namely *UserAssessmentBasedAttribute* representing a group of concepts, which values cannot be computed automatically, but require user assessment or in general human involvement in the process. An example may be user satisfaction.

4.3.6 Requirements and their representation

Business users specify requirements considering properties of their processes. One distinguishes between constraints and preferences as indicated in Chapter 3. A constraint can be defined for three levels: process model, process fragment and a single task. A process model potentially can have several constraints on all of these three levels. If there are several constraints, they all have to be satisfied ("AND"). A constraint definition thus should contain:

- a reference to the non-functional property it wants to restrict,
- a constraint value which should be met,
- the unit of the constraint value (e.g., seconds),
- a comparator operator or function which determines whether the constraint is satisfied.

In order to allow for easy specification of constraints each property mentioned in Section 4.3.3 has a value attribute. It facilitates the process of automatic constraints processing. In addition, within information model, a set of instances for the needs of definition of constraints and preferences has been added.

Moreover, one should distinguish between fuzzy and stringent constraints. Stringent constraints (hard or critical constraints) need to be met otherwise an error message is returned to a user. The fuzzy constraint should be understood as "if possible try to achieve it" i.e. the selection is performed even if the constraint cannot be entirely met by available services.

The constraint concept allows to apply constraints to both quantitative as well as qualitative properties.

In case of preferences, the general preference is specified as follows:

- a reference to the property that is important,
- a priority assigned to the property in question.

In addition, a subconcept of *PropertyPreference* has been created namely *NTBPropertyPreference* allowing to model a special kind of preferences regarding the NTB properties. This concept includes additional attributes i.e. an operator allowing ordering the set of values we want to optimize.

Both preferences and constraints can be attached to various artefacts at various levels or become a part of the profile.

4.3.7 Actors and Profiles

The profile concept included in the information model allows to represent the expectations and requirements of certain entities or to provide rich characteristics of other entities (see Figure 21).

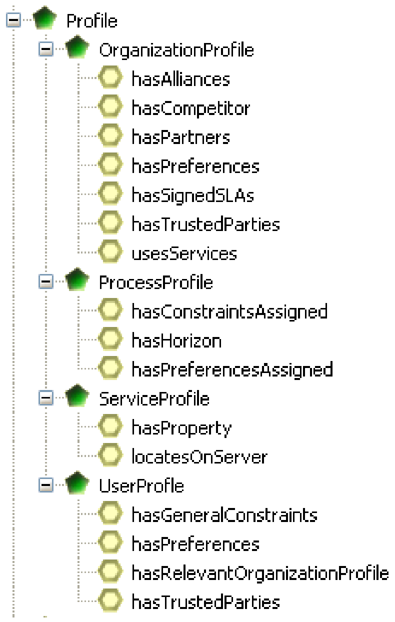


Figure 21 Profiles

The distinguished profiles are shortly described below:

- organization context – it is quite constant in time and constitutes a common base for at least a few, if not all, service requests. It may be applied to more that one user context (profile),
- user context – describing business users context and their general preferences,
- process model context– constraints and preferences relevant to the process, process fragment or a task, additional rules that should be fulfilled,
- service context - provides service profiles and characteristics.

The organization profile is quite constant in time and do not change very often. The organization context includes information on business partners the organization collaborates with, forms alliances or information on competitors. In addition, the organization profile includes information on the trusted parties along with the trust level assigned to each party. This information is very useful during service selection as then only reliable partners are selected and not e.g. services of competitors.

In addition, the information on already signed SLA is stored (*hasSLA*) that points to specific SLA instance of concept modelled within the supporting artefacts group.

The collaborators context informs us with which providers are preferred by an organization. To each of the providers, the organization may add the level of trust. Trust is relationship between two parties such that one party believes the other one does what it promises to do. Trust constitutes the highest layer in the Semantic Web architecture (Galizia et al., 2007). However, only few approaches dealing with SWS (as well as Semantic Web applications) really provide a methodology to represent it[48]. (Galizia et al., 2007) argue that the main difficulty lies in the context-based nature of the trust related issues as the same user may have different trust policies in different contexts.

In SWS as well as social networks, trust is a central issue. In both cases, interactions take place whenever there is trustworthiness. Business people do not perform transactions with companies they do not trust. Therefore, trusted parties concept need to be modelled. However, the use of static contractual agreements prevents dynamic selection among competing service providers based on changes in customer profile or context or based on the performance profile of individual service providers. In addition, it would be hard if not impossible to sign a contract with all service providers. If e.g. it is critical for our company to send a packet overseas and our delivery company is overloaded and we would assume that we deal only with providers that we trust, the system would not inform us that there exists any other possibility.

To address this issue, the use of the transitive trust relationship mechanism is proposed. The mechanism should be similar to the one implemented on the Semantic Web and used to evaluate different sources of information. Transitive trust relationships may be summarized as: if A trusts B and B trusts C then A trusts C. It allows building a web of trust, reducing one-one relationships and easing collaboration, simultaneously broadening available resources. Therefore, the developed model may take into account transitive trust relationship model that is used in the Semantic Web and assign appropriate weights to the network of trust values.

The organization profile encompasses also the general preferences the organization follows. For instance, it may state that the security is of utmost importance. The organization preferences may depend on the domain it operates in etc. In addition, one organization may have several profiles.

The organization profile also states which services (and thus service providers) the organization is using on regular basis. Therefore, a service portfolio context provides information on which services the organization is using and on what terms. This

[48] The most common approaches for describing Semantic Web services i.e. WSMO and OWL-S do not provide exhaustive means for trust annotation.

information may be used during selection. For instance, if during service selection we consider utilising a service that the organization is using anyway and pays monthly fees no matter how many invocations there are, then the cost of the service is different than the one stated in the service description.

In turn, a business user is usually interested in a number of services coming from trusted parties, general preferences of their own as well as some constraints. For instance, a business user may be price sensitive, therefore during the selection an appropriate rule should try to filter out all costly services while selecting a service.

The process context encompasses requirements relevant to a given process, e.g. a predicted repeatability of the process executions. The repeatability of the process executions is relevant when the long or only short-term relationships should be considered. If we are going to execute the process only once in a nearest feature, then we are not really interested in the long-term prognosis of service behaviour. If, however, we are more interested in long-term process that is to be executed numerous times in the future, then we should also take into account the long-term prognosis of service behaviour and be more cautious when it comes to the selection and signing the binding agreements.

The requirements relevant to the process are those that are assigned specifically to the process itself. They consist of preferences as well as constraints. Each of them may be applied to the entire process, to a specific process block or a task.

From the service selection point of view, the most important service context that should be taken into account is the non-functional characteristic of a service as well as additional dependencies that exist between services.

In the proposed framework, when it comes to the versatility, three types of criteria are distinguished: static, semi-static and dynamic. For the static properties, it is assumed that the service providers advertise those values and they are verified by the SWS e-marketplace mechanisms. The service providers use the information model to specify the static properties (they are obliged to do that by the SWS e-marketplace).

Semi-static and dynamic properties are collected in a fair, open and objective manner via already mentioned in Chapter 3 dynamic service profiling mechanism that relies on active monitoring and user's feedback as well as contracted service level agreements (Abramowicz et al., 2008c, Kuropka et al., 2008). The profiling system relies on multiple sources of information and provides up-to-date information on services taking into account the horizon of the prognosis if needed. It is also able to verify SLA violations etc. and thus evaluate the trustworthiness of a certain service or service providers. The service profile used by the selection system may be easily adjusted to suit its needs so as the information provided by a profile matches the created information model required by the selection mechanism in question. Thus, dynamic service profiling mechanism provides the desired values in the expected form and uses the information model presented in the previous section. Although the original DSP mechanism implemented within the ASG system provided the service profiles in the form of XML files, it is relatively easy to modify it to provide the profile as instances of the ontology model.

4.3.8 Developed ontology – modelling decisions and summary

It has been decided to model the ontology (Figure 22) using the WSML notation.

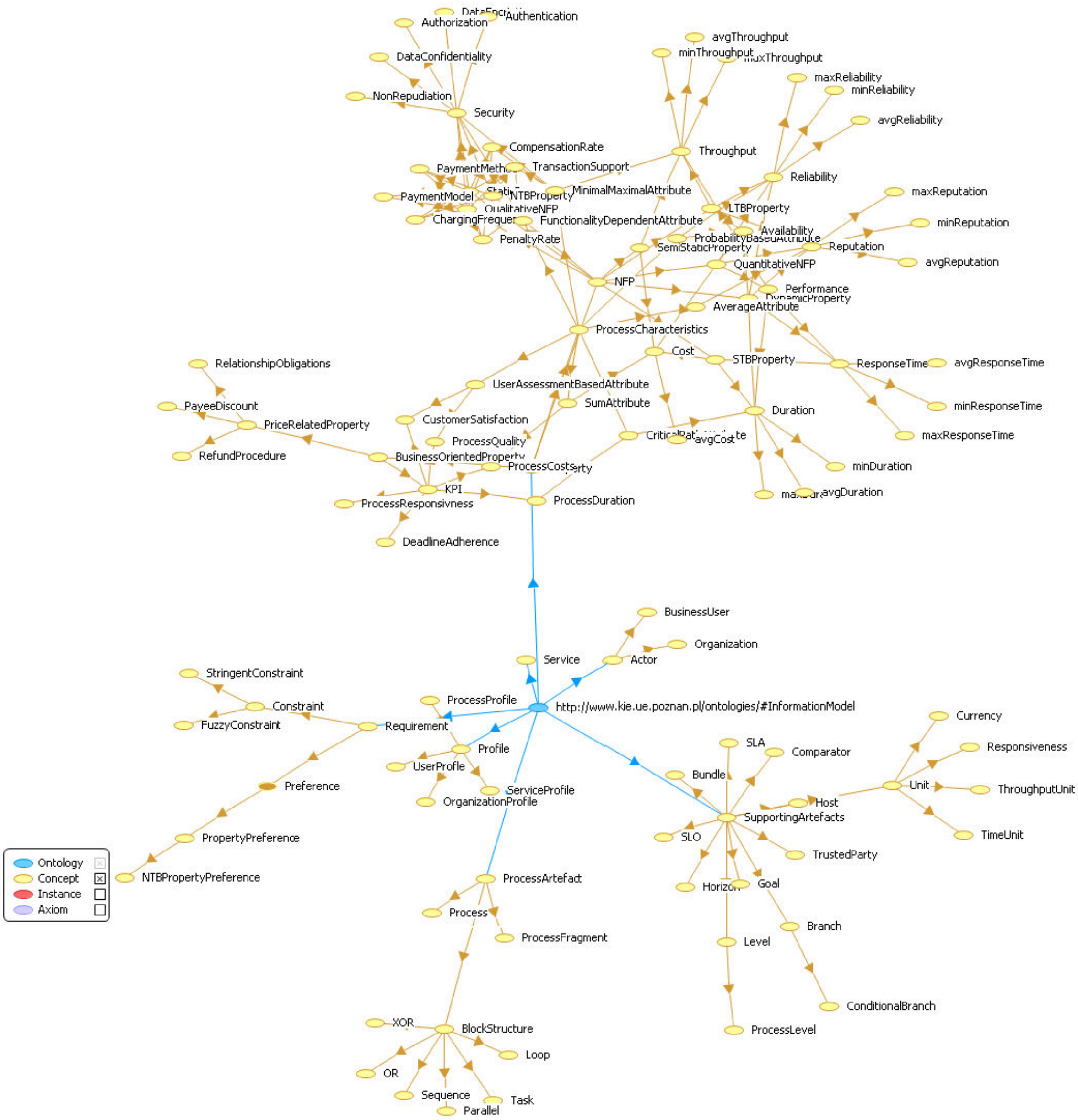


Figure 22 Information model represented using WSMO studio

The decision was based on the performed short survey regarding the availability of the efficient reasoners (Bishop and Fischer, 2008) as well as ready-to-use ontologies that could easily import the information model created. The presented model can be easily expressed (even automatically translated) in any other of the popular notations used to model ontologies (e.g. OWL, RDF).

In the ontology the appropriate axioms and cardinality restrictions make sure that the interrelations between various concepts and properties are correctly modelled e.g. that each service has at least one goal and profile assigned.

The requirements as well as a set of competencies questions influenced the modelling decisions taken during the ontology development. For the needs of further analysis and selection mechanism, the type of properties has been modelled as classes and properties became subclasses of them. The other possibility would be to model this in the form of attributes. However, this approach would not offer required flexibility, as each time a user would create an instance e.g. of reputation, they would have to assign an instance of a type to it. Moreover, it would be impossible to assign a certain type property to the class definition, as classes cannot have assigned instances as values. In addition, it has been decided to model the properties as classes not instances. It allows for easy assignment of different values of properties to different artefacts at the same time the information from the class hierarchy may be efficiently utilised to perform various operations (including aggregation) on the attribute values.

The developed information model has 110 different concepts and around 70 instances along with a set of axioms and relations. The excerpt from the ontology code is included in the listing.

4.4 Selection technique of the proposed mechanism

This section focuses on the selection mechanism itself operating on the information model defined within the previous section.

The general activity diagram depicting the steps performed by the mechanism is depicted in Figure 23.

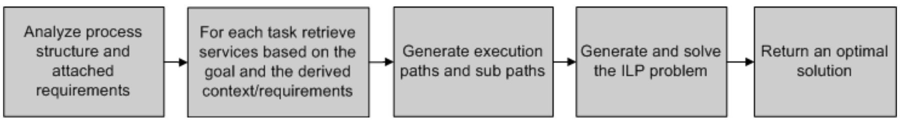


Figure 23 Activity diagram of the selection mechanism

First, the selection component analyzes the process model and requirements attached to it (those included in the process model itself as well as in both user profile and organization profile). Then, it decomposes the constraints that may be derived from the profile as well as the process global ones stated in the process model. Then, for each task, based on the goal assigned to each task and context description (all identified constraints) the relevant services are retrieved using the reasoning infrastructure. In this way, a list of possible services for the execution of each task is created.

Then, the selection mechanism generates all possible execution paths with the appropriate probability of executing the given path assigned. For each execution path the sub paths are created. Within the next step, the ILP problem is generated so that the optimal solution may be found.

For the needs of further discussion, let us formalize the selection model and introduce the following.

The process model (PM) must contain a set of tasks where the number *a* represents the total number of tasks in the process model.

$$T = \{t_1, ..., t_a\} \tag{4}$$

The output of matchmaking process is a set of service candidates where the variable b represents the number of all candidates.

$$S = \{s_1, ..., s_b\} \tag{5}$$

It is presumed that a task potentially requires a different type of functionality when compared to other tasks. Consequently, available services will provide a particular functionality and thus may not be suitable to perform different tasks. If the case occurs that a service can serve two different types of functionality, a candidate identified for one and the other task is counted twice. In addition, if a service creates a bundle with another service and these two services are found to be relevant to the process model, they also appear twice in the returned set with additional attributes assigned. Therefore, it is assumed, that the outcome of a discovery process results in a set of candidate C, consisting of sets S_i, each holding the candidates for a particular task $t_i \in T$.

$$\begin{aligned} &C = \{S_1, ..., S_a\} \\ &S_i = \{s_{i1}, ..., s_{ib_i}\} \\ &i = 1, ..., a \end{aligned} \tag{6}$$

In this definition, b_i denotes the number of candidates found for a task *i*.

In addition to the modelled sets of tasks and candidates, the structure of the process model is also relevant for the proper aggregation of different characteristics. Thus, the problem model must involve a model of the composition structure based on the composition patterns as introduced earlier.

As the selection criteria, the characteristics encompassing both technical as well as business properties are used. For identifying different characteristics, which is necessary when their values are used in optimisation statements, a number from 1 to *p* is used, with *p* denoting the total number of characteristics that may be considered.

Therefore, formally the selection problem may be defined as finding an optimal set of pairs:

$$\{<t_1, s_{i1}>, <t_2, s_{i2}>, \ldots, <t_n, s_{in}>\} \tag{7}$$

such as the value of the scoring function is maximal.

Taking into account the selection model presented above, the possible combinations result from the Cartesian product of the candidate sets in

$$\text{C}: S_1 \times \ldots \times S_a \,. \tag{8}$$

This product results in a set of a-tuples where the number of elements represents the number of possible combinations:

$$| S_1 \times \ldots \times S_a | = | S_1 | \bullet \ldots \bullet | S_a | = \prod_{i=1}^{a} | S_i | \tag{9}$$

This indicates the resulting exponential effort for a rising number of tasks, if an algorithm potentially evaluates all possible combinations. In order to address this issue in the developed model, the first step of the selection narrows down efficiently the number of possible candidates (as indicated later the complexity of this step is polynomial). In addition, during the second step, the heuristic algorithms may be used.

The following subsections discuss in details each of the steps included in the activity diagram.

4.4.1 Decomposition and filtering

Based on the information model, the following decomposition rules have been defined in order to formulate the appropriate filtering criteria for the set of relevant services.

As a result of analysis of the process model description, all constraints applied to the organization, business user or process model are propagated to the task level.

First, it is checked which properties values are restricted (e.g. process duration, mean execution time, security level). If the constraints are applied to properties that directly relate to the task or service level, then new constraints are formulated and attached to all. In addition, decomposition needs to take into account the sum of global and local constraints, i.e. if the constraint already applied to the task level is more stringent that the one on the process, then the global one should be discarded. Thus, for instance, the constraint defined on the process level "process duration less than 10 minutes" imposes constraint that each task should execute in less than 10 minutes (i.e. execution time ≤ 10 minutes). However, if a given task has already attached constraint "execution time ≤ 1 minute", then no additional constraint is formulated.

In case when the constraint relates to the property that cannot be directly mapped to the task level, then the rules embedded in the information model should be used to identify the relevant properties which values should be restricted. For instance in case of KPI using the *relatesTo* attribute, together with *relatesToLevel* and taking into account to which *ProcessCharacteristics* it belongs to, the appropriate decomposition plan can be applied. If no such rules can be found, the given constraint is discarded or a user may be asked to indicate the decomposition plan.

The selection also checks what are the preferences regarding partners and trusted parties. The appropriate rule is applied in order to define the value of the trust parameter for each provider whose services are identified to be relevant to the process. However, the value of the trust property depends on the preferences of the user. For instance, if users want to cooperate only with service providers they directly trust, value of trust for all other providers would equal 0. If, however, they are willing to start collaboration with new providers that some of their partners already trust, then the transitive trust relationship (Mezzeti, 2003) may be used to compute the level of potential trust.

Also taking into account a business user profile and preferences defined, the weights are assigned to each property of interest. Thus, a vector of weights is created and it is used to represent the preference values. The preference vector holds weights value assigned to each property denoted by the index n (n=1,…, p).

Meanwhile, the dynamic service profiling is used in order to obtain the profiles of the services as well as to perform the filtering. As the service profiles are created on request, the horizon of the prognosis can be taken into account. Therefore, service profiles are either created for the needs of short-term relationship or a long-term one, depending on the information from the process profile. In addition, it is assumed that the dynamic service profiling mechanism during profile creation performs analysis of the charging model used and the cost of a service included in the returned profile is unified so that the values can be compared. The work on this issue is conducted by e.g. (Zyskowski, 2010). In addition, the portfolio of already used services within organization is taken into account, in order to check, what is the real cost of potential services to be used within a process. For instance, one of the services may be already utilised within the organization per subscription basis. In that case, the actual cost of using this service will be different from that provided within a service profile.

Once the constraints and profiles are gathered, services that do not meet the defined constraints may be filtered out. The filtering is performed on the ontological level by using reasoner to return only services meeting specific criteria.

Some constraints on the task level that were used to discard irrelevant services (e.g. given service provider, appropriate security level, stringent conditions on execution time) will not become a part of the ILP problem.

In addition, during the filtering process, the fuzzy constraints are introduced. Following (Cock et al., 2007), let us consider the following situation. The constraint has been defined by a user "execution time should be at most 20 ms and the availability should be at least 90%" and attached to a given task. However, during filtering no such service with the required functionality is available. Thus, one can imagine that a service doing the job in 21 ms with availability of 96% is also acceptable, in fact in this case it might even be preferable. Therefore, during filtering the conditions (defined as fuzzy ones) are relaxed. Relaxation value is dynamically computed as:

$$\delta = \frac{p * C}{100} \tag{10}$$

where p is a certain percentage and C stands for the value of a constraint. The underlying idea is to link the size of the fuzzy margin to the absolute value expressed in the constraint. E.g. if a user is looking for an execution time of at most 65 ms then 70 ms might still be acceptable. However, when he is looking for 5 ms, then 10 ms is probably not going to be acceptable. Higher p-values typically allow for a larger difference between the baseline and fuzzy approach. In case p=0 both approaches coincide. (Cock et al., 2007) postulate that the value of 20% should meet business expectations. Therefore, during simulation, the value of p parameter has been set to 20.

Therefore, the following activities are performed during the first step of the selection:

- profiles and requirements are analysed and constraints decomposed,
- relevant services based on goal/capability are retrieved using the reasoner or service discovery engine (e.g. WSMX),
- service profiles are generated by the dynamic service profiling mechanism,
- retrieved services are evaluated based on constraints (also the decomposed ones) using the reasoning infrastructure and those not meeting the requirements are discarded,
- a list of possible services to each task is created.

Therefore, the outcome of this stage is the list of services attached to each task.

The role of this stage cannot be overestimated. It allows limiting the set of possible services that are to be taken into account while looking for the optimal configuration. By taking into account the business context (e.g., partner, competitors) and other qualitative features services not meeting business expectations can be filtered out. In addition, thanks to the decomposition rules, the constraint defined on the more global level can be also taken into account.

Due to the computational complexity of the algorithms used, the less possibilities the better (the more efficient the algorithm is). Therefore, the proposed approach limits the possible set that should be checked. Whereas the algorithm has the exponential complexity, decomposition has only polynomial time complexity.

4.4.2 Process structure decomposition

As an input to the next step, the following artefacts are given: process model definition along with the process-specific requirements and constraints as well as a user and organization profile (as described in the previous subchapter).

As already discussed within the information model, the following workflow patterns and structures are considered: sequence, parallel, XOR, OR, Loop. The above mentioned constructs are the most often used within a business process model (zur

Muehlen and Recker, 2008). In addition, it is assumed that a composed service is characterized by a single initial task and a single end task and that task composition follows a block structure so that in particular only structured loops can be specified i.e. loops with only one entry and exit point. In the following, it is assumed that cycles are unfolded. However, if a process would contain loops, the appropriate technique to transform it to acyclic one would first need to be applied. The most common technique to perform such transformation, is the analysis of the log files from the previous executions in order to determine the average number of times that each loop is taken. Then the elements between the beginning and the end of the loop are copied as many times as required (as many times as the loop is executed on average). In case when no execution log files are available, then the heuristic models need to be applied and in case when no expert knowledge is available, then the random number should be selected and updated as soon as some additional knowledge will be available.

The approach starts from the work presented in (Zeng et al., 2004), (Ardagna and Pernici, 2005), (Ardagna and Pernici, 2007) as well as (Yu et al., 2007).

In order to precisely define the created model, the two concepts, already mentioned in Chapter 3, namely execution path and execution plan need to be more formally defined. Inspired by (Zeng et al., 2003), an execution path [e_1, e_2, …, e_n] is a sequence of elements consisting of tasks (t) and parallel structures (AND), such that e_1 is the first element of the process (usually an initial task), e_n is the last element of a process (usually a final task) and for every element e_i for i>1 and i<n, the following holds:

- e_i is a direct successor[49] of one of the elements in [e_1, …,e_{i-1}],
- e_i is not a direct successor of any of the elements in [e_{i+1}, …, e_n],
- there is no element e_j in [e_1, …, e_{i-1}] such that e_j and e_i belong to two alternative branches of the process.

If the execution paths are unfolded then they become a set of tasks {t_1, t_2, …, t_a} such that t_1 is the initial task, t_n is the final task and no t_i, t_j belong to alternative branches.

Thus, the goal is to create the routes from start to end, which includes only one branch in each conditional operation but all branches in parallel operations. Each execution path has then a probability assigned, which is the product of all probabilities for all conditional branches selected in the route. In case of OR, the probability is not assigned to each branch, but to their combination.

It needs to be pointed out that a business process has a finite number of execution paths. If a business process contains conditional branching, then it has multiple execution paths. Each execution path represents then a sequence of elements to

[49] The above definition relies on the concept of direct successor of an element or a task. A basic task t_y is a direct successor of another basic task t_x if there is a sequence of adjacent transitions going from t_x to t_y without traversing any other basic task. Two transitions are adjacent if the target state of one is the source state of the other.

complete a composite service execution. Execution paths are denoted by ep_k so k denotes number of execution paths identified within a process model.

The execution paths can encompass parallel sequences. Therefore, the notion of a sub path (also called sequential path) is introduced. A sub path of an execution path ep_k is a sequence of tasks $[t_1, t_2, \ldots, t_x]$, $\forall_i t_i \in ep_k$, from the begin to the end task which does not contain any parallel sequences. A sub path is indexed by m and denoted by sp^k_m.

An example of execution paths and sub paths in an exemplary process model is shown in the following figure.

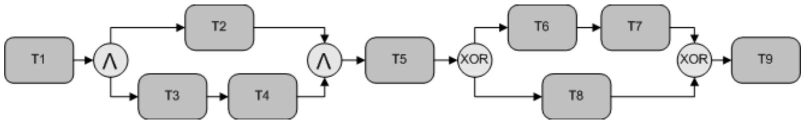


Figure 24 Process model (1)

The considered process has one parallel structure with two branches and one XOR structure with two conditional branches. Thus, within this process, two execution paths may be identified, each including only one of the conditional branches, but two parallel branches. This is shown in the figure below.

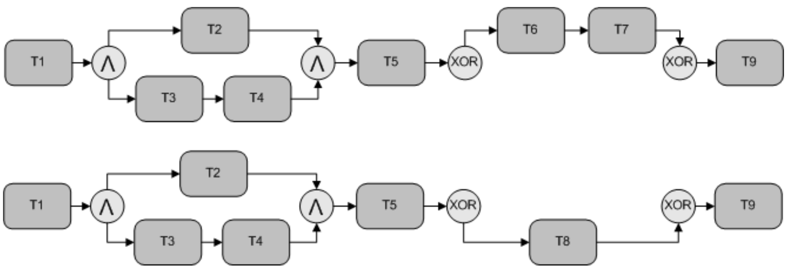


Figure 25 Execution paths within the process model (2)

In turn, each of the execution paths encompass two sub paths as depicted in the two following figure. The execution path 1 has sub paths (1) and (2). Sub paths (3) and (4) belong to the second execution path.

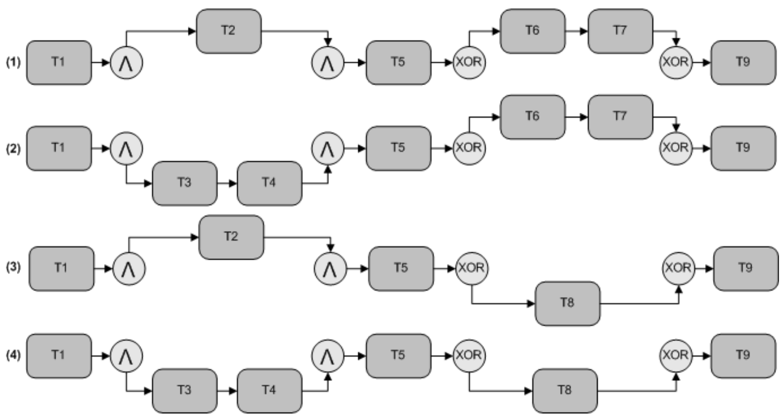


Figure 26 Sub paths of the process model (3)

Note that the critical path of an execution path ep_k is the sub path, which corresponds to the highest execution time of the execution path. Therefore, the critical path attributes value may be computed as maximal sum value for all sub paths:

$$CPA(ep_l) = \max_{sp_m^k \in ep_k} \sum_{t_i \in sp_m^k} cpa_i \tag{11}$$

where cpa_i indicates the value of a critical path algorithm property of a given task i.

In turn, following (Zeng et al., 2003), a set of pairs $\{ \langle t_1, s_{1j} \rangle, \langle t_2, s_{2j} \rangle, \ldots, \langle t_a, s_{aj} \rangle \}$ is an execution plan of an execution path, if and only if:

- $\{t_1, t_2, \ldots, t_a\}$ is the set of tasks in PM
- for each 2-tuple $\langle t_i, s_{ij} \rangle$, service s_{ij} is assigned to SWS to execution of task t_i.

Thus, an execution plan of an execution path ep_k is a set of ordered couples (t_i, s_{ij}) indicating that task t_i included in ep_k is executed by a given Semantic Web service s_{ij}. Execution plans are indexed by *l* and denoted as ep^l_k.

The global plan is a set of ordered couples (t_i, s_{ij}) which associates every task t_i to a given SWS (s_{ij}) and satisfies local and global constraints for all execution paths.

The set of execution paths of a process model identifies all the possible execution scenarios of the process. Within the proposed solution, the model does not optimize all execution paths separately as this would cause non-optimality of the discovered solution as indicated in Chapter 3. Instead, all execution paths are optimized at the same time and thus global constraints can be guaranteed whichever scenario will actually take place during execution.

4.4.3 ILP problem formulation

The output of previous steps are: execution paths and sub paths with assigned probability as well as a set of constraints assigned to various process elements as well as a vector of preferences assigning weights to each property that should be considered together with the set of services and their profiles.

The next steps encompass: generating variables, all constraints and objective function, so that the ILP problem may be formulated in a standard form:

max objective_function F

subject to

set_of_constraints

The ILP model does not require the generation of all possible execution plans. If there are *a* tasks and there are *m* potential SWS for each task, then the total number of execution plans is m^a. Therefore, such an approach would be impractical for large-scale processes, where both the number of tasks and the number of candidate SWS are large.

As already mentioned, three inputs are required by ILP: variables, objective function and constraints on the variables, where both the objective function and constraints must be linear. ILP attempts to maximize or minimize the value of objective function by adjusting the values of variables based on the constraints. The output of ILP is the maximum (or minimum) value of the objective function as well as the values of variables at this point. The global planner will therefore select the execution plan of the entire process model, which has the maximal value of the objective function i.e. max F. The objective function is discussed in more detail further in this section.

In order to utilize the ILP method to select and optimal execution plan, the selection of an optimal execution plan needs to be modelled as ILP problem. The variables of the ILP problem are y_{ij} representing the participation of service s_{ij} in the selected execution plan of a process. The value of each variable y_{ij} is 1 if service s_{ij} is selected and 0 otherwise. Thus, *b* variables are created.

The next issue is the definition of constraints on the variables of the ILP problem. T is the set of all tasks within a process. For each task t_i, there is a set of SWS i.e. S_i that can be assigned to this task, but in the end, for each task t_i, only one SWS should be selected. Given that y_{ij} =0 or =1 denote the participation of SWS s_{ij} in the selected plan, this latter fact is captured by:

$$\underset{i \in T}{\forall} \sum_{j \in S_i} y_{ij} = 1 \tag{12}$$

This constraints family (12) guarantees that every task is associated to exactly one Semantic Web service (i.e. for every i only one variable y_{ij} is set to 1).

$$\underset{cpa}{\forall} \ \underset{i \in T}{\forall} \sum_{j \in S_i} cpa_{ij} y_{ij} = cpa_i \tag{13}$$

Constraints family (13) expresses the value of critical path attribute of every task in term of the critical path attribute value of the selected service (note that for (12) only one service is selected and hence the critical path attribute of task is given by the selected service critical path attribute value). Constraints family (13) is generated for each critical path attribute considered.

$$\underset{cpa}{\forall} \ \underset{ep_k}{\forall} \ \underset{sp_m^k \in ep_k}{\forall} \sum_{i \in sp_m^k} cpa_i \le cpa_k \tag{14}$$

Constraints family (14) evaluates the critical path attribute value of every execution path as the maximum attribute value over the set of sub paths of the execution path. Indeed, the maximum value v_{max} of a set V is defined as the value in the set that:

$$\underset{v \in V}{\forall} v \le v_{\max} \tag{15}$$

The variable cpa_k is generated for each execution path. Thus, the type of the problem becomes Mixed Integer Linear Programming.

$$\underset{sa}{\forall} \ \underset{ep_k}{\forall} \sum_{i \in ep_k} \sum_{j \in S_i} sa_{ij} * y_{ij} = sa_k \tag{16}$$

Constraints families (16) express execution path ep_k value of sum attributes. Constraints family (16) is generated for each sum attribute considered.

$$\underset{aba}{\forall} \; \underset{ep_k}{\forall} \; \frac{1}{|ep_k|} \sum_{i \in ep_k} \sum_{j \in S_i} aba_{ij} * y_{ij} = aba_k \tag{17}$$

where $|ep_k|$ denotes the number of tasks in a given execution path k.

Constraints families (17) express execution path ep_k value of average based attributes. Constraints family (17) is generated for each average based attribute considered.

$$\underset{mma\,LTB}{\forall} \; \underset{ep_k}{\forall} \; \underset{i \in ep_k}{\forall} \sum_{j \in S_i} mma_{ij} * y_{ij} \geq mma_k \tag{18}$$

Constraints families (18) express execution path ep_k value of minimum based attributes in case they are LTB. For each execution path, the appropriate variable is generated. Constraints family (18) is generated for each minimum based attribute considered.

Constraints families (19) show example of STB.

$$\underset{mma\,STB}{\forall} \; \underset{ep_k}{\forall} \; \underset{i \in ep_k}{\forall} \sum_{j \in S_i} mma_{ij} * y_{ij} \leq mma_k \tag{19}$$

In case of probability based attributes the constraint is not linear and looks like follows:

$$\underset{pba}{\forall} \; \underset{ep_k}{\forall} \; \prod_{i \in ep_k} \prod_{j \in S_i} pba_{ij}^{\;y_{ij}} = pba_k \tag{20}$$

Probability based attributes constraints families can be linearized by applying the logarithm function as shown below:

$$\underset{pba}{\forall} \; \underset{ep_k}{\forall} \; \sum_{i \in ep_k} \sum_{j \in S_i} \ln(pba_{ij}) * y_{ij} = \ln(pba_k) \tag{21}$$

In turn in case of functionality dependent attributes, the constraint is also not linear and looks like follows:

$$\underset{fda}{\forall} \; \underset{ep_k}{\forall} \; \prod_{i \in ep_k} \prod_{j \in S_i} f(s_{ij})^{z_i y_{ij}} \geq fda_k \tag{22}$$

They can be also linearized by applying the logarithm function as shown below:

$$\underset{fda}{\forall} \; \underset{ep_k}{\forall} \; \sum_{i \in ep_k} \sum_{j \in S_i} \ln(f(s_{ij})) z_i * y_{ij} \geq \ln(fda_k) \tag{23}$$

The same linearization needs to be of course applied in case of objective function for these two groups of attributes.

Constraints families (24) through (30) are the global constraints to be fulfilled.

$$\underset{cpa}{\forall} \; \underset{ep_k}{\forall} \; cpa_k \leq CPA + \delta_{CPA} \tag{24}$$

where CPA is the constraint value (CPA>0) with added additional fuzzy margin as already discussed. If the critical path attribute happens to be the process duration, the time necessary for binding operations should be also included. Therefore, the constraint value, instead of relaxing it, should be more stringent as overhead for binding each task should be subtracted.

$$\underset{sa}{\forall} \underset{ep_k}{\forall} sa_k \leq SA + \delta_{SA} \tag{25}$$

where SA is the constraint value (SA>0).

$$\underset{aba}{\forall} \underset{ep_k}{\forall} aba_k \geq ABA - \delta_{ABA} \tag{26}$$

where ABA is the constraint value (ABA>0). The direction depends on the character of the parameter.

$$\underset{mma}{\forall} \underset{ep_k}{\forall} mma_k \geq MMA \tag{27}$$

where MMA is the constraint value (MMA>0). However, the constraint family (27) in fact will not become a part of the ILP problem. These constraints are checked during the first step of the selection.

$$\underset{pba}{\forall} \underset{ep_l}{\forall} pba_l \geq PBA \tag{28}$$

where PBA is the constraint value (PBA>0) thus the following constraint is used:

$$\underset{pba}{\forall} \underset{ep_l}{\forall} (-1) * \sum_{i \in ep_l} \sum_{j \in S_i} \ln(pba_{ij}) * y_{ij} \leq (-1) * \ln(PBA) \tag{29}$$

Note that PBA and pba_{ij} in (29) are positive real numbers less or equal to 1, then their logarithm is negative. Hence the inequality (28) can be written as a less or inequality with positive coefficient (29). The same applies to constraint (30).

$$\underset{fda}{\forall} \underset{ep_k}{\forall} (-1) * \sum_{i \in ep_k} \sum_{j \in S_i} \ln(f(s_{ij})) z_{ij} * y_{ij} \leq (-1) * \ln(FDA), \; FDA > 0 \tag{30}$$

Regarding the constraints (26) written as demand constraints (greater or equal inequality with positive coefficient), it can be transformed into capacity constraints. The aba of a service is a real number in [0,1] and it is a positive criteria . Thus, a complementary property, denoted as *caba* may be defined as follows *caba=1-aba*. Then, the equation takes the following form:

$$\underset{aba}{\forall} \frac{1}{|ep_k|} \sum_{ep_k} \sum_{j \in S_i} caba_{ij} * y_{ij} \leq CABA \tag{31}$$

In the ILP problem only the global constraints or those attached to process fragments or structures are included, as those attached to tasks, have already been considered within the first step. All constraints attached to process fragments are generated similarly to those described above.

For the sake of simplicity, most of the authors assume that a Web service implements a single operation. This, however, does not allow to take into account statefull services or various dependencies between services (and their operations) in general. Therefore the additional constraints, resulting from the rules that if you choose s_x then you have to select also s_y need to be added.

These additional constraints are as follows. If two tasks t_{i1} and t_{i2}, $i_1, i_2 \in T$, must be executed by the same Semantic Web service, the following constraint families are introduced (dependency modelled by a task attribute *implementedByTheSameService*):

$$\begin{aligned} &\forall_{j \in S_{i_1} \cap S_{i_2}} \; y_{i_1 j} = y_{i_2 j} \\ &\forall_{j \in S_{i_1} \setminus S_{i_2}} \; y_{i_1 j} = 0 \\ &\forall_{j \in S_{i_2} \setminus S_{i_1}} \; y_{i_2 j} = 0 \end{aligned} \qquad (32)$$

In fact two last set of equations are not necessary as during the first stage of selection, the services that may be used to execute one task but not the other are discarded.

This constraint is also relevant in case of loops. If the loops are unfolded, usually each task performed within the loop should be executed by the same service (especially if the statefull services are considered).

In turn, if two tasks t_{i1} and t_{i2}, $i_1, i_2 \in T$, should not be executed by the same Semantic Web service (the attribute notImplementedByTheSameService), the following constraint families are introduced:

$$\forall_{j \in S_{i_1} \cap S_{i_2}} \; y_{i_1 j} = 0 \vee y_{i_2 j} = 0 \qquad (33)$$

what is expressed as:

$$\forall_{j \in S_{i_1} \cap S_{i_2}} \; y_{i_1 j} + y_{i_2 j} = 1 - M \qquad (34)$$

where M is a binary variable. Then, if M equals 1, none is selected. If M equals 0, one of them is selected.

In case of dependencies between services (if one is selected, the other also needs to be selected) are to be taken into account, the following constraints are generated:

$$\forall_{d} \; \forall_{s_{i_1 j_1}, s_{i_2 j_2} \in D_d} \; y_{i_1 j_1} = y_{i_2 j_2} \qquad (35)$$

where D_d is the set of dependent services within the d-th dependency (the attribute *needsToBeUsedTogetherWith or* based on the information that the service creates a bundle with other service may also be used within the process[50]).

[50] If services to be used within the process belong to the same bundle, then they appear twice within the set of relevant services, once with the cost as assigned in the service profile and second time as a services with the cost being the a part of the cost of the bundle and additional constraint saying they need to be used together.

In case of constraints modelled by the attribute *implementedByTheSameProviderAs*, the following set of constraints is modelled:

$$\forall_{p \in S_{i_1} \cap S_{i_2}} \sum_{j \in p} y_{i_1 j} = \sum_{j \in p} y_{i_2 j}$$
$$\forall_{p \in S_{i_1} \setminus S_{i_2}} y^p{}_{i_1 j} = 0 \tag{36}$$
$$\forall_{p \in S_{i_2} \setminus S_{i_1}} y^p{}_{i_2 j} = 0$$

Similarly to the case with the constraint (32), the two last sets of constraints are taken into account during the first stage of filtering.

Finally, in case of constraints modelled by using the task attribute *notImplementedByTheSameProviderAs*, the following set of constraints is added:

$$\forall_{p \in S_{i_1} \cap S_{i_2}} \sum_{j \in p} y_{i_1 j} = 0 \vee \sum_{j \in p} y_{i_2 j} = 0 \tag{37}$$

what is expressed as:

$$\forall_{p \in S_{i_1} \cap S_{i_2}} \sum_{j \in p} y_{i_1 j} + \sum_{j \in p} y_{i_2 j} = 1 - M \tag{38}$$

where M is also a binary variable.

The user's utility is expressed by means of preference function (the objective function F) that maps from a multidimensional space to a one-dimensional utility space. F is an additive scoring function composed of the attribute-specific functions and their relative weights. Due to the additive form of the scoring function F, mutual preferential independency between the attributes in the scoring function, as suggested in (Keeney and Raiffa, 1976), needs to be assumed. This assumption holds if the utility of an attribute does not depend on the value of another attribute.

The objective function F of the ILP problem that is to be maximized is defined as:

$$F = \sum_k prob(ep_k) * score(ep_k) \tag{39}$$

where $prob(ep_k)$ denotes the probability of execution of given execution path k and $score(ep_k)$ denotes the synthetic indicator value of the given execution path k.

The synthetic indicator value of a given execution path k is computed as follows:

$$score(ep_k) = \sum_{p=1}^{n} (V_{k,p} * W_p) \tag{40}$$

where $W_p \in [0,1]$ and

$$\sum_{p=1}^{n} W_p = 1 \tag{41}$$

Thus, W_p represents the weights of each property included. The weights value is computed based on the preference given by a user. In general, n denotes the number of all properties considered. If a weight equals 0, it means that the given property is of no importance for a user so it should be discarded while computing the overall score of the execution path and in consequence also a process. A user may of course change at any time the assigned weights.

The values $V_{k,p}$ for each execution path k are computed in the following way. Given the execution plan the values of the certain property assigned to each task (i.e. the values of property of an assigned service) are aggregated taking into account the property character and defined aggregation method (e.g. in case of price, the sum of price associated to each task is taken). The value $V'_{k,p}$ (for each property) obtained in this way needs to be unified and normalized in order to be used to compute the synthetic indicator value of the execution path.

For all STB attributes, values are scaled according to the following equation:

$$V_{k,p_{STB}} = \begin{cases} \dfrac{V'^{\max}_{p_{STB}} - V'_{k,p_{STB}}}{V'^{\max}_{p_{STB}} - V'^{\min}_{p_{STB}}} & if\ V'^{\max}_{p_{STB}} - V'^{\min}_{p_{STB}} \neq 0 \\ 1 & if\ V'^{\max}_{p_{STB}} - V'^{\min}_{p_{STB}} = 0 \end{cases} \tag{42}$$

where p_{STB} denotes the STB property considered.

For positive criteria, values are scaled according to following equation:

$$V_{k,p_{LTB}} = \begin{cases} \dfrac{V'_{k,p_{LTB}} - V'^{\min}_{p_{LTB}}}{V'^{\max}_{p_{LTBj}} - V'^{\min}_{p_{LTBj}}} & if\ V'^{\max}_{p_{LTBj}} - V'^{\min}_{p_{LTBj}} \neq 0 \\ 1 & if\ V'^{\max}_{p_{LTBj}} - V'^{\min}_{p_{LTBj}} = 0 \end{cases} \tag{43}$$

where p_{LTB} denotes the LTB property considered.

In the above-mentioned equations, V'^{max}_p is maximal value of a quality criterion for all execution plans associated with the given execution path. In turn, V'^{min}_p is minimal value of a quality criterion for all execution plans of a given execution path. The maximal and minimal values can be computed without generating all possible execution plans. E.g. in order to compute maximal price of all the execution plans to a given execution path, the most expensive SWS for each task is selected and then execution prices are summed up. In order to compute the minimum duration of all execution plans, the WS that has shortest execution duration for each task is identified and use CPA to compute min value. Thus, the computation cost of min and max values is polynomial (Zeng et al., 2003).

In case of probability based properties, e.g. availability and reliability, they are normalized using the following:

$$\frac{pba_i - \min(pba)}{\max(pba) - \min(pba)} \tag{44}$$

is replaced with:

$$\frac{\ln(pba_l) - \min(\ln(pba))}{\max(\ln(pba)) - \min(\ln(pba))} = \frac{\sum_{i \in ep_l} \sum_{j \in S_i} \ln pba_{ij} * y_{ij} - \min(\ln(pba))}{\max(\ln(pba)) - \min(\ln(pba))} \quad (45)$$

Thus, within the second step, service selection problem is modelled as a mixed integer linear problem where local constraints and global constraints can be specified.

The approach proposes the formulation of the optimization problem as a global optimization not optimizing separately each possible execution path as in other approaches except for (Ardagna and Pernici, 2005). To the ILP problem model formulated in this way, one of the existing algorithms either heuristic based or guaranteeing finding an optimal solution can be applied. Thus, once the selection problem is modelled as a MILP problem, well-known algorithms to find the optimal service selection may be utilised.

The summary of the notation used follows in Table 21.

Table 21 Notation summary

Symbol	Description
i	Task index
t_i	i-th task
j	Web service index
s_j	j-th Web service
$prob(ep_k)$	Probability of execution of the k execution path
N	Number of quality dimensions of interest
n	Quality dimension index
q_n	n-th quality dimension
W_n	Weight associated to the n-th quality dimension by the end user
XOR	Exclusive split
a	Total number of tasks
b	Number of candidate web services
sa_j	Value of a sum attribute of the j-th service
aba_j	Value of an average based attribute of the j-th service
cpa_j	Value of a critical path attribute of the j-th service
pba_j	Value of a probability based attribute of the j-th service
fda	Value of a functionally dependent attribute of the j-th service
k	Execution path index
K	Number of execution paths
ep_k	k-th execution path
m	Sub path index
sp^k_m	m-th sub path of the k-th execution path
$V'_{k,p}$	Value of the p-th quality dimension evaluated on the k-th execution path
$V_{k,p}$	Normalized value of the p-th quality dimension evaluated on the k-th execution path
CPA, SA, PBA, FDA, ABA	The global constraints value for cpa, sa, pba, fda, aba properties

4.5 Selection mechanism – overview and criticism

The two-step business conditions aware selection mechanism selects a set of services from a plurality of providers and their services in order to maximize the overall value of a process according to user requirements. In order to do that, it requires an access to:

- a process model it is to optimize - the composed process plans with service capabilities assigned (understood as a set of services capable to fulfil certain goal) along with additional information;
- a set of feasible services for each task;
- a service profiler module that stores performance information from previous Web service invocations as well as is able to create service profiles on demand;
- a set of user requirements comprising organization profile, user profile, as well as requirements of business users regarding the process itself.

Based on the above-mentioned data, a selection mechanism determines a set of feasible (in the light of the requirements) services and then, their optimal configuration.

The selection mechanism described within the last two sections being a part of the SWS e-marketplace is depicted in Figure 27.

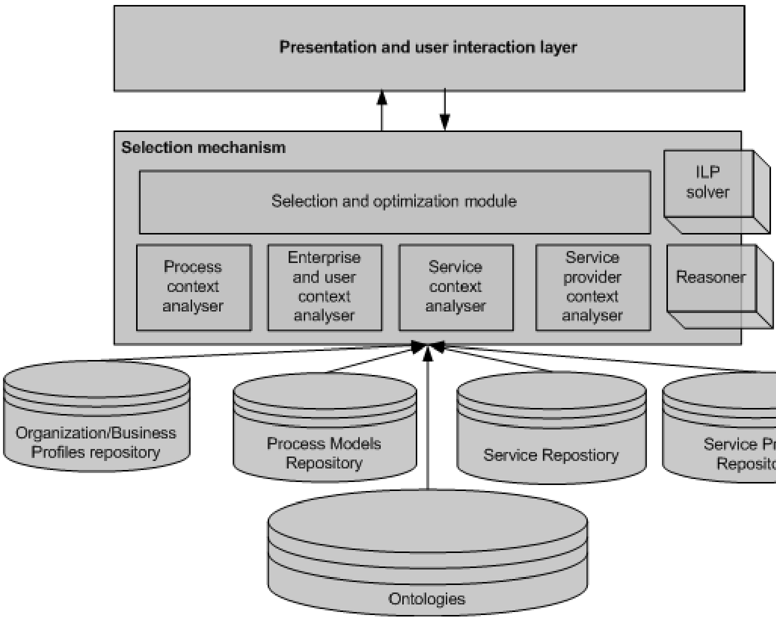


Figure 27 Overview of the selection mechanism

The decision, which configuration of SWS to use is based on the number of information. Therefore, the selection mechanism operates on the number of data and information sources to perform its task and requires an access to the following repositories: domain ontologies repository, process models repository, service repository, service profile repository as well as organization and business profiles repository.

Ontologies used by the e-marketplace in question encompass the ontology used to describe the SWS, domain ontologies as well as ontologies representing the information model developed within this thesis along with the selection criteria used within the selection mechanism. The selection mechanism uses a reasoner to support inference on the ontological representation.

Process models repository stores the process models generated by the composition mechanism. Each of the process models has additional requirements attached and. Service repository stores the information on the services registered on the e-marketplace. In turn, service profile repository stores service profiles that were created by the service profiling mechanism such as e.g. the one implemented within the ASG system or enhanced with semantics.

Finally, the organization and business profile repository stores information about the business and organization context, rules and policies, as well as business requirements (general) ones that are not dependent on the single process model.

The most important part of the selection mechanisms is the selection and optimization module that takes into account all available information on services, process model along with all user requirements. First, it narrows down the acceptable configurations space and then performs the optimization.

The main contribution of the proposed solution is as follows:

- the easy extensible information model developed as an upper level ontology that encompass both business (high-level) as well as technical view on the both process and task layer;
- inclusion of business properties i.e. adding e.g. KPI and linking them to the lower layers of abstraction;
- inclusion of the aggregation methods into the logic of the information model thus ensuring the flexibility and extensibility of the model;
- inclusion of a set of profiles allowing to express requirements, business conditions as well as service provisioning terms. Taking into account business perspective and service provisioning conditions the results delivered are more suited to the business needs and expectations. In addition, relationships such as partnerships, alliances and competitors among the service providers are also taken into account;
- various mappings between business and technical views have been incorporated into the logic of the information model thus allowing for efficient decomposition of various requirements;
- the proposed mechanism, by using profiles and additional set of requirements (including business oriented ones) allows to narrow down the number of available services. Due to the computational complexity of the algorithm used to solve the MILP problem, the less possibilities the better (the more efficient the algorithm is). Therefore, the proposed approach cuts off the possible set that should be checked by

decomposing the constraints. Whereas the algorithm has the exponential complexity, decomposition has only polynomial time complexity.

- the formulation of the MILP problem, although starting from the work of others e.g. (Ardagna and Pernici, 2007):
  - uses different aggregation techniques,
  - breaks with sometimes postulated equal partition assumptions and probability of execution of each conditional branch is taken into account;
  - performs optimization of all execution paths at once and thus guarantees the global optimum.;
  - takes into account various business conditions in the form of requirements as well as fuzziness of the constraints;
  - takes into account also dependencies between services thus allowing dealing with statefull services as well as service bundles. These dependencies are modelled as a set of additional constraints within the MILP model;
  - the set of additional constraints allow to take into account some business rules and dependencies e.g.: task A and B need to/should not be performed by the same service/provider; or if one service or provider is selected then also other needs to be selected etc.
  - it is possible to indicate that all tasks resulting from the loop unfolding need to have the same service assigned.
  - reasonable relaxation of constraints so as to avoid situation that if the end-user introduces severe constraints i.e., limited resources which set the problem close to unfeasibility conditions no solutions can be identified and the composed service execution fails.

The proposed solution has of course several disadvantages or weak points that still need to be addressed. The model assumes that the process model can be represented as a sequence of structured elements. This is not always the case and the transformation, although possible, has its limitations (Aalst et al., 2003).

The model depends on the statistical information on the number of iterations of loops as well as probability of execution of conditional branches. This data is used in order to identify the optimal execution plan. If the data is not accurate, the identified plan will be suboptimal. In addition, the aggregation methods, similarly to those proposed in the literature, follow the assumptions of independence. It is assumed that the services in the composition do not depend on each other regarding their successful execution. This may not be always the case.

In addition, the assumption of uniformity ensured by the dynamic service profiling mechanism is followed. For the aggregation, it is assumed that the given values are compatible to each other. All individual values must conform to a common group of measure. For example, it is assumed that a property describing the response time uses compatible measures such as milliseconds seconds or minutes. Further more it is assumed that all given values conform to the same definition. For example, all services start and end from the same point of measurement to define the response time, e.g. when the request has been submitted and the response has been delivered completely. This is however, difficult to achieve in real life conditions.

In addition, it is assumed users and providers share a common ontology to describe various concepts. One common model in reality it is rarely possible.

In addition, the proposed model operates on organization sensitive data, therefore, trust issue and privacy issue needs to be considered. The creation of the profiles may be a complex task and business users would require an appropriate interface that would allow them to model a process, compose a process as well as add additional constraint and requirements to the process.

The proper operation of the proposed two step selection approach requires utilization of the developed information model. Designing it as an upper level ontology has certain advantages i.e. extensibility and that it could be applied to any domain. However, the extensibility becomes also its downside as ontology modelling and instantiation of various concepts is time-consuming approach.

4.6 Summary

In this chapter, the conceptual model is presented. A conceptual model is an abstract framework for understanding significant relationships among the entities of some environment (OASIS, 2006a). It should consist of a minimal set of unifying concepts, axioms and relationships within a particular problem domain. It is independent of specific standards, technologies, implementations or other concrete details. They are given in the validation part.

The presented model is grounded on previous work in this area (Ardagna and Pernici, 2005, 2007, Liu et al., 2004, Yu et al., 2007). The proposed model builds upon the previous solutions and enhances them in such a way as to meet business expectations.

Within the next chapter the validation of the developed service selection is presented.

PART III: VALIDATION

The objective of this last part of the thesis – the validation part - is to provide a proof-of-concept for the viability of the selection approach. Additionally, this part shows how the selection model can meet requirements outlined in Chapter 4.

The validation, in addition to the conceptual model, aims at providing a fundament for the concept and design of a selection system, which can be extended and adapted to certain domains and usage areas. This, in turn, may support practitioners in implementing such a system and solving involved problems.

The validation of the approach represents an important augmentation of the theoretical part. It also presents an alternative viewpoint on the research results and can facilitate the understanding of the approach. On this basis, the validation helps in the identification of open issues to be tackled in future research.

5 Evaluation of the approach

Within this chapter, the evaluation of the research results is presented and discussed. The chapter is structured as follows. First, the evaluation approach and verification measures are discussed. Then, the description of the conducted experiments follows. Next, the developed model is evaluated using the requirements defined within the previous chapter. Finally, conclusions follow.

5.1 Evaluation approach and verification measures

In general, there are three basic measures used to verify the quality of a solution in a computer science, namely: effectiveness, efficiency and elegance e.g. (Eades, 2008). The effectiveness checks whether the solution is logically correct and optimal. In addition, the effectiveness should state whether it is satisfactory for the customer. Efficiency checks the usage of computational resources by the solution. In turn, elegance checks whether the solution is pleasingly ingenious and simple.

(Eades, 2008) distinguishes three basic evaluation methods that may be used to discover the values of these measures, namely: mathematics, experiments and use-case scenarios as well as combinations of these three approaches. The following table summarizes the features of these methods.

Table 22 Evaluation methods (Eades, 2008)

Method	Strengths	Weaknesses
Mathematics	Robust to model changes Good evaluation of pathological behaviour	Does not evaluate the model
Experiments	Evaluates the model Good evaluation of normal behaviour	Poor evaluator for pathological behaviour
Use cases scenarios	Convinces the non-scientific customer. Appropriate to evaluate elegance	Poor evaluator of efficiency and effectiveness

The assumptions, concepts, and mechanisms that are the basis for the approach presented within the previous chapter are rather young and are topics of ongoing research. In-progress research includes issues such as: B2B SWS e-marketplaces in which services are requested and provided on-demand; availability of the SWS and their description; availability of service profiles which can be created, modified, and monitored automatically and which serve as an instrument to control and manage the provision of services. In addition, any proposed selection mechanism is to operate on company specific and private data that are not publicly available. Therefore, the validation of the approach could not be realized in a real business environment and existing tools and methods could only be reused partially. Thus, the validation of the approach is rather explorative and based on use case scenarios, which only partially can be verified in a production environment. In addition, a simulation setup allowing for verification of the proposed model needed to be prepared.

Therefore, taking into account guidelines of the research methodology, features of the developed model and the fact that the fully-fledged SWS marketplace are not available, the validation of the proposed solution has been performed as indicated in Table 23.

Table 23 Validation of the conceptual model (own study)

Measures	Criteria	Evaluation method	Comments
Effectiveness	Logically correct	Competency question	Competency questions allow to check whether the information model fulfils defined requirements and is logically correct
		Scenario run through	The experiment is to show that the returned results are in accordance with expectations and are logically correct
	Optimal	Mathematics	The optimality of the assignment depends on the quality of the data used as well as the approach (i.e. method) used to solve the formulated ILP problem. The correctness of the data is checked using the competency questions as well as use case scenarios. Evaluation of the methods is not necessary because well known optimal or heuristic algorithms to solve ILP problem may be used to which mathematical proofs showing their features may be found in the relevant literature.
	Satisfactory to the customer	Qualitative analysis - comparison	The developed model is evaluated using 10 requirements illustrating expectations of e-marketplace participants. A table with all requirements and comments pointing to the level of fulfilment may be found in 5.4
Efficiency	Usage of resources	Experiment	The usage of resources and complexity of the problem depends on the algorithm used as well as concrete requirements and available solution. Some insights into this issue are given in general experiment section.
Elegance	Simplicity, user friendliness	The qualitative analysis of the developed solution points to user friendliness	Showing the run-through scenario, the user role in certain activities is discussed. Thus, the simplicity and user friendliness of the developed solution is mentioned.

According to the presented table, the validation of the approach has been divided into three parts. First, the information model is evaluated using a set of formally represented competency questions. Its logical correctness and appropriate coverage are also tested using the two use case scenarios developed within two European projects.

The main mechanism of the selection process is evaluated using a simulation setup allowing for verification of the proposed algorithm. As indicated in the table the optimality of the assignment depends on the algorithm selected to solve the selection problem. This issue is not discussed in depth during validation.

Finally, the proposed model is validated against the set of requirements defined within Chapter 4. The goal is to check whether the proposed solution meets the expectations of business users.

It is deemed that the positive results of all of the mentioned parts sufficiently prove the validity of the thesis of this dissertation and show the fulfilment of the research goals.

5.2 Evaluation of information model

As stated in (Uschold and Grueninger, 1996) the competency questions are not only useful during the conceptualization process but are also used to validate the ontology. The evaluation should determine whether the proposed ontology ought to be extended or all competency questions can already be answered by the developed ontology. To this aim, the formal competency questions i.e. expressed as queries in the given ontology language, are used.

5.2.1 Evaluation of competency questions

The competency questions have been validated using the IRIS III reasoner embedded in the WSMO studio v. 0.7.3.8. (Bishop and Fischer, 2008) report that the functionality and performance of IRIS compare favourably with similar systems. IRIS III allows for asking WSML-Flight queries. WSML-Flight is an extension of WSML-Core which provides a rule language. It adds features such as meta-modelling, constraints and non-monotonic negation.

Based on the informal competencies questions formulated in Chapter 4, the following formal competencies questions have been expressed and tested on the reasoner.

- What is the value of a characteristic X of a given service?

  Query for listing all properties assigned to a given service “s1”:

```
?x memberOf Property and ?x[value hasValue ?value] and ?x[hasUnit hasValue
?unit] and ?z memberOf ServiceProfile and s1[hasProfile hasValue ?z] and
?z[hasProperty hasValue ?x]
```

  Query for listing only cost associated with a given service “s1”:

```
?x memberOf Cost and ?x[value hasValue ?value] and ?x[hasUnit hasValue
?unit] and ?z memberOf ServiceProfile and s1[hasProfile hasValue ?z] and
?z[hasProperty hasValue ?x]
```

Query for listing all properties assigned to all services:

```
?x memberOf Property and ?x[value hasValue ?value] and ?x[hasUnit hasValue
?unit] and ?y memberOf Service and ?z memberOf ServiceProfile and
?y[hasProfile hasValue ?z] and ?z[hasProperty hasValue ?x]
```

- What are the constraints connected with the task A?

```
?x memberOf Constraint and taskA[hasConstraints hasValue ?x]
```

- What preferences has a user towards the given process model?

```
?x memberOf Preference and ?profile[hasPreference hasValue ?x] and
process[hasProcessProfile hasValue ?profile]
```

- What is the type of the property? What is the character of the property? What is the changeability?

As the disjunctions are currently not supported by the reasoner in question, there are two possibilities. Either to ask about all superconcepts of a property (see Figure 28):

```
PropertyX memberOf ?y and ?x subConceptOf Property
```

And then identify which type was returned (e.g. Qualitative, Quantitative) as indicated below.

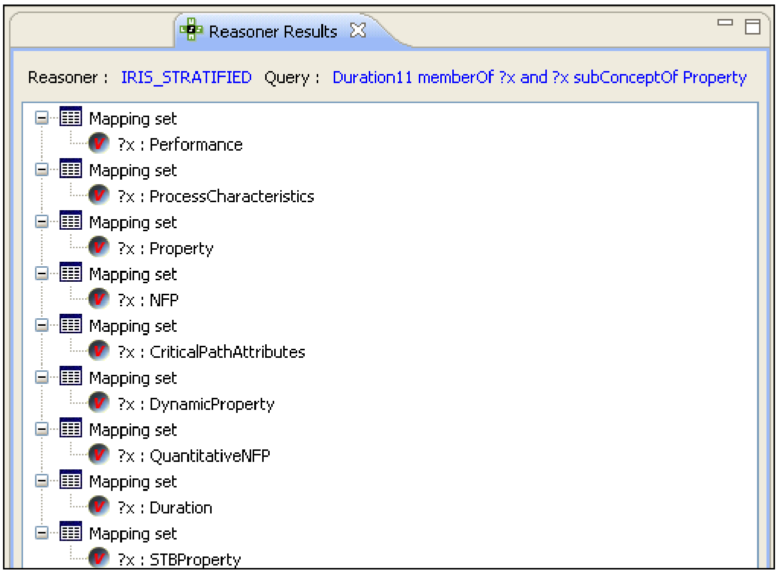


Figure 28 WSML Query – superconcepts of an instance

It is also possible to ask whether a given property is a *memberOf* a certain type. If no result is returned, then it is not, e.g. `Cost1 memberOf LTB`. If this is not true, then no results matched message is returned by the reasoner.

- Which elements does the process consist of? (see Figure 29)

```
?x memberOf Sequence and process1[hasSequence hasValue ?x] and
?x[hasElements hasValue ?y]
```

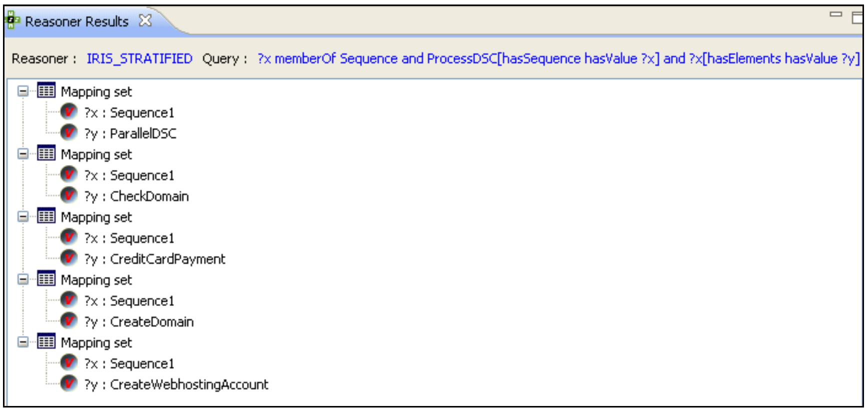


Figure 29 WSML Query – process structure

- Which workflow structures are present?

  ?x memberOf Sequence and process1[hasSequence hasValue ?x] and ?x[hasElements hasValue ?y] and ?y memberOf ?z and ?z subConceptOf BlockStructure

- Which service may realize the goal of a certain task? (see Figure 30)

  Task1[hasGoal hasValue ?x] and ?z memberOf Service and ?z[fulfillsSWSGoal hasValue ?x]

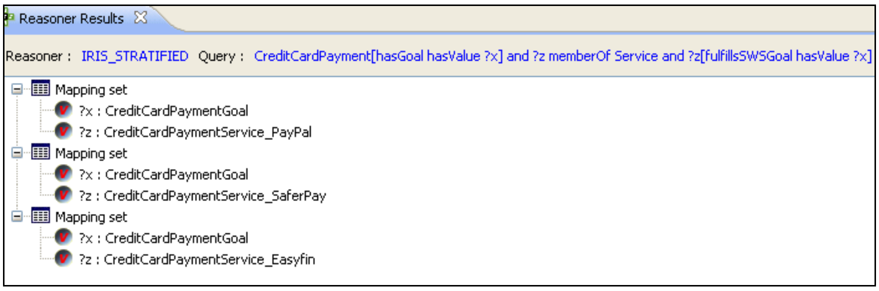


Figure 30 WSML query – which service can realize the goal of a certain task

- Which properties are defined within the SLA?

  SLA1[hasSLO hasValue ?x] and ?x[hasPropertyValuesGuarantees hasValue ?y]

- Are there any constraints assigned to tasks or any other process structure belonging to a given process?

  Process1[hasSequence hasValue ?x] and ?x[hasElements hasValue ?y] and ?y[hasConstraints hasValue ?z]

- What is the relation between X business property and more technical layer?

  businessPropertyX[relatesTo hasValue ?x]

  businessPropertyX[relatesToLevel hasValue ?x]

  businessPropertyX[hasCalculation hasValue ?x]

- What is the trust level assigned to a given partner?

  ?z memberOf TrustedParty and ?z[relatesToProvider hasValue provider1] and ?z[hasTrustValue hasValue ?y]

- What is the probability of execution of a given alternative structure?

```
Xor1[hasBranches hasValue ?x] and ?x[hasProbability hasValue ?y]
```

- Which services fulfil certain goal?

```
?x memberOf Service and ?x[fulfillsSWSGoal hasValue certainGoal]
```

- What is the horizon of the process?

```
Process1[hasProcessProfile hasValue ?x] and ?x[hasHorizon hasValue ?y]
```

- What is the organization profile associated with the given business user profile?

```
?x memberOf BusinessUser and ?x[hasUserProfile hasValue Profile1] and
?x[belongsToOrganization hasValue ?y] and ?y[hasOrganizationProfile
hasValue ?z]
```

- Which property is constrained by the given constraint? What is the desired value?

```
Constraint1[constraintsValueOf hasValue ?z] and
Constraint1[hasConstraintValue hasValue ?y]
```

The extended list of formal competency questions (queries expressed in WSML notation) formulated during the evaluation may be found in Listing 8.4.

The validation results showed that the designed information model is able to provide answers to all considered questions, although, sometimes some additional operations and transformation need to take place. For instance, due to the lack of support for disjunctions more than one query needs to be formulated in order to get to know the type of the instance. This is due to the modelling decisions taken during the information model creation.

The information model is also used to answer more complex questions within the first step of the selection process in order to discard some candidates, e.g.:

- Give all preferences that has priority greater than3

```
?z memberOf PropertyPreference and ?z[hasPriority hasValue ?x] and ?x>3
```

- List all services that fulfil a certain goal that is realized by taskA and the services need to be provided by PayPal and has encryption implemented

```
?z memberOf Service and ?z[fulfillsSWSGoal hasValue ?x] and TaskA[hasGoal
hasValue ?x] and ?z[hasProvider hasValue ?v] and ?v[hasName hasValue
"PayPal"] and ?z[hasProfile hasValue ?p] and ?[p hasProperty ?t] and ?t
memberOf Encryption and ?t[implemented hasValue true].
```

5.2.2 Dynamic Supply Chain scenario

The information model has been validated using the use case scenarios developed within two European projects that the author participated in. Based on these two scenarios, the logical correctness of the model has been checked.

The first use case is based on a business-to-business (B2B) wholesale model of an Internet service provider (ISP). The use case description is based on a usage scenario, which was developed, described, and implemented in the Adaptive Services Grid project (Noll, 2004). Issues that are not relevant to the topic of this dissertation are neglected.

The ISP offers the whole spectrum of Internet services for Web space provisioning. For this, it uses several atomic services, like domain name checking and registration, Web hosting configuration, operation and maintaining of Domain Name Server information, handling of payments bundled to end customer products, messaging, and so forth. All services are provided by external service providers. Each service can be provided by different external service providers with different functional and non-functional characteristics.

The concrete process model that is presented in the use case is a dynamic supply chain for automated Domain Name registration and provisioning of Web space (later on referred to as DSC service). Starting from a semantic request of a business customer "Provide a service for Domain Name registration, Web space provision, and payment processing" (initial state request and desired goal), a dynamic service composition engine (Service Composer) tries to reach from the initial state to the desired goal by connecting atomic services. The result of this service composition process is a generic service composition script that fulfils the service requester's request. The following figure shows the generic service composition script (i.e. the abstract process model) composed by a Service Composer for the received request.

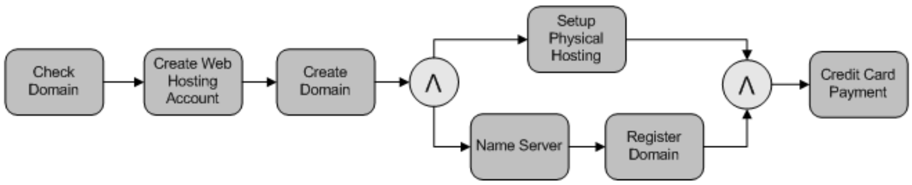


Figure 31 DSC abstract process model

Atomic services included in the DSC service are as follows:

- CheckDomain: This service checks whether a certain domain is available for registration. This is necessary since a domain can only be registered if it is not registered by someone else.
- CreateWebhostingAccount: This service creates a Web hosting account which can be used to access the Webspace.
- CreateDomain: This service creates a domain for the Webspace.
- SetupPhysicalHosting: This service sets up the physical Web hosting/Webspace which hosts the displayed files of a Website owner.
- Nameserver: This server updates the name server with the new domain connected to the created Webspace.
- RegisterDomain: This service registers a certain domain.
- CreditCardPayment: This service executes a complete credit card authorization and payment process. If the authorization process is successful, all further payments are processed with this type of payment.

As already mentioned, some of these atomic services can be provided by different service providers and some services can only be provided by one service provider. The following table provides an overview of the atomic service implementations that are available for integration into the generic service composition script. The DSC service and all atomic services are characterized by service parameters, not included in the table. These are inter alia response time, availability, throughput, and encryption level. Furthermore, services are offered at different financial terms.

Table 24 Service space in DSC scenario

Service specification	Service implementation	Providers
CheckDomain	checkDomain	myNic, Direct, domainPro
CreateWebHostingAccount	createWebHostingAccount	Hostit
CreateDomain	createDomain	CeDe, Symbo
SetupPhyscialHosting	setupPhysicalHosting	Integ
NameServer	nameserver	Franke
RegisterDomain	registerDomain	myNic, Directi, domainPro
CreditCardPayment	creditCardPayment	PayPal, SaferPay, Easyfin

Here, only excerpts from the description are presented. The process model representation is as follows:

```
instance ProcessDSC memberOf Process
     hasSequence hasValue Sequence1
     hasOwner hasValue User1
     hasProcessProfile hasValue DSCProcessProfile
     hasPreferences hasValue Preference3
     hasConstraints hasValue {StringentConstraint1, FuzzyConstraint1 }

instance Sequence1 memberOf Sequence
     hasElements hasValue {CheckDomain, CreateWebhostingAccount,
CreateDomain, ParallelDSC, CreditCardPayment }

instance ParallelDSC memberOf Parallel
     hasBranches hasValue {Branch1, Branch2 }

instance Branch1 memberOf Branch
     hasElements hasValue SetupPhysicalHosting

instance Branch2 memberOf Branch
     hasElements hasValue {Nameserver, RegisterDomain }
```

All tasks and goals needed to be instantiated:

```
instance CheckDomain memberOf Task
     hasGoal hasValue CheckDomainGoal

instance CheckDomainGoal memberOf Goal
```

A set of service providers has been modelled:

```
instance myNic memberOf Organization
     hasName hasValue "myNic"
```

A set of services:

```
instance CheckDomainService_myNic memberOf Service
     fulfillsSWSGoal hasValue CheckDomainGoal
     hasProfile hasValue CheckDomainService_myNic_Profile
     hasProvider hasValue myNic
```

A set of service profiles with properties assigned:

```
instance CheckDomainService_myNic_Profile memberOf ServiceProfile
hasProperty hasValue {availability_CheckDomainService_myNic,
executiontime_CheckDomainService_myNic,
throughput_CheckDomainService_myNic, cost_CheckDomainService_myNic,
chargingmethod_CheckDomainService_myNic,
encryption_CheckDomainService_myNic }
```

The business user expressed following requirements:

```
instance User1 memberOf BusinessUser
     hasName hasValue "John Smith"
     belongsToOrganization hasValue ISP
     hasUserProfile hasValue User1Profile
```

- The process cannot be costly.

```
instance Preference1 memberOf PropertyPreference
     optimizeValueOf hasValue ProcessCost
     hasPriority hasValue 5
```

- The process duration needs to be smaller than X minutes

```
instance FuzzyConstraint1 memberOf FuzzyConstraint
     constraintsValueOf hasValue processDuration
     hasComparator hasValue SmallerThan
     hasConstraintValue hasValue "X"
     hasUnit hasValue minute
```

- The responsiveness of the process should be greater than 100 per minute.

```
instance process_responsivness_constraint memberOf StringentConstraint
constraintsValueOf hasValue processResponsiveness
hasComparator hasValue GreaterThan
hasConstraintValue hasValue 100
hasUnit hasValue ResponsesPerMinute
```

- The task credit card payment should be fulfilled only by trusted parties. The critical functionality for this task is the encryption mechanism implemented.

```
instance CreditCardPayment memberOf Task
     hasGoal hasValue CreditCardPaymentGoal
     hasCriticalProperty hasValue dataEncryption
     implementedByTrustedPartyWithTrustAtLeast hasValue _float(0.5)
     implementedByProvider hasValue X
```

- Physical hosting should be performed by the service of provider Y.

```
instance SetupPhysicalHosting memberOf Task
     hasGoal hasValue SetupPhysicalHostingGoal
     implementedByProvider hasValue Y
```

In addition, on the e-marketplace, the service provider Directi declared the following dependency: if the Check Domain and Create Domain services are used together then the price discount is offered (i.e. they create a bundle).

```
instance bundle_1 memberOf Bundle
hasProvider hasValue Directi
hasServices hasValue {CheckDomainService_Directi,
CreateDomainService_Directi}
hasDoscountPrice hasValue price_directi_bundle
```

Thus, during the first step of the selection mechanism, the following operations took place. The business level properties i.e. process responsiveness and duration have been mapped to services layer. To each task, the services performing the defined goal and such as their execution takes less than X (constraints resulting from requirements on process duration), their throughput is greater than 1,66 per second (a constraint resulting from the requirement on process responsiveness) may be identified. In addition, for task 2 only the services of a given provider were considered. Also for task 4 only those services were considered that belong to service providers that we trust and that have the encryption implemented. The first step, based on the ontology model and using the reasoner, allowed to filter out some services and providers (Easyfin. because of the lack of encryption and not being a trusted party as well as domainPro services, because not meeting the responsiveness constraint even after its relaxation). However, the services constituting a bundle have been added twice to the set with additional attributes formulated.

After the MILP problem formulation with one execution path and two sub paths, the optimal selection of services could be identified. The assignment of services in local and two step selection is depicted in Table 25.

Table 25 DSC scenario - results

Approach	No. services	Assigned services	Process characteristics
Local, naïve approach	14	(t1) Directi, (t3) CeDe, (t6) myNic, (t7) Easyfin	overall utility : 0,15, cost =7 ; duration =5,8 ;
Two-step selection	12 + 2 from the bundle	(1) Directi, (t3) CeDe, (t6) Directi, (t7) PayPal	overall utility : 0,34 ; cost =6, duration= 5,6.

Although, within this scenario, the two step selection reduced the number of services to consider, two additional services forming a bundle have been added. The local selection assigns to each task a service with the highest ranking. Thus to t1, t6 and t7 – myNic, domainPro and Easyfin were selected as their cost is very low. However, the selection of these services to the process exceeded slightly the global constraint on the execution duration assigned to the process. In addition, the trust related issue was not considered during the selection. Also, the existence of service bundle was not considered.

In turn, the result of the two-step selection meets all of the requirements e.g. a service with encryption implemented and trusted party is assigned to t7 The process duration meets the defined constraint. The solver assessed that the usage of bundle instead of services of two separate providers is more favourable to the user. Therefore, the bundle was selected.

Therefore, the information model allowed to express the entire scenario as well business requirements and further on perform the optimal selection respecting the formulated requirements.

5.2.3 Voice over IP Service Ordering Process (SUPER project)

The business process described within this section is inspired by the scenario by Telekomunikacja Polska (TP) (Evenson et al., 2008) developed within already mentioned SUPER project (Semantics Utilised for Process Management within and between Enterprises). Telekomunikacja Polska Group is the dominant player in the Polish telecommunications market serving 10.6 million fixed-line subscribers and over 12 million mobile customers, as of Q1/2007, employing about 28.000 people. One of the services they offer to their customers is the voice over IP service.

The simplified version of voice-over-IP (VoIP) ordering business process is illustrated in Figure 32 using a simplified notation. Most events and all semantic annotations were removed from the process model for the illustration purposes.

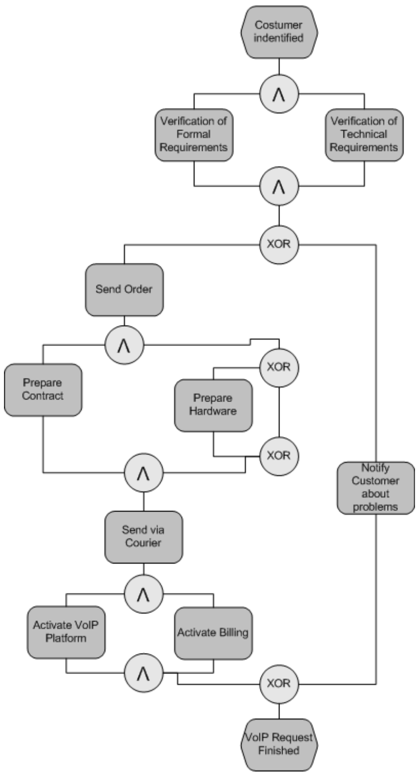


Figure 32 VoIP Ordering Process at Telekomunikacja Polska

The VoIP ordering process allows TP's customers to order the VoIP service for an existing contract. The ordering process is initiated by the customer through TP's web portal. After identifying the customer, the process first checks if all technical and formal requirements are fulfilled. If verification is positive, a new contract is created and simultaneously a check is run to see if the customer already has the necessary hardware. If not, the hardware is sent together with the contract to the customer. After TP receives the signed contract, the contract is archived, the billing system is activated, and finally the VoIP service is activated.

As already mentioned, some of these atomic services can be provided by different service providers and some services can only be provided by one service provider. In addition, within this scenario, differently from the previous scenario, some of the services need to be company internal ones (e.g. formal verification) owned by TP while others may be outsourced to external parties (e.g. send hardware). Even if services come from internal sources they still may be provided in various classes of quality. In fact, due to the complexity of the information system with a high number of legacy systems wrapped in services interfaces as well as appearance of new systems, there are a number of different services that could perform given task (Evenson et al., 2008).

Due to the privacy policy, the number and characteristics of actually existing services could not be included within this dissertation. Therefore, for each task a random number of SWS (from 1 to 6) has been assigned along with random properties. For each service, a provider from the set of available providers, has been assigned. Among them is also TPSA. An excerpt from the process model description is included in Listing 8.5.

Table 26 provides an overview of the atomic service implementations that are available for integration into the process template as well as assigned requirements (presented informally for the space reasons).

Table 26 VoIP scenario – service space (own study)

Service specification	No. of services	Constraints	No. of services left
VerifyFormalRequirements	5	The task should be realized by internal, secured services	1
VerifyTechnicalRequirements	2	None	2
NotifyCustomer	4	Internal service should be used to notify customers.	2
PrepareContract	8	The contract need to be prepared by an internal system X.	3
PrepareHardware	3	None	3
SendViaCourier	10	Both contract and hardware should be send using services of a trusted party.	4
ActivateVoIPPlatform	3	Internal SWS should be used.	2
ActivateBilling	5	One of the internal services should be used and correlated with the service activating the VoIP service to the client.	2

All of the requirements included in the table have been modelled using the information model proposed.

In addition, the company offers this process in various configurations – for gold, silver and regular clients. Thus, depending on which group of clients is using the process, different preferences are used during the selection. In consequence, a different set of services should be assigned. Indeed, the performed simulation allowed for assignment of services with the increased objective function of 20% in comparison to the local approach. For profile 1 (regular clients), the cost of the process is 1 and the duration is 3. In case, of gold clients, the process cost is 1,5 and the duration is 1,5.

5.2.4 Quality of developed ontology – validation results

In order to validate the quality of the developed ontology, the following six criteria have been considered, as described in (Fernandez-Lopez and Gomez-Perez, 2002, Gruber, 1993a). These criteria are as follows:

- consistency – referring to the absence (or not) of contradictory information in the ontology,
- completeness – referring to how well the ontology covers the real world,
- conciseness – referring to the absence (or not) of needless information or details,
- clarity – referring to how effectively the intended meaning is communicated,
- generality – referring to the possibility of using the ontology for various purposes inside the domain,
- robustness – referring to the ability of the ontology to support change.

The results of the evaluation are summarized in the following table.

Table 27 Information model evaluation results (own study)

Criterion	Comment
consistency	No contradictory information has been identified within the model. However, it was discovered that some information was redundant.
completeness	The information model was designed as an upper ontology. The competency questions as well as use case scenarios modelled showed that all relevant information is covered, however, as planned, its application in a selected domain requires using additional ontologies if a user is interested in domain-specific attributes
conciseness	No inconsistencies have been identified.
clarity	The clarity of the developed model was not evaluated. It may be only assumed that using the common terminology and adding explanations to each concept in the ontology ensures the clarity of the model to the audience.
generality	The information model proved to be general and extensible. It has not been checked in scenarios other than selection.
robustness	For the needs of the use case scenario, the ontology has been extended. No problems with this operations occurred.

As already mentioned, the completeness and conciseness of the information model has been checked by instantiating the concepts from two use case scenarios (described earlier within this chapter). Table 28 shows the number of concepts that were instantiated for all its logical parts. As indicated in the table, 43 concepts from the ontology were not instantiated. However, the considered scenarios required some additional concepts that needed to be defined.

Table 28 Number of concepts instantiated for the needs of use cases (own data)

Logical part	# concepts in the ontology (without concepts modelled for the aggregation purposes)	# concepts added	Instantiated concepts	
			#	%
Actor	3	0	3	100
Process artefacts	8	0	6	66%
Profiles	4	0	4	100%
Properties	51	2	20	39%
Requirement	6	0	5	83%
Service	1	0	1	100%
Supporting artefacts	17	0	7	41%

It must be noted that the fact that a concept has been instantiated does not state whether it is useful for the selection or not, but only that it appeared in the use cases considered. Thus, some concepts were not instantiated as they were not relevant from the point of view of the scope of the considered use case scenario. Some of the concepts were already instantiated e.g. comparator or unit instances. Therefore, there was not any additional need to define them. Moreover, the use case scenario required more specialised properties than those included within the ontology in question (an upper level ontology). Thus, new concepts needed to be added. Each new concept added has been fitted into the already existing structure. In overall, the description of the use cases required creating almost 500 instances.

5.3 Simulation

For all simulations, number of problem instances need to be generated. Depending on the simulation setup, some of them are fixed, some are step-wisely increased and some are randomly set. Considering the problem model, the elements of a problem instance are generated in the following way:

- task and candidates – if a problem instance is generated, then only the amount of tasks and candidates is relevant. The simulation ignores particular functionality of possible tasks or possible composition goals because the focus of this evaluation is the non-functional side of the candidates and process. In the simulation the amount of tasks and candidates is either step-wisely increased or set to fixed values;
- non-functional properties of services – the values of non-functional properties of candidates are stochastically generated as is discussed later in this section;
- optimization goals – in the majority of the simulations the optimization goals remain the same and generated problem instances have the requirement to optimize selected non-functional characteristics. These characteristics are as follows e.g.: cost, execution time, availability, reliability, throughput, security, trust.

- requirements and profiles– since the simulation evaluates the optimization capabilities a number of various constraints and preferences have been generated in order to show that the proposed solution is capable of meeting various requirements;
- structure – the structure of the process is generated stochastically. However, to evaluate the influence of the structural arrangements the likeliness of generating particular structural elements can vary.

In summary, performing the simulation i.e. performing a run, is divided into three main steps. First, a problem instance is generated, then the implemented algorithms try to solve the given problem and then the results are evaluated.

The simulation tool has been implemented in Java. The input of the tool are the relevant instances describing the process model, profiles and requirements along with service profiles specifying the set of candidate services along with their quality values. It is assumed that cycles are already unfolded and the process model encompasses the following elements: sequences, XOR, parallel structures. The transformation of relevant artefacts from WSML to Java objects is performed.

Execution paths are extracted and a depth-first like algorithm identifies the set of sub paths of every execution path. The depth-first approach allows optimizing the sub path generation process. If for example two sub paths have a common path, then the second sub path can be obtained from the partial result computed for the first one. During extraction of execution paths and sub paths, the probability of execution is assigned to each path based on the values in the information model.

Then, after decomposition of all constraints and discarding all unfeasible services, as well as generation of vector of weights, MILP model constraints are generated and the optimization problem is solved by running CPLEX (see Figure 33).

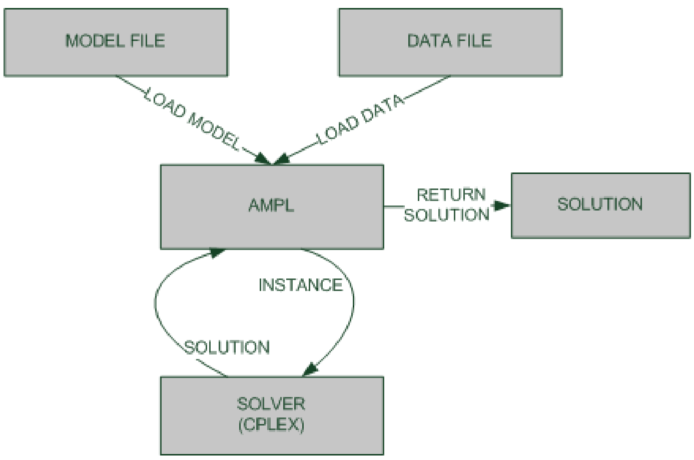


Figure 33 Solving selection problem using CPLEX

The MILP and its variants are well known problems. The operations research literature, which provides several methods to solve the MMKP or MILP, is widely available. Commercial integer linear programming solvers can solve problem instances of medium size. Therefore, instead of focusing on yet another implementation of MILP problem solver, the instances of the MILP problem considered here have been solved using CPLEX, a state of the art integer linear programming solver. The solver identifies the optimum solution of the MKKP i.e. for a given composite Web service the optimum global plan is identified. The free student version of AMPL (available at www.ampl.com), can handle up to 300 variables and 300 constraints. This is sufficient for the needs of the experiment.

For the needs of experiment, the general model of the problem in the APML language has been created. AMPL is a platform that allows to express mathematical models in a standardized form. Depending on which solver is selected, AMPL converts the model into one that can be read by the solver. After the solver solves the model, it sends its results back to AMPL, which in turn displays them in a standardized form. An excerpt from the developed model is shown in the following listing:

```
set execution_paths;
set tasks;
set subpaths;
set subpaths_task{subpaths} within tasks;
set subpaths_exec{execution_paths} within subpaths;
set services;
set dependencies_the_same;
set dependencies_the_same_task{dependencies_the_same} within tasks;
set dependencies_the_same_service{dependencies_the_same} within services;
set dependencies_not;
set dependencies_not_the_same_task{dependencies_not} within tasks;
set dependencies_not_the_same_service{dependencies_not} within services;
set dependencies_ser;
set dependencies_ser_task{dependencies_ser} within tasks;
set dependencies_ser_service{dependencies_ser} within services;
set properties;
set task_allowed{execution_paths} within tasks;
set services_task{tasks} within services;
set NTBproperties within properties;
set STBproperties within properties;
set LTBproperties within properties;
set SumProperties within properties;
set ProbabilityProperties within properties;
set MinProperties within properties;
set PathProperties within properties;
set AverageAttributes within properties;
set FunctionalityProperties within properties;

#parameters

param probability{execution_paths};
param preferences{properties};
param values{s in services, p in properties};
param critical_task{p in FunctionalityProperties, t in tasks};
param constraint_value{p in properties};
# in case of probability based attributes log(max_value) is given instead
param max_values{ p in properties, ep in execution_paths};
param min_values{ p in properties, ep in execution_paths};
param elem_number{execution_paths};
param number_of{dependencies_ser};

#variables

var ser{t in tasks, services_task[t]} binary;
var k{p in properties,ep in execution_paths};
var M{d in dependencies_the_same} binary;
```

```
var M1{d in dependencies_not} binary;
var M2{d in dependencies_ser} binary;

maximize overall_quality: sum{ep in execution_paths}
(probability[ep]*((sum{p in LTBproperties} preferences[p]*((k[p,ep]-
min_values[p,ep])/(max_values[p,ep]-min_values[p,ep])))+sum{p1 in
STBproperties} preferences[p1]*((max_values[p1,ep]-
k[p1,ep])/(max_values[p1,ep]-min_values[p1,ep]))));

subject to constraints1 {t in tasks}: sum{s in services_task[t]}
ser[t,s]=1;

subject to SumConstraints {p in SumProperties, ep in execution_paths}:
sum{t in task_allowed[ep], s in services_task[t]}
values[s,p]*ser[t,s]=k[p,ep];

subject to SumConstraintX {p in SumProperties, ep in execution_paths}: if
(constraint_value[p]>0) then sum{t in task_allowed[ep], s in
services_task[t]} values[s,p]*ser[t,s]<=constraint_value[p];

#ProbabilityProperties - in this case k[p,ep]=log(k')
subject to ProbabilityConstraints{p in ProbabilityProperties, ep in
execution_paths}: sum{t in task_allowed[ep], s in services_task[t]}
log(values[s,p])*ser[t,s]=k[p,ep];

subject to ProbabilityConstraintsX {p in ProbabilityProperties, ep in
execution_paths}: if (constraint_value[p]>0) then sum{t in
task_allowed[ep], s in services_task[t]} (-
1)*log(values[s,p])*ser[t,s]<=(-1)*constraint_value[p];

subject to MinConstraints {p in MinProperties, ep in execution_paths, t in
task_allowed[ep]}: sum{s in services_task[t]}
values[s,p]*ser[t,s]>=k[p,ep];

subject to PathConstraints {p in PathProperties, ep in execution_paths, su
in subpaths_exec[ep]}: sum{t in subpaths_task[su], s in services_task[t]}
values[s,p]*ser[t,s]<= k[p,ep];

subject to PathConstraintsX {p in PathProperties, ep in execution_paths,
su in subpaths_exec[ep]}:  if (constraint_value[p]>0) then sum{t in
subpaths_task[su], s in services_task[t]} values[s,p]*ser[t,s]<=
constraint_value[p];

subject to AverageConstraints {p in AverageAttributes, ep in
execution_paths}: (1/elem_number[ep])*(sum{t in task_allowed[ep], s in
services_task[t]} values[s,p]*ser[t,s])=k[p,ep];

subject to AverageConstraintsX {p in AverageAttributes, ep in
execution_paths}: if (constraint_value[p]>0) then
(1/elem_number[ep])*(sum{t in task_allowed[ep], s in services_task[t]} (1-
values[s,p])*ser[t,s])<=(1-constraint_value[p]);

subject to FunctionalityConstraints{p in FunctionalityProperties, ep in
execution_paths}: sum{t in task_allowed[ep], s in services_task[t]}
log(values[s,p])*critical_task[p,t]*ser[t,s]>=k[p,ep];

subject to dependent_constraint {d in dependencies_the_same}: sum{t1 in
dependencies_the_same_task[d], s1 in dependencies_the_same_service[d]: s1
in  services_task[t1]} ser[t1,s1]=2*M[d];

subject to dependent_not_constraint {d in dependencies_not}: sum{t1 in
dependencies_not_the_same_task[d], s1 in
dependencies_not_the_same_service[d]: s1 in services_task[t1]}
ser[t1,s1]=1-M1[d];

subject to dependent_constraint_services {d in dependencies_ser}: sum{t1
in dependencies_ser_task[d], s1 in dependencies_ser_service[d]: s1 in
services_task[t1]} ser[t1,s1]=number_of[d]*M2[d];
```

Then, the java simulation tool prepares the data set. It is then send to AMPL CPLEX solver:

```
option gentimes 1; #show time to generate each model component
option times 1; # show time taken in each model translation phase
model selection.mod; # load model file
```

```
data selection.dat; # load data file
option solver cplex; # use CPLEX to solve model
solve; # solve the model
# display solution values
display {i in 1.._nvars} _varname[i], {i in 1.._nvars} _var[i];
```

5.3.1 Values of non-functional properties

The approach proposed within this dissertation operates on various dimensions according to specific application requirements. It considers all dimensions in the weighted sum in the objective function under the hypothesis that the aggregated value for quality dimensions can be evaluated as sum, average, product, min or max of the corresponding quality dimension of services. Within the experiment, one property from each group has been selected, namely: cost, reputation, availability, throughput, execution duration, as indicated in the following table.

Table 29 Simulation values for selected properties (own study)

Group	Property	Simulation values
Sum based attributes	Cost	Proportional to reputation, availability, throughput and inversely dependent on duration
Average based attributes	Reputation	Uniform distribution in the range [0.8, 0.99]
Probability based	Availability	Uniform distribution in the range [0.85, 0.999] %
Min or max attribute	Throughput	Uniform distribution in the range [1,20] invocations per second
Critical path	Duration	Uniform distribution in the range [400, 8900] ms.

For the needs of experiment, values of properties of SWS have been mainly randomly generated according to the values reported in the literature.

The experiments conducted in (Tosic et al., 2004) provide information that a setup of Web services in a local network results in a response time of about 150 milliseconds. Thus, a value of 150 milliseconds may be regarded as the quickest service execution possible. (Gillmann et al., 2002) have evaluated the typical duration for the average turnaround times of activities in a workflow. Their evaluation (regarding automatic and non-interactive services) suggests that services usually take about 2 seconds to execute. In (Chafle et al., 2004), authors assume a uniform distribution of service execution time in the range between 0.5 sec and 8 sec. In turn, the test collection of 2507 Web services and their non-functional properties gathered by (Al-Masri and Mahmoud, 2007b) feature values from the range of 400 ms to 8900 ms. Therefore, during values generation a uniform distribution in the range of 400 milliseconds to 8900 milliseconds is assumed and generated using the Java Random library.

Regarding the availability of services, the work of Gillmann et al. (Gillmann et al., 2002) considers a typical downtime in the area of 20 minutes each day. This would result in an availability of 0.985% (Gillmann et al., 2002). In turn, in (Hiles, 2002) the author suggests that the interval of 0.95 to 0.999 should be considered, whereas the test collection gathered by (Al-Masri and Mahmoud, 2007b) feature the mean availability of about 0.82. Therefore, availability values have been randomly generated assuming a uniform distribution in the interval 0.85 and 0.999.

Following (Ardagna and Pernici, 2007) reputation has been determined in the similar way, considering the range of 0.8 to 0.99.

Regarding the throughput and cost, a study of an existing work would not result in many benefits for this simulation. In case of throughput, an inspiration was obtained from the test collection created by Masri – the 90% of services had throughput in the range of 1 to 20 invocations per second. Therefore, throughput values have been randomly generated assuming a uniform distribution in the interval 1 and 20.

Regarding cost, the expert knowledge to generate the appropriate values has been utilized. In addition, trade-offs between various properties have been formulated. It has been assumed the cost is proportional to service reputation, throughput and availability and inversely proportional to the execution time.

5.3.2 General experiment

To the author's best knowledge, the literature does not provide a dedicated evaluation about how many tasks a composition usually consists of. Some inspiration may be taken from the Gartner study, which informs about typical number of services used in companies and enterprises (Pezzini, 2005). According to the mentioned study, small companies deploy about 25 services on average while large enterprises may be deploying more than 1000 services. In addition, some evaluations of workflow management facilities can be found in the literature. For instance, in (Heinis et al., 2005) and (Cranford et al., 2001) the underlying number of tasks in workflows varies between 10 and 25.

Therefore, it has been assumed during simulation that a process consists of 20 to 50 tasks with various workflow patterns and to each task a random number from one to ten services has been assigned. Within the experiment, the following process structures have been considered: sequence, XOR, parallel branches; in various combinations. It has been assumed again that the loop structure is already unfolded.

After the composition specification and the candidates have been determined, a user profile along with the set of constraints is defined. In order to facilitate the requirements creation, a wizard-style tool may be used (Figure 34).

Each process model has been tested with various profiles and with or without a set of constraints. The set of weights has been randomly generated and weights were adjusted to sum to one.

In general, during the performed simulations problems with up to 13 execution paths, 50 tasks, up to 10 candidate SWS per task, and 5 global constraints have been considered.

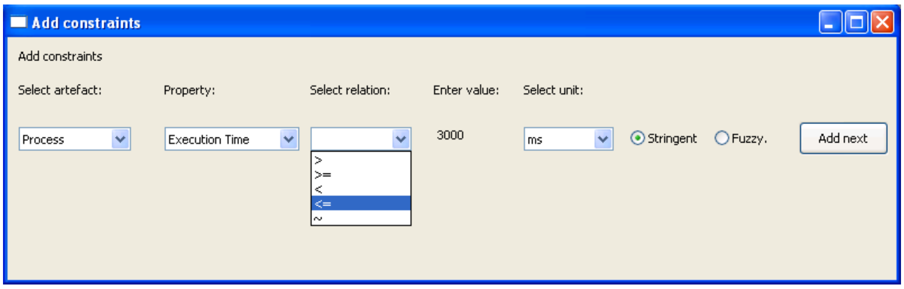


Figure 34 Interface to specify constraints

As already mentioned, the developed selection approach may be evaluated taking into account the number of services it operates on, the time of searching for the solution as well as the quality of the returned solution. Therefore, having the composition and the candidates with their description and enterprise profile, the simulation software performs selection methods on this setup (Figure 35). For each run, the software captures the resulting assignment of services, objective function, aggregated values for each execution path, computation time in seconds and memory usage.

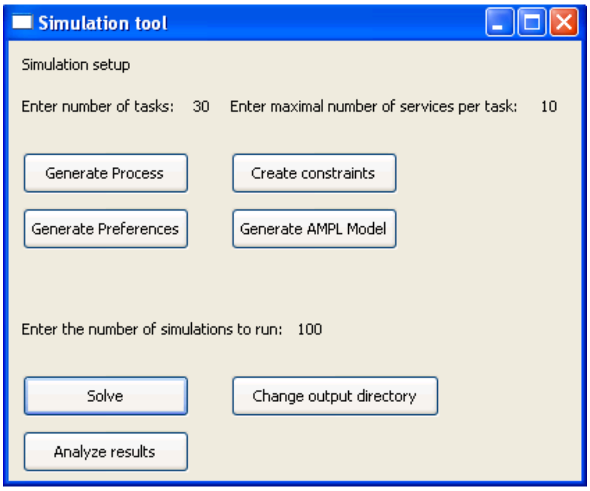


Figure 35 Simulation interface

In addition, in order to evaluate the effectiveness of the proposed approach, the software compares it with the solutions provided by the local optimization approach proposed in (Zeng et al., 2004). For every test case, first the local optimization algorithm has been performed, then the global optimization, including as global constraints the value of quality dimensions obtained by the local optimization. The comparison between local selection and the two-step selection with or without additional constraints follows.

Within the first simulation presented, the process consisting of no more than 20 tasks and up to 7 services that could be assigned to each task has been considered. The two-step selection slightly outperforms the local optimization.

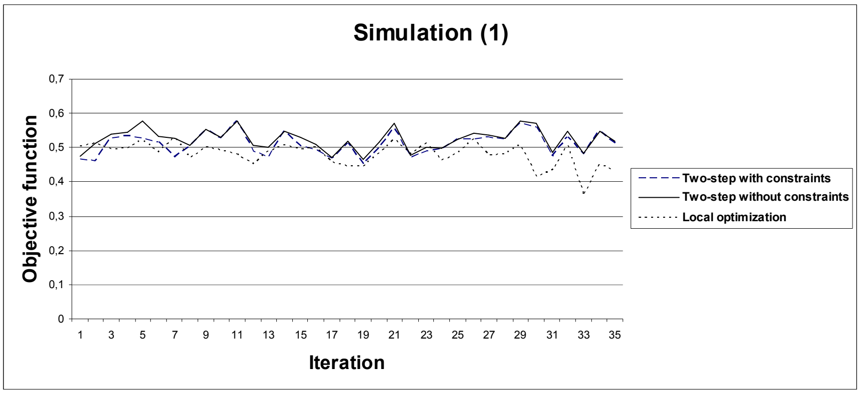


Figure 36 Simulation (1)

Within the second simulation, the process consisting of at least 25 tasks and a random number of services 1 to 7. The value of the objective function outperforms the local selection results.

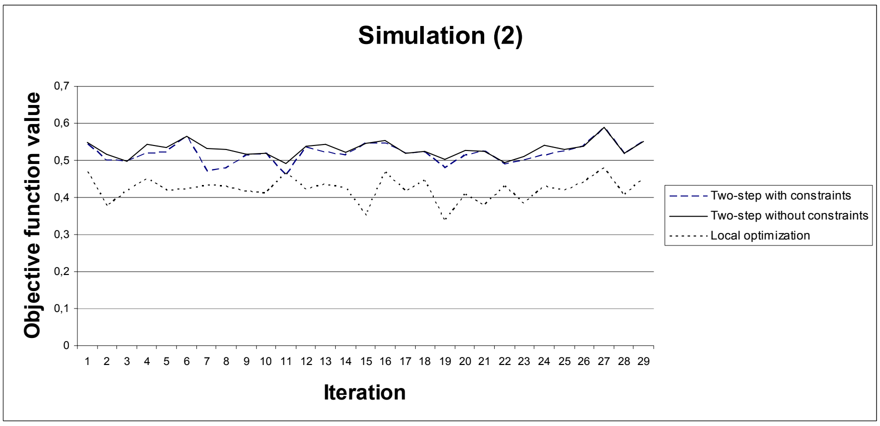


Figure 37 Simulation (2)

Within the third figure, the simulation on a more complex process consisting of a number of XOR structures is presented. To each task, a random number of 1 to 9 services has been assigned. As may be seen, the results are even more favourable than the results of the second simulation.

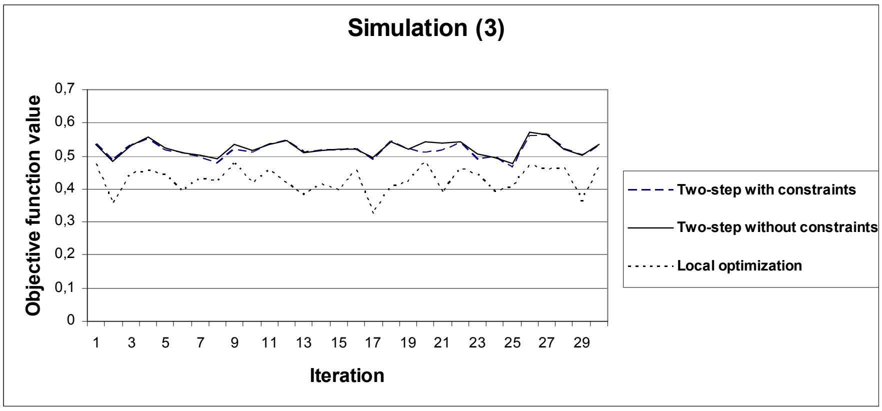


Figure 38 Simulation (3)

Finally, within the last simulation presented, the process with a complicated structure and approximately 300 services to select from was considered. The comparison between the local optimization and two-step selection is depicted in the following figure.

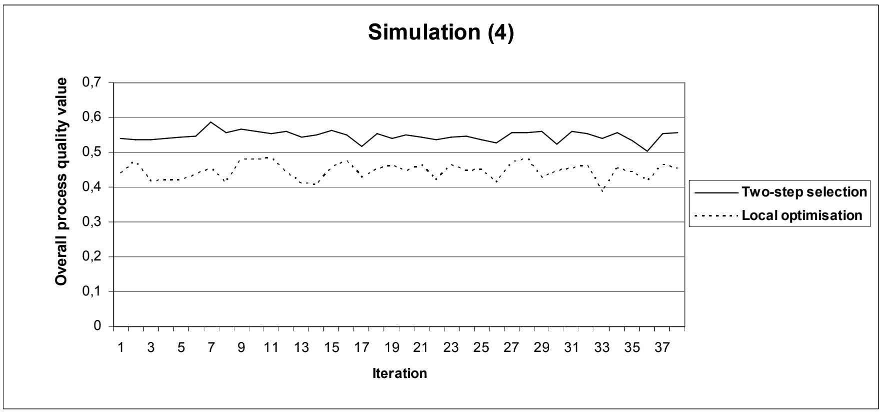


Figure 39 Simulation (4)

Results of all four simulations presented in the figures show that the global optimization gives significantly better results and the objective function is improved by 10-40%. The difference depends on the complexity of the process structure to be considered.

Note that a small improvement in the objective function corresponds to an improvement more significant in the value of the quality dimension. As an example Table 30 reports results of the solutions of some problem instances, where the optimization function includes the five selected attributes and corresponding weights are set as follows: cost 0.4, execution time 0.2, availability 0,1, throughput – 0,15 and reputation 0,15. Results show that an improvement of few percent units in the objective function, corresponds to an improvement in the price with the deterioration of the execution time. For instance a reduction in cost of 7 unit per process corresponds to 7 second extension of the execution duration of the process.

Table 30 Global and local optimization comparison

MILP optimization			Local optimization			Global vs. local optimization		
Obj. function	Cost	Execution time	Obj. function	Cost	Execution Time	Delta of obj. func.	Delta price	Delta exec. time
0,536	61,488	42974,15	0,475	63,306	42434,785	12,932	2,872	-1,271
0,485	56,598	43826,338	0,355	62,008	43443,543	36,625	8,726	-0,881
0,531	57,812	49761,15	0,446	63,117	42814,064	19,24	8,404	-16,226
0,559	56,598	43826,338	0,455	63,874	37988,991	22,653	11,392	-15,366
0,524	57,234	51561,45	0,444	60,414	42623,375	18,097	5,264	-20,97
0,508	54,459	46919,813	0,393	59,6	46333,896	29,015	8,624	-1,265

As the used solver allowed only for dealing with small and medium size problems, query answering has been realized in all cases in less than 3 seconds.

In addition, the next figure shows the interdependency between the number of offers and the memory usage. As the complexity is exponential, the reduction of variables to be taken into account within the selection problem by conducting the first step of the selection should allow for quite important reduction both in memory usage as well as execution time.

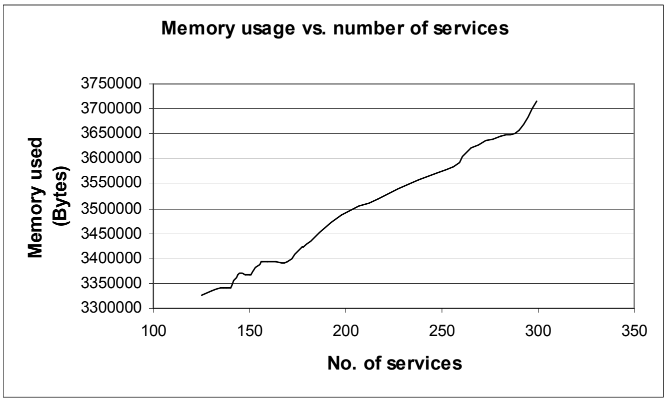


Figure 40 Memory usage

Generally, the evaluation shows that the selection mechanism might be applicable in scenarios where either the number of offers or the number of configurations per offer is moderate or where the selection performance is not crucial.

5.4 Evaluation of two-step selection model using the defined requirements

In the following table evaluation according to the criteria defined in Chapter 5 is presented.

Table 31 Evaluation of the proposed model using the defined requirements (own analysis)

Requirements	Level of fulfilment		Comments
	Other approaches	**Two step selection**	
Flexible, extensible and commonly accepted NFP model	Lack of commonly acceptable model	Partially supported.	The general and extensible information model has been proposed. It has been designed to be easily imported by other onotlogies. The author does not claim that the model will be commonly accepted. However, as it is based on best-known initiatives in this direction, it provides a general framework that may be easily used in various domains
Fair and open computation of NFPs values	Supported	Supported	The appropriate mechanisms have been put in place to gather the reliable information, namely the dynamic service profiling mechanism is utilised in order to gather the reliable and up-to-date information about service.
Inclusion of business properties	Partially supported.	Supported	The approach considers business and technical properties. The exemplary KPI and dependencies between partners has been defined.
Facilitation of transition from the business to the more technical layer	Not supported	Partially supported	The transition between various layers is supported by the model. In addition, a set of axioms has been defined in order to allow for easy decomposition of business properties to more technical layer.
Flexibility and fuzziness in the specification of requirements	Not supported	Supported	The mechanism offers support for various level of abstraction. The user may define various requirements and indicate whether they are stringent or not. Decomposition of constraints from higher to lower levels is supported.
Context	Not supported	Supported	Various contexts in the form of profiles (organization, business user, service, long and short term prognosis,) as well as groups of scenarios are considered.
Personalized, business properties based and constraints aware service selection	Partially supported	Supported	The selection mechanism proposed is NFP-based. Profiles suited to business needs have been defined. The specification of requirements is adjusted to the business needs and taken into account during the selection. The mechanism allows for personalization of the process.
Trust and contract-aware selection	Slightly supported	Supported	The model is prepared for inclusion of SLAs. The matter of trust is considered.
Dynamic binding and change support	Supported	Supported	The dynamic biding and runtime service selection is possible
Precision, scalability and computation efficiency of the mechanisms	Supported	Supported	The existing mechanisms either use the algorithms ensuring finding the optimal solution but with exponential complexity or use heuristics with polynomial complexity. The formulated model allows to identify the solutions in accordance with the defined requirements based on service characteristics. If the data is correct the mechanism returns precise results.

As may be concluded from the table, in comparison to the previous solutions the proposed mechanism better meets the expectations of potential participants of the e-marketplace in question. The greatest improvement is for the requirements from three to eight. This is credited to the integrated information model that partially bridges the gap between the business and more technical world of services.

As the model has been designed with extendibility in mind, it may be reused in various domains and it may be easily tailored to the specific needs of certain domains or service parks. Thanks to the structure of the proposed information model it is possible to take into account larger sets of parameters and properties. In addition, adding new properties does not affect the underlying selection model as long as one of the defined aggregation rules may be utilised.

Because the information model encompasses various profiles, it is possible to take into account during the selection process relationships and dependencies between organizations (such as partnerships, alliances, competitors) and between services . It is also possible to consider rules linked to tasks. In addition, more complex constraints and preferences may be assigned to various process artefacts. Introduction of fuzzy constraints guarantees that even if the end-user introduces severe constraints for the composed service execution, i.e., limited resources that set the problem close to unfeasibility conditions (e.g., limited budget or stringent execution time limit), a relaxed solution might be identified.

Second, the proposed algorithm is semantic-enabled and, therefore, allows the service selection process to work more flexibly and produce results that are more accurate and tailored to the users needs. This allows for automation of the selection process from the business perspective, as services are selected based on business expectations and not only IT perspective.

However, not all requirements could be fully addressed. Additional research is required to extend mappings between various layers and facilitate easy transition between various perspectives.

5.5 Summary

The conducted evaluation of the proposed model has shown that it fulfils the formulated requirements to the higher extent than the existing approaches. The application of the enhanced model and creation of more personalized approach allows for diminishing the possible solution space to be considered during the optimization process. In addition, the delivered results are better suited to the business needs and users expectations as well as to the business reality.

Therefore, the thesis formulated within the first chapter of this dissertation has been proved.

CONCLUSIONS AND OUTLOOK

6 Conclusions and outlook

This final chapter provides conclusions and outlook for this thesis. It describes the most pressing open issues that should be tackled by future and continuous research.

6.1 Dissertation summary

The SOA model brings several new benefits to software design and architecture by enabling re-use and sharing of components through dynamic discovery and binding of services. Service composition enables complex applications (i.e. business processes) to be put together in a variety of ways. Each possible assignment of services to tasks within a process, may feature different levels of process quality. Thus, there is a need to devise fast and efficient mechanisms that can be used for dynamic service selection among a set of service providers.

This dissertation has explored selection mechanisms that may be implemented on the B2B SWS e-marketplace. In the first part, the dissertation has presented an overview of the problem domain, namely: service oriented paradigm, services and service market. In order to provide a clear terminology, the general background and concept formation has been offered in the form of explicit definitions, characterizations, and classifications. This has been fundamental for the understanding of the thesis and it has revealed the diversity of interpretation that still exists among different authors in the respective topics. Based on the presented concepts and mechanisms, the context for this thesis has been set by outlining the anticipated development towards B2B SWS e-marketplace, the establishment of composite service providers, and the need for automated business-oriented service selection.

In addition, current selection approaches have been presented and discussed. The thesis has acknowledged that comprehensive research has been conducted by different groups in the area of the selection of Web services and Semantic Web services. However, the approaches so far have focused mainly on technical issues and have almost completely ignored the business aspects as well as service-provisioning context. In fact, none of the existing approaches addresses dependencies between services as well as relations between business partners.

In the second part of the thesis, a novel, integrated selection approach, namely two-step business-conditions aware selection mechanism has been proposed. It provides an automated support for B2B SWS e-marketplaces and it incorporates the business perspective and service provisioning conditions into its model. Thus, an e-marketplace may use the proposed mechanism to optimize its suggestions and recommendations. The fundamental elements of created selection model have been introduced and presented in detail. These are as follows:

- selection model – a conceptual framework which outlines the components which are necessary for an automated selection for the needs of B2B SWS e-marketplace,
- selection information model – an integrated information model which encapsulates all information necessary to perform efficient selection, as well as inclusions of relationships and dependencies that exist between relevant artefacts, data validation restrictions as well as individual aggregation formulas,
- selection mechanism – a selection technique based on the mixed integer logic programming used in order to identify the optimal, meeting business requirements assignment of services to a business process.

In the third part of the dissertation, the selection approach has been validated. First, the developed information model has been evaluated using a set of formal competency questions. In addition, it has been demonstrated how the proposed selection approach can be applied to use case scenarios which were developed based on two European projects the author participated in, namely ASG and SUPER. A very prototypical implementation displays the presented concepts and mechanisms. In addition, the proposed model has been checked against the identified requirements of business users. The evaluation has proved the validity of presented research outcomes and in consequence also the thesis of this dissertation.

Within these three parts, the thesis has addressed the research goals and problems identified in section 1.2. According to the research methodology followed within this thesis and mentioned in section 1.3, the seven guidelines outlined by (Hevner et al., 2004) have been addressed in research conducted for this dissertation.

6.2 Main contribution

The major contribution is two fold. On the one hand, the analysis of the expectations of marketplace participants has been carried out and a set of ten requirements has been identified. To the author's best knowledge, there are no similar studies performed. The most important conclusion from the performed analysis is that business people are interested in the technology that supports fast adaptation and personalization of business processes to meet the expectations of their clients. This requires that the technology in question needs to be adjusted to business reality i.e. business terms and service provisioning conditions. Although, the performed analysis cannot claim to be statistically valid, knowing the expectations of potential users may help shaping the directions of further research in the field of SWS and their e-marketplaces.

On the other hand, a new two-step business conditions aware service selection approach has been proposed and shown to be accurate, efficient and reliable. Experimental results have demonstrated that the proposed service selection yields good results that are in accordance with the expectations of business users. The proposed selection mechanism operates on an abstract service description. This facilitates its reusability and guarantees that it may be used together with most existing languages.

For the needs of the selection, an integrated information model has been proposed bridging partially the gap between the business and more technical world of services. As the model has been designed with extendibility in mind, it is possible to reuse it in various domains. It may be also used to support other interactions on the SWS e-marketplaces like service discovery, composition or negotiations. Second, the proposed algorithm is general and semantic-enabled by offering support for the semantic similarity among various properties advertised by providers and the ones required by users. This allows the service selection process to work more flexibly and produce results that are more accurate.

In addition, thanks to more sophisticated and complex information model encompassing also various profiles, it is possible to take into account during the selection process relationships and dependencies between organizations and between services as well as various rules. In addition, more complex constraints and preferences may be assigned to various process artefacts. This allows for automation of the selection process from the business perspective, as e.g. services are selected based on business expectations and only from trusted parties (or the parties that they trust if the transitive trust mechanism is introduced).

Thanks to the structure of the proposed information model, it is possible to take into account larger sets of parameters and properties. Moreover, adding new properties does not affect the underlying selection mechanism as long as one of the defined aggregation rules may be utilised.

The division of the selection process into two steps and the rich information model allows in a more efficient way to narrow down the list of available services that should be considered in a solution. Therefore, the computation is less demanding because of the smaller number of variables taken into account within the MILP problem. Thus, whereas the computational complexity of the second step is exponential, the computational complexity of the first step, taking into account also decomposition of structure and constraints, is only polynomial.

The second step of the selection offers support for the statefull Web services as well as existing formal or informal dependencies between various services or their providers. To the author's best knowledge this has not been fully supported by any existing model.

The proposed model allows to consider both global and local constraints, as well as it optimizes the global process model at once and not by addressing each of the execution paths separately. Although, the optimization is only statistical (i.e. it takes into account the probability of execution of conditional branches as well as the number of cycles), the results should be satisfactory and the re-optimization can take place whenever new data is obtained or something changes in the environment.

Introduction of fuzzy constraints guarantees that even if the end-user introduces severe constraints for the composed service execution, i.e., limited resources that set the problem close to unfeasibility conditions (e.g., limited budget or stringent execution time limit), relaxed solution might be identified.

Therefore, the research conducted within this dissertation has set the basis for overcoming the limits of the previous approaches to Semantic Web services selection and meet business users' expectations. The aim to discover the optimum mapping between an abstract process and a composite Semantic Web service that implements the abstract description, such that the overall quality of the process perceived by the user is maximized under many constraints including business expectation, has been realized.

6.3 Further research

Although, the results presented in this thesis, are quite promising, there are many challenges and issues that still should be addressed.

The selection mechanism is not a perfect and omniscience solution. Uncertainty, incomplete knowledge and competition between providers resulting in inadequate SWS representation pose many challenges that demand more intelligence to be integrated into selection strategy. The adequacy of the results returned by the selection mechanism depends strongly on the quality of the data it operates on. If the profiles of services are not accurate, or service discovery returns results that do not match the requested by a given task functionality, then the assignment retuned by the selection mechanism (although theoretically correct) will not meet the expectations of a business user.

One of the main challenges that from the authors' perspective have been only partially addressed is bridging the gap between business and IT worlds. This is visible in two aspects. First, the quantification of the business value still remains a challenge. The information model takes into account only the most common metrics that are quantifiable in some way. Therefore, the issue of mapping between not easy quantifiable business values and more technical service layer needs to be more thoroughly investigated. The decomposition rules included in the information model should be regarded only as a very first step in this direction.

Moreover, more complex analysis of service pricing and provisioning concept should be performed. The currently introduced analysis of the cost of a service has been simplified for the needs of the discussion or it has been assumed that the dynamic service profiling mechanism conducts the necessary unification.

Although, the relaxation of constraints allows for identification of solution in case too severe constraints have been applied, another mechanism, namely negotiation, should be included within the selection process. Thus, in case no feasible solution can be identified, automatic agent-based negotiations should be performed.

Another important issue is the matter of trust. The selection mechanism operates on company sensitive data i.e. organization and business user profile encompassing information on strategies and preferences. In addition, the process model used to implement a business process is guarded information the organizations will not be willing to share outside. Therefore, as already mentioned, the e-marketplace should be a trusted party as only then the users will be willing to share this information. In addition, appropriate mechanism ensuring the data privacy should be implemented.

The other issue is the facilitation of interactions with ontologies to business users. For instance, a business user would require an appropriate interface that would allow him to model a process, compose a process, add additional constraint to the process as well as express other requirements. Although, a tool partially supporting it has been already developed within the SUPER project, still an additional effort is needed to make the interfaces and interactions with SWS more user friendly.

As underlined by (Vu et al., 2005) semantic service selection mechanisms will play an essential role in service-oriented architectures because e-business applications want to use services that most accurately meet their requirements. The prerequisite for semantic service selection is the existence and application of SWS. However, as mentioned in the dissertation, SWS technology has not yet reached a state of full maturity. Ontologies have become popular largely because of what they promise: a shared and common understanding that reaches across people and application systems (Fensel et al., 2001). However, at present there are still a lot of problems unresolved that need to be addressed. Some of these drawbacks are as follows (Haniewicz et al., 2008):

- too complex service description as well as limits of the ontologies and reasoning,
- lack of recommendations and standards that may be followed,
- lack of tools that would sufficiently support different WS and SWS languages.
- inclusion of business perspective.

SWS technology is gaining momentum; it is an object of the intensive research. Yet, up to now, the developed technologies have rarely been applied in industry. Although the impression may be that the industry is not really eager to adopt heavy weight solutions such as SWS technology and is looking for something simpler, the interviewees coming from industry stated that the business world sees a potential in the SWS technology, if the drawbacks mentioned above will be first considered.

Although there are many antagonists of SWS idea and the WS technology in general, following the Cringley's law, "a short term adoption of new technologies never occurs as quickly as we expect but the long-term impact is far greater than we realize".

7 References

1. AALST, W. V. D. (2003) Don't go with the flow: Web services composition standards exposed. IEEE Intelligent Systems, Vol. Jan/Feb 2003, pp. 72-76, January.
2. AALST, W. V. D., HOFSTEDE, A. T., KIEPUSZEWSKI, B. & BARROS, A. (2003) Workflow Patterns. Distributed and Parallel Databases, Vol. 14, Issue:3, pp. 5-51.
3. ABRAMOWICZ, W. (2008) Filtrowanie informacji, pp. 626, Poznań, Wydawnictwo Akademii Ekonomicznej (AE) w Poznaniu, ISBN: 978-83-7417-315-5.
4. ABRAMOWICZ, W., EKELHART, A., FENZ, S., KACZMAREK, M., TJOA, A. M., WEIPPL, E. R. & ZYSKOWSKI, D. (2007a) Security Aspects in Semantic Web Services Filtering In Proceedings Of 9th International Conference on Information Integration and Web-based Applications and Services, GABRIELE KOTSIS, D. T., ERIC PARDEDE, ISMAIL K. IBRAHIM (Ed.), Vol., Austrian Computer Society, pp. 21-31, Jakarta.
5. ABRAMOWICZ, W., FILIPOWSKA, A., KACZMAREK, M., KACZMAREK, T., KOWALKIEWICZ, M., RUTKOWSKI, W., WIELOCH, K. & ZYSKOWSKI, D. (2006a) Service interdependencies: insights into use cases for service compositions. In Proceedings Of IFIP - Professional Practice in Artificial Intelligence, Lecture Notes in Computer Science, Springer Verlag, pp. 101-110, ISBN: 0-387-34655-4.
6. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M., PALMA, R. & ZYSKOWSKI, D. (2008a) NFP ontology for discovery and sharing Web services in distributed registries. In Proceedings Of 22nd International Conference on Advanced Information Networking and Applications: Workshops, IEEE, pp. 1416-1421, Gino-wan, ISBN: 978-0769530963.
7. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2006b) Automatic Web Services Interactions -Requirements, Challenges and Limits from the F-WebS System Perspective. In Proceedings Of International Conference on Next Generation Web Services Practices, ABRAHAM, A. & HAN, S. Y. (Eds.), IEEE Press, pp. 143-148, Seoul, Korea, ISBN: 0-7695-2664-0.
8. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2006c) Filtering of Semantic Web services with F-WebS system. In Proceedings Of The Semantic Web: ASWC 2006 Workshop, LI, G., LIANG, Y. & RONCHETTI, M. (Eds.), Jilin University Press, pp. 317-324, Beijing, ISBN: 7-5601-3528-5.
9. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2007b) Architecture for Web Services Filtering and Clustering. In Proceedings Of Second International Conference on Internet and Web Applications and Services (ICIW 2007), IEEE Press, pp. 18-23, Mauritius, ISBN: 0-7695-2844.
10. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2008b) E-Marketplace for Semantic Web Services. Service-Oriented Computing - ICSOC 2008, 6th International Conference, Sydney, Australia, December 1-5, 2008. Springer Berlin/Heidelberg.
11. ABRAMOWICZ, W., KACZMAREK, M., KOWALKIEWICZ, M. & ZYSKOWSKI, D. (2005a) A survey of QoS computation for Web services Profiling. In Proceedings Of 18th International Conference on Computer Applications in Industry and Engineering (ISCA), DASCALU, S. (Ed.), ISCA (International Society for Computers and their Applications), pp. 192-197, Honolulu, USA, ISBN: 1-880843-57-9.
12. ABRAMOWICZ, W., KACZMAREK, M., KOWALKIEWICZ, M. & ZYSKOWSKI, D. (2006d) Architecture for Service Profiling. In Proceedings Of SCW '06: Proceedings of the IEEE Services Computing Workshops, MALU CASTELLANOS, J. Y. (Ed.), IEEE Computer Society, pp. 121-127, September, ISBN: 0-7695-2681-0.
13. ABRAMOWICZ, W., KACZMAREK, M. & ZYSKOWSKI, D. (2006e) Duality in Web Services Reliability. In Proceedings Of the Advanced International Conference on

Telecommunications and International Conference on Internet and Web Applications and Services 2006 (AICT/ICIW 2006) DINI, P., LORENZ, P., ROMAN, D. & FREIRE, M. (Eds.), IEEE Press, pp. 165-170, Guadeloupe, French Caribbean, ISBN: 0-7695-2522-9.

14. ABRAMOWICZ, W., KACZMAREK, M. & ZYSKOWSKI, D. (2008c) Profiling of WS to Measure and Verify their Non-Functional Properties. Managing Web Service Quality, Measuring Outcomes and Effectiveness. IGI Publishing, ISBN: 978-160566042-4.

15. ABRAMOWICZ, W., KACZMAREK, M., ZYSKOWSKI, D., GWIZDALA, J. & GODLEWSKA, A. (2005b) Application-oriented Web services filtering. In Proceedings Of International Conference on Next Generation Web Services Practices, ABRAHAM, A., HAN, S. Y., DU, D. H.-C. & PAPRZYCKI, M. (Eds.), IEEE Press, pp. 63-68, Seoul, Korea, August, ISBN: 0-7695-2452-4.

16. ADAMS, J., KOUSHIK, S., VASUDEVA, G. & GALAMBOS, G. (2001) Patterns for e-business - a strategy for reuse. IBM Press.

17. AGARWAL, S. & LAMPARTER, S. (2005) User Preference based Automated Selection of Web Services Composition. In Proceedings Of ICSOC Workshop on Dynamic Web Processes, KUNAL VERMA, A. S., MICHAL ZAREMBA, CHRISTOPH BUSSLER (Ed.), pp. 1-12, Amsterdam, Netherlands, December.

18. AGARWAL, V., DASGUPTA, K., KARNIK, N., KUMAR, A., KUNDU, A., MITTAL, S. & SRIVASTAVA, B. (2005) A service creation environment based on end to end composition of Web services. In Proceedings Of 14th International Conference on World Wide Web, ACM Press, pp. 128-137, Chiba, Japan, ISBN: 1-59593-046-9.

19. AIB, I., AGOULMINE, N. & PUJOLLE, G. (2004) Capturing adaptive B2B Service Relationships Management through a generalized Service Information Model. In Proceedings Of 11th Workshop of the HP Open View University Association (HPOVUA), pp. 1-16, Paris.

20. AL-MASRI, E. & MAHMOUD, Q. H. (2007a) Discovering the best Web service. In Proceedings Of 16th International conference on World Wide Web, ACM Press, pp. 1257-1258, Banff, Alberta, Canada, ISBN: 978-1-59593-654-7.

21. AL-MASRI, E. & MAHMOUD, Q. H. (2007b) QoS-based Discovery and Ranking of Web Services. In Proceedings Of IEEE 16th International Conference on Computer Communications and Networks (ICCCN), IEEE, pp. 529-534.

22. ANAND, S., PADMANABHUNI, S. & GANESH, J. (2005) Perspectives on service oriented architecture. In Proceedings Of International Conference on Services Computing, IEEE, Orlando, Florida, USA, July 2005.

23. ANTONIOU, G. & VAN HARMELEN, F. (2003) Web Ontology Language: OWL. Handbook on Ontologies in Information Systems. Springer-Verlag.

24. ARDAGNA, D. & PERNICI, B. (2005) Global and Local QoS Guarantee in Web Service Selection. In Proceedings Of Business Process Management Workshop, pp. 32-46.

25. ARDAGNA, D. & PERNICI, B. (2007) Adaptive Service Composition in Flexible Processes. IEEE Transactions on Software Engineering, Vol. 33, Issue:6, pp. 369-384.

26. ARPINAR, I. B., ALEMAN-MEZA, B., ZHANG, R. & MADUKO, A. (2004) Ontology-Driven Web Services Composition Platform. In Proceedings Of IEEE International Conference on E-Commerce Technology (CEC'04), IEEE Computer Society, pp. 146-152, ISBN: 0-7695-2098-7.

27. BAKOS, J. Y. (1997) Reducing buyer search costs: implications for electronic marketplaces, Management Science, Vol. 43, No. 12, December 1997

28. BALKE, W. T. & WAGNER, M. (2003) Towards personalized selection of Web services. In Proceedings Of 12th International World Wide Web Conference, Budapest, Hungary.

29. BARROS, A., DUMAS, M. & BRUZA, P. (2005) The move to Web services ecosystem. BPTrends, November 2005.

30. BELLWOOD, T., CLEMENT, L. E., D., HATELY, A., HONDO, M., HUSBAND, Y. L., JANUSZEWSKI, K., LEE, S., MCKEE, B., MUNTER, J. & VON RIEGEN, C. (2002) UDDI version 3.0. Technical report, July.

31. BENATALLAH, B., DUMAS, M., FAUVET, M.-C. & RABHI, F. A. (2003) Towards patterns of Web services composition. Patterns and skeletons for parallel and distributive computing. Springer-Verlag.

32. BENATALLAH, B., DUMAS, M. & SHENG, Q. Z. (2005a) Facilitating the rapid development and scalable orchestration of composite services. Distributed and Parallel Databases, Vol. 17.

33. BENATALLAH, B., HACID, M.-S., LEGER, A., REY, C. & TOUMANI, F. (2005b) On automating Web service discovery. VLDB Journal, Vol. 14, pp. 84-96.

34. BERBNER, R., SPAHN, M., REPP, N., HECKMANN, O. & STEINMETZ, R. (2006) Heuristics for QoS-aware Web Service Composition. In Proceedings Of International Conference on Web Services, September.

35. BERNERS-LEE, T., HENDLER, J. & LASSILA, O. (2001) The semantic web. Scientific American.

36. BICHLER, M. & LIN, K. J. (2006) Service oriented computing. Computer, Vol. 39, Issue:3, pp. 99-101.

37. BIH, J. (2006) Service oriented architecture a new paradigm to implement dynamic e-business solutions. Ubiquity, ACM Press, Vol. 7, Issue:30, August 8, 2006.

38. BILGIN, A. & SINGH, M. (2004) A DAML-Based Repository for QoS-Aware Semantic Web Service Selection. In Proceedings Of IEEE International Conference on Web services (ICWS'04), San Diego, USA.

39. BISHOP, B. & FISCHER, F. (2008) IRIS - Integrated Rule Inference System. In Proceedings Of Proceedings of the 1st Workshop on Advancing Reasoning on the Web: Scalability and Commonsense (ARea2008) hosted by the 5th European Semantic Web Conference (ESWC-08), Tenerife, Spain, June.

40. BOEHM, B. W. & SULLIVAN, K. J. (2000) Software economics: a roadmap. In Proceedings Of International Conference on Software Engineering, ACM Press, New York, USA, pp. 319-343, Limerick, Ireland.

41. BORN, M., DRUMM, C., MARKOVIC, I. & WEBER, I. (2007) SUPER - Raising Business Process Management Back to the Business Level. ERCIM News - Special Theme on Service-Oriented Computing.

42. BRAMBILLA, M., FACCA, F. M., CELINO, I., CERIZZA, D. & VALLE, E. D. (2007) Model driven design and development of semantic Web service applications. ACM Transactions - Internet Technologies, Vol. 8, Issue:1, pp. 3, ISSN: 1533-5399.

43. BRELAGE, C. & DOMINGUE, J. (2008) Layers in Semantic Business Process Management. EU SUPER Project Internal Materials, SAP Research, Karlsruhe.

44. BRUIJN, J. D., EHRIG, M., FEIER, C., MARTÍNS-RECUERDA, F., SCHARFFE, F. & WEITEN, M. (2006) Ontology Mediation, Merging, and Aligning. IN JOHN DAVIES, R. S., PAUL WARREN (Ed.) Semantic Web Technologies. John Wiley & Sons, Ltd, ISBN: 9780470025963.

45. BULTAN, T., FU, X., HULL, R. & SU, J. (2003) Conversation specification: a new approach to design and analysis of e-service composition. In Proceedings Of World Wide Web Conference 2003, pp. 403-410.

46. BUSSLER, C., MAEDCHE, A. & FENSEL, D. (2002) A Conceptual Architecture for Semantic Web Enabled Web Services. ACM Special Interest Group on Management of Data, Vol. 31, Issue:4, December.

47. CANFORA, G., DI PENTA, M., ESPOSITO, R. & VILLANI, M. L. (2005) QoS-Aware Re-planning of Composite Web Services. In Proceedings Of IEEE international Conference on Web Services, ICWS, IEEE Computer Society, pp. 121-129, Washington, DC, July 11 - 15.

48. CARDELLINI, V., CASALICCHIO, E., GRASSI, V. & PRESTI, F. L. (2007) Scalable service selection for Web service composition supporting differentiated QoS classes. Technical Report RR-07.59, Italy, Dip. di Informatica, Sistemi e Produzione, Università di Roma, February.

49. CARDOSO, J. (2002) Quality of Service and Semantic Composition of Workflows. University of Georgia.

50. CASATI, F., CASTELLANOS, M., DAYAL, U. & SHAN, M. C. (2004) Probabilistic, context-sensitive and goal-oriented service selection. In Proceedings Of ICSOC'04, ACM Press, New York, USA, November 2004.

51. CASATI, F., ILNICKI, S. & JIN, L. (2000) Adaptive and dynamic service composition in EFlow. In Proceedings Of 12th International Conference on Advanced Information Systems Engineering (CAiSE), Springer Verlag, Stockholm, Sweden, June.

52. CASATI, F. & SHAN, M. C. (2001a) Dynamic and Adaptive Composition of E-Services. Information Systems, Vol. 26, Issue:3, pp. 143-162, May.

53. CASATI, F. & SHAN, M. C. (2001b) Models and languages for describing and discovering e-services. In Proceedings Of SIGMOD Conference, pp. 626, Santa Barbara, California, USA.

54. CHAFLE, G. B., CHANDRA, S., MANN, V. & NANDA, M. G. (2004) Decentralized orchestration of composite web services. In Proceedings Of 13th international WWW conference on Alternate track papers & posters, ACM, pp. 134-143, New York, USA.

55. CHANDERASEKARAN, S., MILLER, J. A., SILVER, G. S., ARPINAR, I. B. & SHETH, A. P. (2003) Performance Analysis and Simulation of Composite Web services. Electronic Market, Vol. 13, Issue:2, June.

56. CHANNABASAVAIAH, K., HOLLEY, K. & TUGGLE, E. (2003) Migrating to a service-oriented architecture. IN IBM (Ed.), IBM.

57. CHIU, D. K. W., POON, J. K. M., LAM, W. C., TSE, C. Y., SU, W. H. T. & POON, W. S. (2005) How ontologies can help in an e-marketplace. In Proceedings Of 13th European Conference on Information Systems, Information Systems in Rapidly Changing Economy (ECIS 2005), Regensburg, Germany, May 26-28, 2005.

58. CHONG, F. & CARRARO, G. (2006) Architecture Strategies for Catching the Long Tail. Microsoft.

59. CLARO, D. B., ALBERS, P. & HAO, J. K. (2005) Selecting Web Services for Optimal Composition. In Proceedings Of IEEE ICWS Workshop, Orlando.

60. COCK, M. D., CHUNG, S. & HAFEEZ, O. (2007) Selection of Web Services with Imprecise QoS Constraints. Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society.

61. CRANFORD, J., MUKKAMALA, R. & ATLURI, V. (2001) Modelling and Evaluation of Distributed Workflow Algorithms. In Proceedings Of World Multi-conference on Systems, Cybernetics and Informatics: Information Systems Development, IIIS, pp. 183-188, Orlando, Florida, USA, July.

62. DAN, A., DAVIS, D., KEARNEY, R., KELLER, A., KING, R., KUEBLER, D., LUDWIG, H., POLAN, M., SPREITZER, M. & YOUSSEF, A. (2004) Web services on demand: WSLA-driven automated management. IBM Systems Journal, Vol. 43, Issue:1.

63. DAN, A., LUDWIG, H. & PACIFICI, G. (2008) Web services Differentiation with Service Level Agreements. http://citeseer.ist.psu.edu/697275.html, accessed on 12.03.2008.

64. DAY, J. & DETERS, R. (2004) Selecting the Best Web Service. In Proceedings Of 2004 Conference of the Centre for Advanced Studies on Collaborative Research (CASCON), pp. 293–307, Markham, Ontario, Canada, October, ISBN: 1705-7345.

65. DENG, S., WU, Z. & LI, Y. (2004) ASCEND: a framework for automatic service composition and execution in dynamic environment In Proceedings Of International

Conference Systems, Man and Cybernetics, Vol. 4, IEEE, pp. 3457-3461, Netherlands, 10-13 October, ISBN: 0-7803-8566-7.

66. DIJKMANN, R. M. & DUMAS, M. (2004) Service-Oriented Design: A Multi-Viewpoint Approach. International Journal of Cooperative Information Systems (IJCIS), Vol. 13(4), pp. 337-368.

67. DIMITROV, M., SIMOV, A., STEIN, S. & KONSTANTINOV, M. (2007) A BPMO based semantic business process modelling environment. In Proceedings Of Workshop on Semantic Business Process and Product Lifecycle Management (SBPM-2007), M. HEPP, K. H., D. KARAGIANNIS, R. KLEIN, N. STOJANOVIC (Ed.), Vol. 251, CEUR-WS, Innsbruck, April 7, ISBN: 1613-0073.

68. DIP (2007) Data, Information, and Process Integration with Semantic Web Services – Technical Report of the 6th FP EU DIP project.

69. DIPANJAN, C., SURAJ KUMAR, J., ARCHAN, M. & AMIT, A. N. (2005) Middleware Architecture for Evaluation and Selection of 3rd-Party Web Services for Service Providers. In Proceedings Of EEE International Conference on Web Services, IEEE Computer Society, pp. 647-654, ISBN: 0-7695-2409-5.

70. DOMINIK KUROPKA & WESKE, M. (2008) Implementing a semantic service provision platform. Concepts and Experiences. Wirtschaftsinformatik, Vol. 2008-1, pp. 16-24.

71. DUMAS, M., O'SULLIVAN, J., HERAVIZADEH, M., EDMOND, D. & HOFSTEDE, A. T. (2003) Towards a Semantic Framework for Service Description. Vol.

72. DUSTDAR, S. (2005) Web services, Technical presentation.

73. EADES, P. (2008) How to Get a PhD in Computer Science. Keynote speech at Seventh Macquarie Computing Mini-Conference (MCMC-08), Sydney, Australia, Macquarie Univeristy, June 16-20.

74. EENOO, C. V., HYLOOZ, O. & KHAN, K. M. (2005) Addressing non-functional properties in software architecture using ADL. In Proceedings Of 6th Australian Workshop on Software and Systems Architectures, pp. 6-13, Brisbane, Australia.

75. ESTIER, T., MICHEL, B. & REINHARD, O. (2006) Modeling Services using Contracts. In Proceedings Of Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD'06) held in conjunction with the 18th Conference on Advanced Information Systems (CAiSE'06), J. KROGSTIE (Ed.), Namur University Press, Namur, Belgium, pp. 445-452, Luxembourg, ISBN: 2870375255.

76. EVENSON, M., SCHREDER, B., FRANKOWSKI, J., RUBACH, P. & SKORCZYK, M. (2008) CRM and billing prototypes, version 1.2. Technical Report Deliverable 9.2, SUPER Research Project, November.

77. FAHRINGER, T., KRAUSE, H., KUROPKA, D., MAYER, H., OCAMPO, A., SCHREDER, B., STAAB, S., TROEGER, P., WAHLER, A. & ZAREMBA, M. (2007) ASG technology - advantages and disadvantages, exploitation possibilities and its business impact. ASG Research Project Deliverable.

78. FAN, J. & KAMBHAMPATI, S. (2005) A snapshot of public web services. ACM SiGMOD Record, Vol. 34, Issue:1 (March 2005), pp. 24-32, March 2005, ISSN: 0163-5808.

79. FANTINI, P., SAVOLDELLI, A., MILANESI, M., CARIZZONI, G., KOEHLER, J., STEIN, S., ANGELI, R., HEPP, M., ROMAN, D., BRELAGE, C. & BORN, M. (2007) Semantic Business Process Life Cycle. Deliverable of SUPER Project, IST 026850.

80. FARREL, J. & LAUSEN, H. (2006) Semantic Annotations for WSDL. DERI Innsbruck.

81. FEDORUK, A. & DETERS, R. (2002) Improving Fault-tolerance by Replicating Agents. In Proceedings Of First International Joint Conference on Autonomous Agents And Multiagent Systems, ACM, pp. 737-744, Bologna, Italy, ISBN: 1-58113-480-0.

82. FELDMAN, S. (2000) E-business: Electronic Marketplaces. IEEE Internet Computing, Issue:4, pp. 93-95.

83. FENSEL, D., BUSSLER, C. & MAEDCHE, A. (2002) Semantic Web enabled Web services. In Proceedings Of International Semantic Web conference, pp. 1-2, Sardinia, Italy, June.

84. FENSEL, D., HORROCKS, I., HARMELEN, F. V., MCGUINNESS, D. L. & PATEL-SCHNEIDER, P. F. (2001) OIL: An ontology infrastructure for the Semantic Web. IEEE Intelligent systems. Special Issue on the Semantic Web, Vol. 16, Issue:2.

85. FERNANDEZ-LOPEZ, M. & GOMEZ-PEREZ, A. (2002) Overview and analysis of methodologies for building ontologies. Knowledge Engineering Review, Vol. 17, Issue:2, pp. 129-156, ISSN: 0269-8889.

86. FODOR, O. & WERTHNER, H. (2005) Harmonise: A step towards an interoperable e-tourism marketplace. International Journal of Electronic Commerce, Vol. 9, Issue:2

87. FRIESEN, A. (2007) Semantic Web Service Discovery and Selection in B2B Integration Scenarios. Reasoning Web. Springer Berlin/Heidelberg, ISBN: 0302-9743.

88. FRIESEN, A. & NAMIRI, K. (2006) Towards semantic service selection for B2B integration. Workshop proceedings of the sixth international conference on Web engineering, Palo Alto, California, ACM.

89. GALIZIA, S., GUGLIOTTA, A. & DOMINGUE, J. (2007) A Trust Based Methodology for Web Service Selection. In Proceedings Of International Conference on Semantic Computing (ICSC 2007), IEEE Press, pp. 193-200, Irvine, CA, USA, September, ISBN: 978-0-7695-2997-4.

90. GAO, Y., NA, J., ZHANG, B., YANG, L. & GONG, Q. (2006) Optimal Web Services Selection Using Dynamic Programming. In Proceedings Of 11th IEEE Symposium on Computers and Communications (ISCC'06), IEEE Computer Society, pp. 365-370, ISBN: 0-7695-2588-1.

91. GILLMANN, M., WEIKUM, G. & WONNER, W. (2002) Workflow management with service quality guarantees. In Proceedings Of 2002 ACM SIGMOD international conference on Management of data, ACM Press, pp. 228-239, Madison, Wisconsin, USA, June, ISBN: 1-58113-497-5.

92. GOLD, N., MOHAN, A., KNIGHT, C. & MUNRO, M. (2004) Understanding service-oriented software. Software, IEEE, Vol. 21, Issue:2, pp. 71-77.

93. GRALLA, P. (2005) The Web services advisor: The ROI of Web services and SOA. 04.12.2005.

94. GREY, W., OLAVSON, T. & SHI, D. (2005) The role of e-marketplaces in relationship-based supply chains: a survey. IBM Systems Journal, Vol. 44, Issue:1.

95. GRONMO, R. & JAEGER, M. C. (2005) Model-Driven Methodology for Building QoS-Optimised Web Service Compositions. In Proceedings Of Distributed Applications and Interoperable Systems, 5th IFIP WG 6.1 International Conference, DAIS, KUTVONEN, L. & ALONISTIOTI, N. (Eds.),Lecture Notes in Computer Science, Vol. 3543, Springer, pp. 68-82, Athens, Greece, June 15-17.

96. GRUBER, T. R. (1993a) Toward principles for the design of ontologies used for knowledge sharing In Proceedings Of Padua workshop on Formal Ontologies, GUARINO, N. (Ed.).

97. GRUBER, T. R. (1993b) A translation approach to portable ontology specifications. Knowledge Acquisition, Vol. 5, Issue:2, pp. 199-200, June 1993, ISSN: 1042-8143.

98. GRUENINGER, M. & FOX, M. S. (1995) Methodology for the Design and Evaluation of Ontologies. Technical Report, Toronto, Canada, University of Toronto.

99. HADAYA, P. (2006) Determinants of the future level of use of electronic marketplaces: The case of Canadian firms. Electronic Commerce Research, Vol. 6, Issue:2, pp. 173-185, April, 2006, ISSN: 1389-5753.

100. HAFFNER, R., SCHWABER, C., BROWNE, J., SHEEDY, T., LEGANZA, G. & STONE, J. (2007) Planned SOA Usage Grows Faster than actual SOA usage. IN

REPORT, F. (Ed.) Business Data Services North America, Europe, And Asia Pacific, February 28, 2007.

101. HALLER, A., CIMPIAN, E., MOCAN, A., OREN, E. & BUSSLER, C. (2005) WSMX - A semantic service oriented architecture, In Proceedings Of ICWS, IEEE, USA.
102. HAMMER, M. & CHAMPY, J. (2001) Reenginering the Corporation: A Manifesto for Business Revolution, pp. 257, Harper Business, ISBN: 978-0066621128.
103. HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2008) Semantic Web Services Applications - A Reality Check. Special Issue: Service-Oriented Architectures and Web Services. Wirtschaftsinformatik (WI) Vol. 1, pp. 39-46.
104. HAO, Y. & ZHANG, Y. (2007) Web services discovery based on schema matching. In Proceedings Of 13th Australasian Conference on Computer Science, Vol. 244, ACM, pp. 107-113, Australia.
105. HAREL, D. & NAAMAD, A. (1996) The STATEMATE Semantics of Statecharts ACM Transactions - Software Engineering and Methodology, Vol. 5, Issue:4, pp. 293-333.
106. HARS, A. (1994) Referenzdatenmodelle - Grundlagen effizienter Datenmodellierung, Wiesbaden Gabler.
107. HE, H. (2003) What is service oriented architecture?, Online http://webservices.xml.com/.
108. HEINIS, T., PAUTASSO, C. & ALONSO, G. (2005) Design and Evaluation of an Autonomic Workflow Engine. In Proceedings Of Second International Conference onAutonomic Computing, 2005. ICAC 2005, IEEE Press, pp. 27-38, Seattle, Washington, USA, June.
109. HEPP, M., LEYMANN, F., DOMINGUE, J., WAHLER, A. & FENSEL, D. (2005) Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. . In Proceedings Of ICEBE, IEEE, pp. 535-540, Beijing, China, October 18-20.
110. HESS, A. (2006) Supervised and Unsupervised ensemble learning for the Semantic Web. School of Computer Science and Informatics, Dublin, Ireland, National University of Ireland pp. 148.
111. HEVNER, A. R., MARCH, S. T., PARK, J. & RAM, S. (2004) Design Science in Information Systems Research. MIS Quarterly, Vol. 28, Issue:1, pp. 75-105.
112. HIDALGA, A. N. D. L., ZHAO, L. & FALCONE-SAMPAIO, P. R. (2006) Leveraging e-marketplaces models for Web service-based application development. IN PAGES-CASAS, L. (Ed.) Web services.
113. HILES, A. (2002) E-Business Service Level Agreements: Strategies for Service Providers, E-Commerce and Outsourcing, pp. 177, Rothstein Associates Inc., ISBN: 0-9641648-9-2.
114. HOFFMANN, J., SCICLUNA, J., KACZMAREK, T. & WEBER, I. (2007) Polynomial-Time Reasoning for Semantic Web Service Composition. In Proceedings Of 2007 IEEE Congress on Services, IEEE, pp. 229-236, Salt Lake City, UT, July 9-13, ISBN: 978-0-7695-2926-4.
115. HOLLINGSWORTH, D. (1995) The Workflow Reference Model. Lighthouse point, Florida, USA, Workflow Management Coalition.
116. HUHNS, M. N. & SINGH, M. P. (2005) Service-oriented computing: Key concepts and principles. IEEE Internet Computing, pp. 75-81, January and February.
117. HUNHS, M. N. & SINGH, M. P. (2005) Service-oriented computing: Key concepts and principles. IEEE Internet Computing, pp. 75-81, January and February.
118. HWANG, C. L. (1981) Multiple Attribute Decision Making: Methods and Applications Lecture Notes in Economics and Mathematical Systems, pp. 259, Springer, ISBN: 978-0387105581.
119. IEEE (1990) IEEE Std. 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology, Piscataway.

120. JAEGER, M. C., MUHL, G. & GOLZE, S. (2005) QoS-Aware Composition of Web Services: A Look at Selection Algorithms. In Proceedings Of IEEE International Conference on Web Services, IEEE, pp. 807-808, ISBN:0-7695-2409-5.
121. JAEGER, M. C. & ROJEC-GOLDMANN, G. (2006) SENECA – Simulation of Algorithms for the Selection of Web Services for Compositions. Technologies for E-Services. Springer Berlin / Heidelberg, ISBN: 978-3-540-31067-9.
122. JAEGER, M. C., ROJEC-GOLDMANN, G. & MUEHL, G. (2004) QoS Aggregation for Service Composition using Workflow Patterns. In Proceedings Of 8th International Enterprise Distributed Object Computing Conference (EDOC'04), IEEE Press, pp. 149-159, Monterey, California, September.
123. JAEGER, M. C. & TANG, S. (2004) Ranked Matching for Service Descriptions using DAML-S. In Proceedings Of CAiSE'04 Workshops, KIRIKOVA, J. G. A. M. (Ed.), Vol., Riga Technical University, pp. 217-228, Riga, Latvia, June.
124. KACZMAREK, M., KACZMAREK, T., VULCU, G., WEBER, I., WETZSTEIN, B., MA, Z., ZYSKOWSKI, D. & AL., E. (2008) D3.5 SUPER Deliverable on Business Process Composition. SUPER Consortium.
125. KACZMAREK, M., ZYSKOWSKI, D., HANIEWICZ, K. & ABRAMOWICZ, W. (2009) SOA, Web services and QoS - perception on the Polish market. Technical Report, Department of Information Systems, Poznan University of Economics.
126. KAO, H., JIN, H., WU, S. & QI, L. (2006) ServiceFlow: QoS Based Service Composition in CGSP. Enterprise Distributed Object Computing Conference, IEEE, pp. 453 – 458.
127. KAPLAN, R. S. & NORTON, D. P. (1996) The Balanced Scorecard: Translating Strategy into Action, pp. 322, Harvard Business School Press, ISBN: 978-0875846514.
128. KEENEY, R. L. & RAIFFA, H. (1976) Decisions with Multiple Objectives: Preferences and Value Tradeoffs, New York, J. Wiley.
129. KHAN, S. (1998) Quality Adaptation in a Multisession Mulitmedia System: Model, Algorithms and Architecture. Department of Electrical and Computer Engineering, Victoria, University of Victoria, May.
130. KHAN, S., LI, K. F., MANNING, E. G. & AKBAR, M. (2002) Solving the knapsack problem for adaptive multimedia. Studia Informatica 2 (special issue on Cutting, Packing and Knapsacking Problems), Vol. 1, pp. 161-182.
131. KIM, H. M., SENGUPTA, A. & EVERMANN, J. (2005) MOQ: Web services ontologies for QoS and general quality evaluations. In Proceedings Of European Conference on Information Systems (ECIS05), May.
132. KING, S. (2002) The e-Marketplace Evolution: Moving Beyond Transactions. www.ebizq.net, 07/14/2002.
133. KOWALKIEWICZ, M., LUDWIG, A., KACZMAREK, M. & ZYSKOWSKI, D. (2005) Documented mechanisms for Dynamic Service Prpofiling and Agreement Life-cycle Management. Adaptive Services Grid Project Deliverables, Potsdam, ASG internal deliverable, FP6-IST-004617.
134. KRAFZIG, D., BANKE, K. & SLAMA, D. (2005) Enterprise SOA: service-oriented architecture best practices., Prentice Hall.
135. KREGER, H. (2003) Fulfilling the Web services promise. Communications of the ACM, Vol. 46, Issue:6, pp. 29-ff, June, ISSN: 0001-0782.
136. KRETZSCHMAR, F. (2007) Negotiations in Service-Oriented Architectures. In Proceedings Of eOrganisation: Service-, Prozess-, Market-Engineering: 8th Internationale Tagung Wirtschaftsinformatik, ANDREAS OBERWEIS, C. W., HENNER GIMPEL, AGNES KOSCHMIDER , VICTOR PANKRATIUS AND BJOERN SCHNIZLER (Ed.), Vol. 1, Universitaetsverlag Karlsruhe, pp. 461-476, Karlsruhe, Germany, February 28-March 2, ISBN: 978-3-86644-094-4.
137. KRILL, P. (2005) Oracle upgrade app server with SOA in mind. Computer world Australia, http://www.computerworld.com.au.

138. KUESTER, U. & KOENIG-RIES, B. (2007) Supporting Dynamics in Service Descriptions - The Key to Automatic Service Usage. In Proceedings Of Fifth International Conference Service-Oriented Computing (ICSOC 2007),Lecture Notes in Computer Science, Vol. 4749, Springer, pp. 220-232, Vienna, Austria, September 17-20, ISBN: 978-3-540-74973-8.

139. KUROPKA, D., TROEGER, P., STAAB, S. & WESKE, M. (Eds.) (2008) Semantic Service Provisioning, Springer Berlin Heidelberg, ISBN: 978-3-540-78616-0.

140. KUROPKA, D. & WESKE, M. (2006) Die Adaptive Services Grid Platform: Motivation, Potential, Funktionsweise und Anwendungsszenarien. EMISA Forum, January 2006.

141. KUSTER, U., KOENIG-RIES, B., STERN, M. & KLEIN, M. (2007) DIANE- An integrated approach to automated service discovery, matchmaking and composition. In Proceedings Of WWW 2007, pp. 1033-1042, Banff, Alberta, Canada, May 8-12, 2007.

142. LAMPARTER, S. & AGARWAL, S. (2005) Specification of Policies for Automatic Negotiations of Web Services. In Proceedings Of Semantic Web and Policy Workshop, held in conjunction with the 4th International Semantic Web Conference, HENDLER, L. K. A. T. F. A. J. (Ed.), pp. 99-109, Galway, Ireland, November 2005.

143. LAMPARTER, S. & ANKOLEKAR, A. (2007) Automated selection of configurable Web services. In Proceedings Of 8th Internationale Tagung Wirtschaftsinformatik, Universitätsverlag Karlsruhe, Karlsruhe, Germany, March.

144. LAMPARTER, S., ANKOLEKAR, A., STUDER, R. & GRIMM, S. (2007) Preference-based selection of highly configurable Web services. In Proceedings Of WWW 2007, ACM, pp. 1013-1022, Banff, Alberta, Canada, May 8-12, 2007.

145. LAMPARTER, S. & SCHNIZLER, B. (2006) Trading services in ontology-driven markets. In Proceedings Of ACM Symposium on Applied computing, ACM Press, pp. 1679-1683.

146. LARA, R., CORELLA, M. A. & CASTELLS, P. (2006) A flexible model for Web service discovery. In Proceedings Of 1st International Workshop on Semantic Matchmaking and Resource Retrieval: Issues and Perspectives, Seul, South Korea.

147. LAUSEN, H. & HASELWANTER, T. (2007) Finding Web services. Technical Report, April 18, 2007, available at: http://seekda.com/download/Finding_Web_Services.pdf.

148. LEYMANN, F. & ALTENHUBER, W. (1994) Managing business processes as information resources. IBM Systems Journal, Vol. 32, Issue:2, pp. 326-348.

149. LI, L. & HORROCKS, I. (2003) A Software Framework for Matching based on Semantic Web Technology. In Proceedings Of 12th International Confernece on the World Wide Web, Budapest, Hungary, May.

150. LI, Z., ZHAO, H. & RAMANATHAN, S. (2003) Pricing web services for optimizing resource allocation - an implementation scheme. In Proceedings Of Web 2003, Seattle.

151. LICAN, H., WALKER, D. W., HUANG, Y. & RANA, O. F. (2005) Dynamic Web Service Selection for Workflow Optimization. In Proceedings Of UK e-Science All Hands Meeting (AHM2005), September.

152. LIN, W.-L., LO, C.-C., CHAO, K.-M. & YOUNAS, M. (2008) Consumer-centric QoS-aware selection of Web services. Journal of Computer and Systems Sciences, Vol. 74, Issue:2, pp. 211-231, March.

153. LINDEMANN, M. A. & SCHMIDT, B. F. (1998) Elements of a reference model for electronic markets. In Proceedings Of Hawaii International Conference on System Sciences, Vol. 4, In: HICSS '98: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences, Hawaii.

154. LIU, C., PENG, Y. & CHEN, J. (2006) Web services description ontology-based service discovery model. In Proceedings Of International Conference on Web Intelligence (WI 2006), IEEE/WIC/ACM, pp. 633-636.

155. LIU, Y., NGU, A. H. H. & ZENG, L. (2004) QoS Computation and Policing in Dynamic Web Service Selection. In Proceedings Of 13th International Conference on World Wide Web (WWW), ACM Press, New York, USA, May.

156. LORD, P. W., ALPER, P., WROE, C. & GOBLE, C. A. (2005) Feta: A Light-weight architecture for user oriented semantic service discovery. In Proceedings Of Second European Semantic Web Conference(ESWC'05), A. GOMEZ-PEREZ, J. E. (Ed.), Vol. 3532, Springer-Verlag, pp. 17-31, Crete, Greece.

157. LUDWIG, H., KELLER, A., DAN, A., KING, R. & FRANCK, R. (2003) Web services level agreement (WSLA) Langauge specifcation.

158. LUDWIG, S. & REYHANI, S. (2007) Selection Algorithm for Grid Services based on a Quality of Service Metric. In Proceedings Of 21st International Symposium on High Performance Computing Systems and Applications (HPCS'07), p. 7.

159. LUO, Z. & LI, J. (2005) A Web Services Community Model For Service Fulfillment In Proceedings Of IEEE International Conference on Service Computing, Vol., IEEE, Orlando, FL, July.

160. MAAMAR, Z., SHENG, Q. Z. & BENATALLAH, B. (2003) Selection of Web services for Composition Using Location of Provider Hosts Criterion. In Proceedings Of Conference Proceeding of the 15th Conference on Advanced Information Systems Engineering (CAiSE'03) Workshops, PERNICI, J. E. A. R. M. A. B. (Ed.),CEUR Workshop Proceedings, Vol. 75, CEUR-WS.org, Klagenfurt, Austria, 16-20 June, ISBN: 86-435-0552-8.

161. MACIASZEK, L. (2007) Requirements analysis and systems design, pp. 642, Harlow England, Addison Wesley, ISBN: 978-0-321-44036-5.

162. MAEDCHE, A. & STAAB, S. (2001) Ontology Learning for the Semantic Web. IEEE Intelligent Systems, Vol. 16, Issue:2, pp. 72-79, March, ISSN: 1541-1672.

163. MALLOY, B. A., KRAFT, N. A., HALLSTROM, J. O. & VOAS, J. M. (2006) Improving the predictable assembly of service-oriented architectures. Software, IEEE, Vol. 23, Issue:2, pp. 12-15, 2006.

164. MANIKRAO, U. S. & PRABHAKAR, T. V. (2005) Dynamic selection of Web services with Recommendation System. In Proceedings Of International Conference on Next Generation Web Services Practices, IEEE Computer Society, pp. 117, Seul, Korea, ISBN: 0-7695-2452-4.

165. MARTELLO, S. & TOTH, P. (1987) Algorithms for Knapsack Problems. IN S. MARTELLO, G. L., M. MINOUX, C. RIBEIRO (Ed.) Surveys in Combinatorial Optimization. Annals of Discrete Mathematics.

166. MAXIMILIEN, E. & SINGH, M. (2004a) A Framework and Ontology for Dynamic Web Services Selection. IEEE Internet Computing, pp. 84–93, September-October.

167. MAXIMILIEN, E. M. (2005) Multi-agent System for Dynamic Web Services Selection. In Proceedings Of 1st Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE at AAMAS), Utrecht, The Netherlands, July 25-29.

168. MAXIMILIEN, E. M. & MUNINDAR, S. P. (2004) A Framework and Ontology for Dynamic Web Services Selection, Internet Computing, IEEE, vol 8(5), pp. 84- 93, ISSN: 1089-7801.

169. MAXIMILIEN, E. M. & SINGH, M. P. (2002 b) Conceptual Model of Web services Reputation. SIGMOD Record, October 2002.

170. MAXIMILIEN, E. M. & SINGH, M. P. (2004b) Towards Autonomic Web services trust and selection. In Proceedings Of ICSOC'04, ACM, New York, USA, November, 2004.

171. MCILRAITH, S. & SON, T. C. (2002) Adapting Golog for composition of Semantic Web services. In Proceedings Of 8th International Conference on Knowledge Representation and Reasoning (KR2002), Toulouse, France, April.

172. MCILRAITH, S., SON, T. C. & ZENG, H. (2001) Semantic Web services. IEEE Intelligent systems. Special Issue on the Semantic Web, Vol. 16, Issue:2, pp. 46-53, March/April 2001.

173. MEDJAHED, B., BENTALLAH, B., BOUGUETTAYA, A., NGU, A. H. & EKMAGERMID, A. K. (2003a) Business-to-business interactions: issues and enabling technologies. VLDB Journal, Vol. 12, pp. 59-85.

174. MEDJAHED, B., REZGUI, A., BOUGUETTAYA, A. & OUZZANI, M. (2003b) Infrastructure for e-government Web services. IEEE Internet Computing, Vol. 7(1), pp. 58-65.

175. MENASCE, D. A. (2002) QoS issues in Web services. IEEE Internet Computing, Vol. 6 (6), pp. 72-75, November/December, 2003.

176. MENASCE, D. A., CASALICCHIO, E. & DUBEY, V. (2008) A heuristic approach to optimal service selection in service oriented architectures. Proceedings of the 7th international workshop on Software and performance, Princeton, NJ, USA, ACM.

177. MENASCE, D. A. & DUBEY, V. (2007) Utility-based QoS Brokering in Service Oriented Architectures. In Proceedings Of IEEE International Conference on Web Services (ICWS'07), pp. 422-430, July 9-23.

178. MEZZETI, N. (2003) Towards a model for trust relationships in virtual enterprises. In Proceedings Of 14th International Workshop on Database and Expert Systems Applications, ISBN: 0-7695-1993-8.

179. MIMOSO, M. (2004) Business needs driving diverse Web services specs. SearchWebServices.com, 12.12.2007.

180. MUELLER, I., BRAUN, P. & ROSSAK, W. (2002) Integrating Mobile Agent Technology into an e-marketplace solution: The InterMarket Marketplace. Friedrich-Schiller-University Jena.

181. NANDIGAM, J., GUDIVADA, V. N. & KALAVALA, M. (2005) Semantic Web services. Journal of Computing Sciences in Colleges, Vol. 21, Issue:2, pp. 50-63.

182. NOLL, J. (2004) ASG based scenarios in Telecommunications, Telematics and enhanced Enterprise IT". IN ASG (Ed.) ASG Research Project Deliverables.

183. O'BRIEN, L., MERSON, P. & BASS, L. (2007) Quality Attributes for Service-Oriented Architecture. In Proceedings Of International Workshop on Systems Development in SOA Environment, IEEE Computer Society, ISBN: 0-7695-2960-7.

184. O'SULLIVAN, J., EDMOND, D. & HOFSTEDE, A. H. T. (2005) Service description. service-description.com.

185. O'SULLIVAN, J., EDMOND, D. & HOFSTEDE, A. T. (2002) What's in a service? Towards an accurate description of non-functional properties of Web services. Distributed and Parallel Databases, Issue:12, pp. 117-133, September 2002.

186. OASIS (2006a) Reference model for service oriented architecture1.0. IN OASIS (Ed.) OASIS Standard.

187. OASIS (2006b) Web services Business Process Execution Language Version 2.0. OASIS.

188. OASIS (2006c) WS Security Core Specification 1.1. OASIS Standard Specification, February 1, 2006.

189. OASIS (2008) Reference Architecture for Service Oriented Architecture, Version 1.0. IN JEFF A. ESTEFAN, K. L., FRANCIS G. MCCABE, DANNY THORNTON (Ed.) Public Review Draft, 23 April.

190. OMG (2004) UML Profile for Modelling Quality of Service and Fault Tolerance Characteristics and Mechanisms. Object Management Group, June.

191. ORCHARD, D. (2002) Web Services Pitfalls. WebServices.XML, 12.12.2007.

192. PADOVITZ, A., KRISHNASWAMY, S. & LOKE, S. (2003) Towards Efficient Selection of Web Services. In Proceedings Of 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 9 pages, July.

193. PAN, J. (2005) Towards improving web service repositories through semantic web techniques. In Proceedings Of Workshop on semantic web enabled software engineering (SWESE) at ISWC'2005, 7 Nov 2005.

194. PAOLUCCI, M., KAWAMURA, T., PAYNE, T. & SYCARA, K. (2002) Semantic matching of Web services capabilities In Proceedings Of International Semantic Web Conference (ISWC), Springer Verlag, pp. 333-347, Sardinia, Italy, June.

195. PAOLUCCI, M., SYCARA, K. & KAWAMURA, T. (2003a) Delivering Semantic Web Services. In Proceedings Of 12th World Wide Web Conference, pp. 111-118, Budapest, Hungary, May 2003.

196. PAOLUCCI, M., SYCARA, K., NISIMURA, T. & SRINIVASAN, N. (2003b) Toward Semantic Web Services. In Proceedings Of Workshop on E-services and the Semantic Web collocated with WWW Conference, Budapest, May, 2003.

197. PAPAZOGLOU, M. P. (2003) Service-Oriented Computing: Concepts, Characteristics and Directions. In Proceedings Of Fourth International Conference on Web Information Systems Engineering (WISE'03), IEEE Computer Society Press, pp. 3-12, Roma, Italy, December.

198. PAPAZOGLOU, M. P. & GEORGAKOPOULOS, A. D. (2003) Service Oriented Computing - Introduction. Communication of the ACM, Vol. 46, Issue:10, pp. 24-28, 2003, ISSN: 0001-0782.

199. PAPAZOGLOU, M. P. & HEUVEL, W. J. (2007) Service oriented architectures: approaches,technologies and research issues. The VLDB Journal, Vol. 16, Issue:3, pp. 389-415, July 2007, ISSN: 1066-8888.

200. PAPAZOGLOU, M. P., TRAVERSO, P., DUSTDAR, S., LEYMANN, F. & KRAEMER, B. J. (2006) Service-Oriented Computing: A research roadmap. IN PAPAZOGLOU, M.P (Ed.) Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, ISBN: 1862-4405.

201. PASCHKE, A. (2006) Rule-based Knowledge Representation for Service Level Agreement. In Proceedings Of Doctoral Symposium of the 4th Conference on Multiagent System Technologies (MATES'06), Germany.

202. PATIL, A. A., OUNDHAKAR, S. A., SHETH, A. P. & VERMA, K. (2004) METEOR-S Web service annotation framework. In Proceedings Of 13th International Conference on World Wide Web, ACM Press, pp. 553-562, New York, USA.

203. PETRIE, C. & BUSSLER, C. (2008) The Myth of Open Web Services. The Rise of the Service Parks. IEEE Internet Computing, Vol. 12, Issue:3, pp. 94-96, May/June.

204. PEZZINI, M. (2005) SOA Beyong Hype and Disillusionment - A Strategic Perspective., Keynote speech given at the SOA Days 2005 Technology Conference, September.

205. POLADIAN, V., GARLAN, D., SHAW, M. & SOUSA, J. P. (2004) Dynamic configuration of resource-aware services. In Proceedings Of 26th International Conference on Software Engineering (ICSE 2004), IEEE Computer Society, pp. 604-613, Edinburgh, Scotland, May.

206. PONNEKANTI, S. R. & FOX, A. (2002) SWORD: A developer toolkit for Web services composition. In Proceedings Of 11th World Wide Web Conference, Honolulu, USA.

207. PREIST, C. (2004) A conceptual architecture for Semantic Web Services. In Proceedings Of International Semantic Web Conference 2004 (ISWC 2004), November 2004.

208. RAISCH, W. D. (2001) The eMarketplace: Strategies for success in B2B ecommerce, New York: McGraw Hill.

209. RAN, S. (2003) A model for Web services discovery with QoS. ACM SIGecom Exchanges, Vol. 4(1), pp. 1-10, 2003.

210. RAO, J., KUENGAS, P. & MATSKIN, M. (2004) Logic-based Web service composition: from service description to process model. In Proceedings Of 2004 International Conference on Web services, IEEE, San Diego, USA, July.

211. RAO, J. & SU, X. (2005) A survey of automated Web service composition methods. Semantic Web Services and Web Process Composition. Springer Berlin/Heidelberg, ISBN: 978-3-540-24328-1.

212. RECKER, J. & MENDLING, J. (2006) On the translation between BPMN and BPEL: Conceptual Mismatch between Process Modeling Languages. In Proceedings Of Workshop at the 18th International Conference on Advanced Information Systems Engineering, , LATOUR, T. A. P., MICHAEL (Ed.), pp. 521-532.

213. ROMAN, D., BRUIJN, J. D., MOCAN, A., LAUSEN, H., BUSSLER, C. & FENSEL, D. (2006) WWW:WSMO, WSML and WSMX in a nutshell. In Proceedings Of First Asian Semantic Web Conference (ASWC 2006), Beijing, China, September 3-7.

214. ROSA, N. S., CUNHA, P. R., FREIRE, L. & JUSTO, G. R. (2002) Process NFL: A language for describing non-functional properties. In Proceedings Of 35th Annual Hawaii International Conference (HICSS), pp. 3676-3685, Hawaii, USA, March 29, 2005.

215. SCHMELZER, H. J. & SESSELMANN, W. (2006) Geschäftsprozessmanagement in der Praxis, pp. 580, München, Carl Hanser Verlag, ISBN: 3-446-40589-5.

216. SCHROEPFER, C., BINSHTOK, M., SHIMONY, S. E., DAYAN, A., BRAFMAN, R., OFFERMANN, P. & HOLSCHKE, O. (2007) Introducing preferences over NFPs into service selectionin SOA. In Proceedings Of Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop, Vienna, Austria, September 17.

217. SCHUSTER, H., GEORGAKOPOULOS, D., CICHOCKI, A. & BAKER, D. (2000) Modelling and composing service-based and reference process-based multi-enteprise processes. In Proceedings Of 12th International Conference on Advanced Information Systems Engineering (CAiSE), Springer Verlag, Stockholm, Sweden, June.

218. SHAIKH, A., LUDWIG, S. A. & RANA, O. F. (2005) A Cognitive Trust-Based Approach for Web Service Discovery and Selection. In Proceedings Of Third European Conference on Web services, IEEE Computer Society, pp. 38, ISBN: 0-7695-2484-2.

219. SHEN, D., YU, G., NIE, T., LI, R. & YANG, X. (2004) Modeling QoS for Semantic Equivalent Web services. In Proceedings Of Fifth International Conference on Web-Age Information Management (WAIM 2004), Springer Berlin, pp. 478-488, Dalian, China, September 27, 2004, ISBN: 978-3-540-22418-1.

220. SHERCHAN, W., LOKE, S. W. & KRISHNASWAMY, S. (2006) Towards Explanation-Aware Selection in Internet-Scale Infrastructures: Generating Rationale for Web Services Ratings and Reputation. In Proceedings Of 10th IEEE International Conference Workshops on Enterprise Distributed Object Computing, IEEE, pp. 8, ISBN: 0-7695-2743-4.

221. SHETH, A., CORDOSO, J., MILLER, J. & KOCHUT, K. (2002) QoS for service-oriented middleware. In Proceedings Of 6rh World Multiconference on Systemics Cybernetics and Informatics (SCI02), pp. 528-534, July, 2002.

222. SHMUELI, O. (2001) Architecture for internal Web services deployment. In Proceedings Of 27th VLDB, Roma, Italy.

223. SIRIN, E., HENDLER, J. A. & PARSIA, B. (2003) Semi-automatic composition of Web services using Semantic Descriptions. In Proceedings Of Web services: Modeling, Architecture and Infrastructure workshop in conjunction with ICEIS2003

224. SREENATH, R. M. & SINGH, M. P. (2003) Agent-based service selection. Journal on Web Semantics (JWC).

225. SRIHAREE, N. (2006) Semantic Web Services Discovery Using Ontology-based Rating Model. In Proceedings Of IEEE/WIC/ACM International Conference on Web Intelligence (WI'06), IEEE.

226. SRINIVASAN, N., PAOLUCCI, M. & SYCARA, K. (2004) An efficient algorihtm for OWL-S based semantic search in UDDI. In Proceedings Of SWSWPC, pp. 96-110.

227. SRIVASTAVA, B. & KOEHLER, J. (2003) Web service composition - current solutions and open problems. In Proceedings Of ICAPS 2003, Workshop on planning for Web services, Trento, Italy.

228. STANKIEWICZ, M. J. (2005) Konkurencyjność przedsiębiorstwa. Budowanie konkurencyjności przedsiębiorstwa w warunkach globalizacji, Toruń, TNOiK.

229. STOLLBERG, M. & LARA, R. (2004) WSMO Use case - Virtual Travel Agency. DERI.

230. TAHER, L., KHATIB, H. E. & BASHA, R. (2005) A framework and QoS matchmaking algorithm for dynamic Web services selection. In Proceedings Of 2nd International Conference on Innovations in Information Technology (IIT'05), September.

231. TIAN, M., GRAMM, A., NAUMOWICZ, T., RITTER, H. & SCHILLER, J. (2003) A Concept for QoS Integration in Web Services,. In Proceedings Of Fourth International Conference on Web Information Systems Engineering Workshop (WISEW'03).

232. TILSNER, D., ABRAMOWICZ, W., WISNIEWSKI, M., MOORE, P. & PEINEL, G. (2005) Usability-driven open platform fro Mobile Government. In Proceedings Of First European Conference on Mobile Government, Brighton, UK, ISBN: 9763341-0-0.

233. TOMA, I. (2006) Non-functional properties in Web services. WSML Working Draft, DERI, January 21, 2006.

234. TOSIC, V., MA, W., PAGUREK, B. & ESFANDIARI, B. (2004) Web Service Offerings Infrastructure (WSOI) - a management infrastructure for XML Web services. In Proceedings Of IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), Vol. 1, IEEE Press, pp. 817-830, Seoul, South Korea, April, ISBN: 0-7803-8230-7.

235. TOSIC, V., PAGUREK, B., ESFANDIARI, B., PATEL, K. & MA, W. (2002a) Web Service Offerings Language (WSOL) and Web Service Composition Management (WSCM). In Proceedings Of Object-Oriented Web Services Workshop at OOPSLA 2002, Vol., Seattle, USA.

236. TOSIC, V., PATEL, K. & PAGUREK, B. (2002b) WSOL- Web service offerings language. In Proceedings Of Workshop on Web services, e-Business and the Semantic Web (WES), pp. 57-67, Toronto, Canada.

237. TRAVERSO, P. & PISTORE, M. (2004) Automated Composition of Semantic Web Services into Executable Processes. The Semantic Web - International Semantic Web Conference 2004, pp. 380-394.

238. TSESMETZIS, D., ROUSSAKI, I. & SYKAS, E. (2007) Modeling and Simulation of QoS-aware Web service Selection for Provider Profit Maximization. Simulation, Vol. 83, Issue:1, pp. 93-106, ISSN: 0037-5497.

239. TSUR, S., ABITEBOUL, S., AGRAWAL, R., DAYAL, U., KLEIN, J. & WEIKUM, G. (2001) Are Web services the next revolution in e-commerce? In Proceedings Of VLDB Conference, pp. 614-617, Rome, Italy.

240. TUOKKOLA, K. (1998) The Future of Electronic Business. Department of Industrial Engineering and Management, Helsinki University of Technology.

241. UDDI (2004) UDDI Version 3.0.2. UDDI Spec Technical Committee.

242. USCHOLD, M. & GRUENINGER, M. (1996) Ontologies: Principles, methods and applications. Knowledge Engineering Review, Vol. 11, pp. 93-136.

243. VERMA, K., GOMADAM, K., SHETH, A. P., MILLER, J. A. & WU, Z. (2005a) The Meteor-s approach for configuring and executing dynamic web processes. LSDIS Lab Technical Report, University of Georgia, June 24.

244. VERMA, K., GOMADAM, K., SHETH, A. P., MILLER, J. A. & WU, Z. (2005b) The METEOR-S Approach for Configuring and executing Dynamic Web Processes. University of Georgia, Athens, June.

245. VERMA, K., SHETH, A., MILLER, J. & AGGARWAL, R. (2004) Dynamic QoS based Supply Chain, http://www.daml.org/.

246. VITVAR, T., MOCAN, A., KERRIGAN, M., ZAREMBA, M., ZAREMBA, M., MORAN, M., CIMPIAN, E., HASELWANTER, T. & FENSEL, D. (2007) Semantically-enabled service oriented architecture: concepts, technology and application Service Oriented Computing and Applications, Vol. 1, Issue:2, pp. 129-154, June, ISSN: 1863-2394.

247. VU, L.-H., HAUSWIRTH, M. & ABERER, K. (2005) QoS-Based Service Selection and Ranking with Trust and Reputation Management. In Proceedings Of On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, OTM Confederated International Conferences CoopIS, DOA, and ODBASE 2005, Agia Napa, Cyprus, October 31 - November 4, 2005, Proceedings, Part I,Lecture Notes in Computer Science, Springer, pp. 466-483, ISBN: 3-540-29736-7.

248. W3C (2004a) Owl-s: Semantic markup for web services, http://www.w3.org.

249. W3C (2004b) Web services architecture. W3C Working Draft, http://www.w3.org.

250. W3C (2006a) Web services Addressing 1.0 - Core. W3C Recommendation, May 9, 2006.

251. W3C (2006b) Web services Policy 1.5. W3C Working Draft, July 31, 2006.

252. W3C (2007) WSDL 2.0., http://www.w3.org.

253. WALKERDINE, J., HUTCHINSON, J., SAWYER, P., DOBSON, G. & ONDITI, V. (2007) A Faceted Approach to Service Specification. In Proceedings Of The 2nd International Conference on Internet and Web Applications and Services (ICIW), Mauritius, May, 2007.

254. WANG, Y. & VASSILEVA, J. (2007) Toward Trust and Reputation Based Web Service Selection: A Survey. Multi-agent and Grid Systems (MAGS) Journal, Special Issue on New Tendecies on Web Services and Multi-agent Systems.

255. WEBMETHODS (2002) Web services in the Telecommunications Industry - Driving new integration solutions. WebMethods White Papers, WebMethods.

256. WETZSTEIN, B., MA, Z., FILIPOWSKA, A., KACZMAREK, M., BHIRI, S., LOSADA, S., LOPEZ-COB, J.-M. & CICUREL, L. (2007) Semantic Business Process Management: A Lifecycle Based Requirements Analysis. In Proceedings Of Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007) HEPP, M., HINKELMANN, K., KARAGIANNIS, D., KLEIN, R. & STOJANOVIC, N. (Eds.),CEUR Workshop Proceedings, CEUR-WS.org, pp. 1-11, Wien, ISBN: 1613-0073.

257. WU, C. & CHANG, E. (2007) Intelligent Web Services Selection based on AHP and Wiki. In Proceedings Of IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society, pp. 767-770, ISBN: 0-7695-3026-5.

258. WU, D., SIRIN, E., HENDLER, J., NAU, D. & PARSIA, B. (2003) Autmatic Web services composition using SHOP2. In Proceedings Of Workshop on planning for Web services, Trento, Italy, June.

259. XIAO, J. & BOUTABA, R. (2005) QoS-aware service composition and adaptation in autonomic communication. IEEE Journal on Selected Areas in Communications, Vol. 23, Issue:12, pp. 2344-2360.

260. XU, Z., MARTIN, P., POWLEY, W. & ZULKERNINE, F. (2007) Reputation-Enhanced QoS-based Web Services Discovery. In Proceedings Of IEEE International Conference on Web Services (ICWS 2007), pp. 249-256, Salt Lake City, UT, ISBN: 0-7695-2924-0.

261. YAN, G., JUN, N., BIN, Z., LEI, Y., QIANG, G. & YU, D. (2006) Immune algorithm for selecting optimum services in Web services composition. Wuhan University Journal of Natural Sciences, Vol. 11, Issue:1, January.

262. YANG, J. (2003) Web Service Componentization. Communication of the ACM, Vol. 46(10), October.

263. YANG, L., DAI, Y., ZHANG, B. & GAO, Y. (2005) Dynamic selection of composite Web services based on a new structured TCNN. In Proceedings Of IEEE International Workshop on Service Oriented Systems Engineering, IEEE, pp. 141-148, 20-21 October.

264. YEOM, G. & MIN, D. (2005) Design and implementation of Web services QoS broker. In Proceedings Of Next Generation Web Services Practices, 2005. NWeSP 2005, 2 pp.

265. YU, H. Q. & MOLINA, H. (2007) A modified Logic Scoring Preference method for dynamic Web service evaluation and selection. In Proceedings Of 2nd European Young Researchers Workshop on Service Oriented Computing, Leicester, UK, 11-12 June.

266. YU, T., ZHANG, Y. & LIN, K.-J. (2007) Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints. ACM Transactions on the Web, Vol. 1, pp. 6, May, ISSN: 1159-1131.

267. ZEITHAML, V. A. (2003) Services marketing: integrating customer focus across the firm, McGraw-Hill Companies, ISBN: 0-07-119914-4.

268. ZENG, L., BENATALLAH, B., DUMAS, M., KALAGNANAM, J. & SHENG, Q. Z. (2003) Quality driven Web services composition. In Proceedings Of 12th International Conference on World Wide Web (WWW), ACM Press, Budapest, Hungary, May.

269. ZENG, L., BENATALLAH, B., NGU, A. H. H., DUMAS, M., KALAGNANAM, J. & CHANG, H. (2004) QoS-aware middleware for web services composition. IEEE Transactions on Software Engineering, Vol. 30(5), pp. 311-327, May.

270. ZHANG, H. & QIU, Y. (2005) Agent-based negotiation in e-commerce: state of the art. In Proceedings Of International Conference on Active Media Technology (AMT 2005), pp. 233-238, May 19-25, ISBN: 0-7803-9035-0.

271. ZHANG, L.-J. & LI, B. (2004) Requirements Driven Dynamic Services Composition for Web Services and Grid Solutions. Journal of Grid Computing, Vol. 2, Issue:2, pp. 121-140, June, ISSN: 1570-7873.

272. ZHIXIAN, Y., CIMPIAN, E., ZAREMBA, M., MAZZARA, M. (2007) BPMO: Semantic Business Process Modelling and WSMO Extension. In the proceedings of IEEE International Conference on Web services (ICWS), IEEE, pp. 1185-1186, Salt Lake City, July, ISBN: 0-7695-2924-0.

273. ZHOU, C., CHIA, L. & LEE, B. (2004) DAML-QoS ontology for web services. In Proceedings Of International Conference on Web services 2004 (ICWS04), July.

274. ZUR MUEHLEN, M. & RECKER, J. (2008) How Much Language is Enough? Theoretical and Practical Use of the Business Process Modelling Notation. In Proceedings Of 20th International Conference on Advanced Information Systems Engineering, http://eprints.qut.edu.au/archive/00012916/, Montpellier, France.

275. ZYSKOWSKI, D. (2010) Data model for SWS e-marketplace. Department of Information Systems, Poznań, Poznan University of Economics.

8 Listings

8.1 Sequence diagrams – enlarged versions

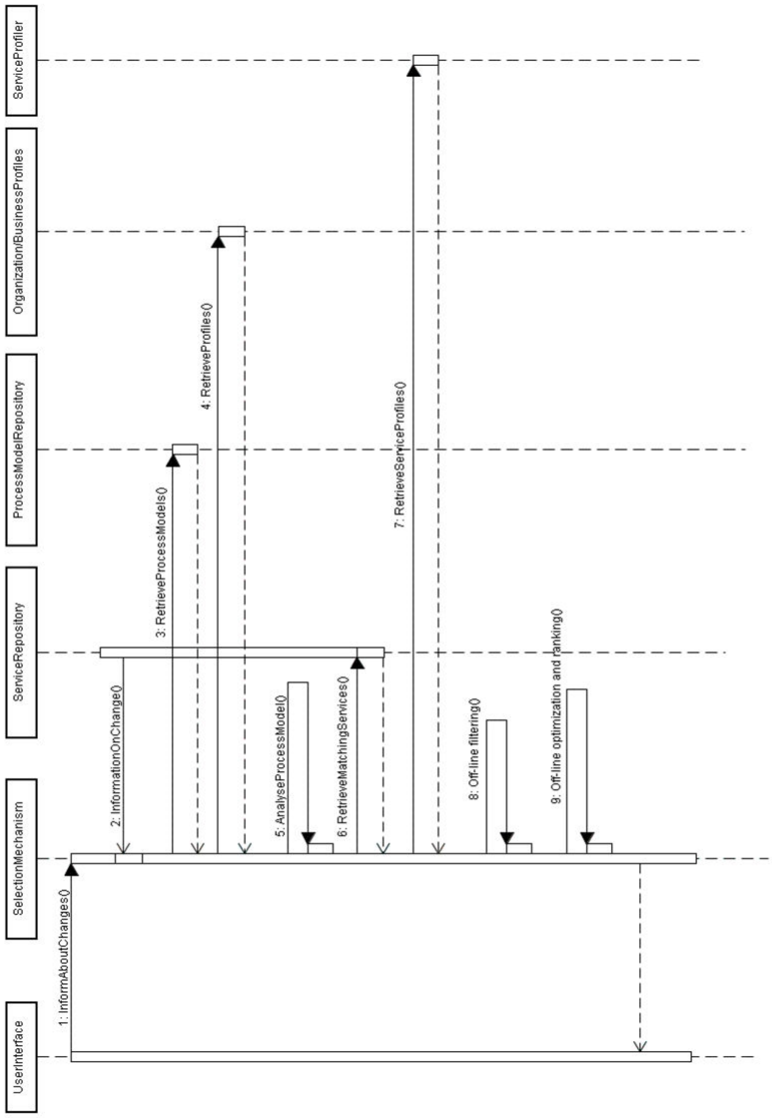


Figure 41 Sequence diagram – re-optimization (enlarged version)

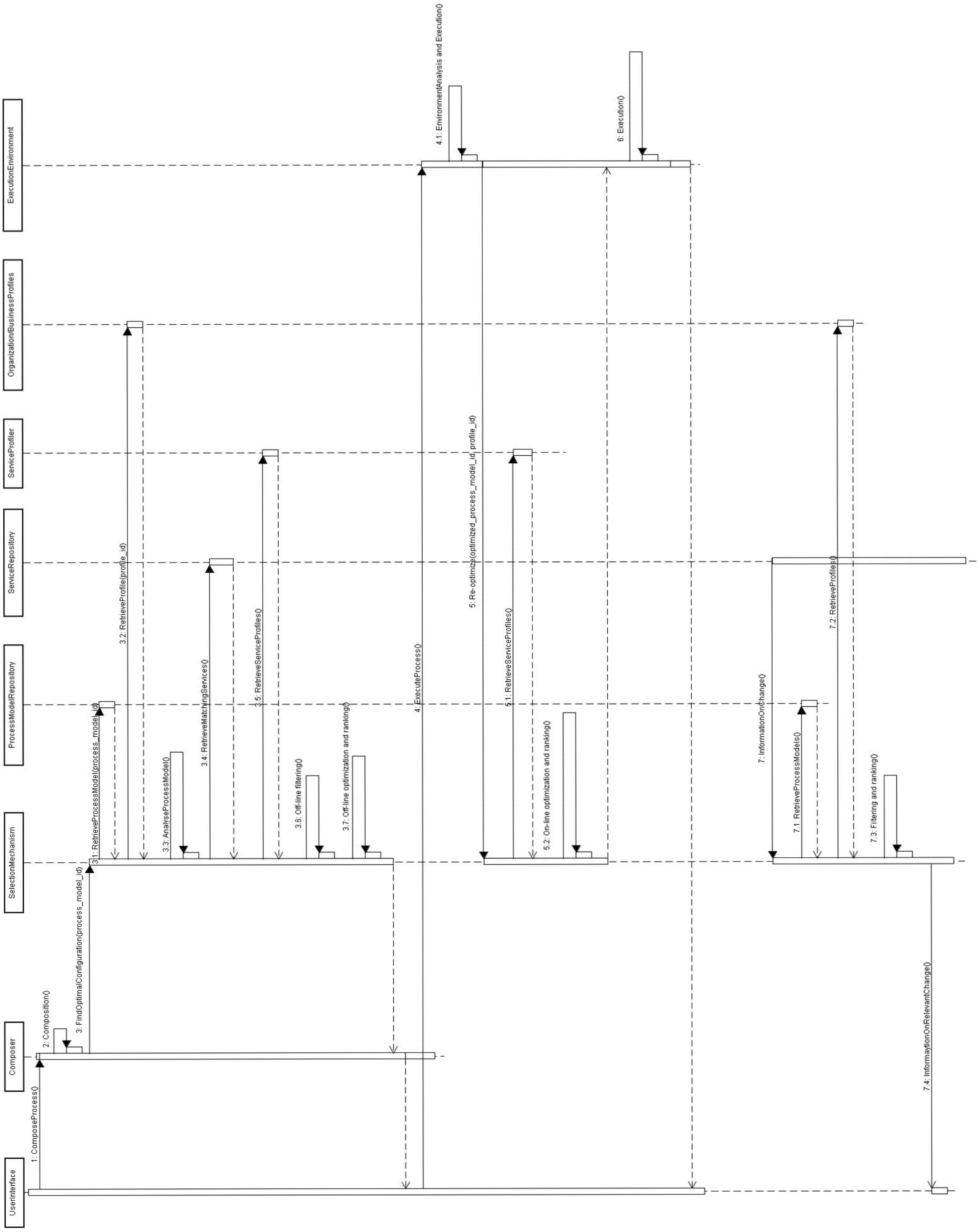


Figure 42 Sequence diagram – two step selection (enlarged version)

8.2 Information model ontology - excerpt

```
   wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"

namespace { _"http://www.kie.ae.poznan.pl/ontologies#",
     wsmostudio _"http://www.wsmostudio.org#",
     dc _"http://purl.org/dc/Elements/1.1#",
     priceOnto _"http://www.wsmo.org/ontologies/nfp/priceNFPOntology#",
     temporalOnto
_"http://www.wsmo.org/ontologies/nfp/temporalNFPOntology#",
     availabilityOnto
_"http://www.wsmo.org/ontologies/nfp/availabilityNFPOntology#",
     providerOnto
_"http://www.wsmo.org/ontologies/nfp/providerNFPOntology#",
     securityOnto
_"http://www.wsmo.org/ontologies/nfp/securityNFPOntology#",
     cur _"http://www.wsmo.org/ontologies/nfp/currencyNFPOntology#" }

ontology _http://www.kie.ue.poznan.pl/ontologies/#InformationModel
     nonFunctionalProperties
          wsmostudio#version hasValue "0.7.3.7"
          dc#subject hasValue "The infomration model developed for the
needs of the B2B SWS e-marketplace."
          dc#language hasValue "English"
          dc#description hasValue "The ontology presents the information
model developed for the needs of the business oriented selection on the
B2B SWS e-marketplace"
          dc#creator hasValue "Monika Kaczmarek"
          dc#title hasValue "Properties Ontology for the needs of the
business oriented selection mechanism"
     endNonFunctionalProperties

//set of axioms..

axiom reliabilityValueRestriction
     definedBy
          ?x memberOf Reliability
:-
?x[value hasValue ?v]
  and ?v >= 0
  and ?v =< 100.

axiom allProcessAndTaskAndServiceLevelsSupported
     definedBy
?x memberOf {Cost, Availability, Owner, DataConfidentiality,
DataEncription, Authentication, Authorization, NonRepudiation} and ?x[
relatesToLevel hasValue {processSubLevel,taskSubLevel, ServiceLevel} ].

//set of relations

relation relatesToProcessStructure ( ofType Property, ofType Structure )
     nonFunctionalProperties
           dc#description hasValue "A relation that defines which  can
influence another goal in a positive or negative way."
     endNonFunctionalProperties

relation higherPriority ( ofType Goal, ofType Goal )
     nonFunctionalProperties
           dc#description hasValue "Goal 1 is of higher priority than goal
2."
     endNonFunctionalProperties

relation conflicts ( ofType Goal, ofType Goal )
     nonFunctionaProperties
           dc#description hasValue "A goal is in conflict with another if
it hinders the other and has lower priority."
     endNonFunctionalProperties

// set of concepts

concept Level subConceptOf SupportingArtefacts
     nonFunctionalProperties
```

```
          dc#description hasValue "The level concept specifies the
perception layer: Process level, Task level, Service level"
     endNonFunctionalProperties
     hasName ofType _string

concept ProcessLevel subConceptOf Level
     nonFunctionalProperties
          dc#description hasValue "The process level concept"
     endNonFunctionalProperties

//properties branch

concept Property
     nonFunctionalProperties
          dc#description hasValue "The generic concept from which all
other will inherit"
     endNonFunctionalProperties
     hasName ofType _string
     relatesToLevel ofType Level

concept BusinessOrientedProperty subConceptOf Property
     hasCalculation ofType _string
     hasDefinition ofType _string
     hasMetric ofType Unit
     relatesTo ofType NFP
     nonFunctionalProperties
          dc#description hasValue "A group of properties targeted at
business users."
     endNonFunctionalProperties

concept KPI subConceptOf BusinessOrientedProperty
nonFunctionalProperties
          dc#description hasValue "A group of properties - key performance
indicators."
     endNonFunctionalProperties

concept ProcessDuration subConceptOf { KPI, CriticalPathAttribute}
     value ofType _float

concept ProcessQuality subConceptOf { KPI, UserAssessmentBasedAttribute}
     value ofType _float

concept ProcessCosts subConceptOf { KPI, SumAttribute}
     value ofType _float

concept NFP subConceptOf Property
     nonFunctionalProperties
          dc#description hasValue "The generic concept from which all the
NFP will inherit"
     endNonFunctionalProperties
     hasUnit ofType Unit

concept QualitativeNFP subConceptOf NFP
     nonFunctionalProperties
          dc#description hasValue "The qualitative properties are those
which specify the quality of an artefact."
     endNonFunctionalProperties

concept QuantitativeNFP subConceptOf NFP
     nonFunctionalProperties
          dc#description hasValue "The quantitative properties are those
that may be expressed as some measurable value."
     endNonFunctionalProperties

concept StaticProperty subConceptOf NFP
     nonFunctionalProperties
          dc#description hasValue "The static properties are those that do
not change over time"
     endNonFunctionalProperties

concept Cost subConceptOf { QuantitativeNFP, priceOnto#AbsolutePrice,
STBProperty, SemiStaticProperty, SumAttribute}
     nonFunctionalProperties
          dc#description hasValue "An amount of money that needs to be
paid to the artefact provider for the execution."
```

```
     endNonFunctionalProperties
     value ofType _float

concept Duration subConceptOf { Performance,
temporalOnto#TemporalDuration, STBProperty, DynamicProperty,
CriticalPathAttribute}
     value ofType _integer

concept Security subConceptOf { QualitativeNFP, NTBProperty,
StaticProperty, MinimalMaximalAttribute, FunctionalityDependentAttribute}
     nonFunctionalProperties
          dc#description hasValue "The security NFP is divided into
dimensions: identification and confidentiality."
     endNonFunctionalProperties

concept Authorization subConceptOf Security
     nonFunctionalProperties
          dc#description hasValue "Users (or other services) should be
authorized so that they only can access the protected services."
     endNonFunctionalProperties
     value ofType _string

//requirements branch

concept Requirement

concept Constraint subConceptOf Requirement
     constraintsValueOf ofType  (0 1) Property
     hasConstraintValue ofType  (0 1) _string
     hasUnit ofType  (0 1) Unit
     hasComparator ofType  (0 1) Comparator

concept FuzzyConstraint subConceptOf Constraint

concept StringentConstraint subConceptOf Constraint

concept Preference subConceptOf Requirement

concept PropertyPreference subConceptOf Preference
     optimizeValueOf ofType  (1 1) Property
     hasPriority ofType  (1 1) _integer

//profiles

concept Profile

concept ProcessProfile subConceptOf Profile
     hasHorizon ofType Horizon
     hasPreferencesAssigned ofType Preference
     hasConstraintsAssigned ofType Constraint

concept OrganizationProfile subConceptOf Profile
     hasPreferences ofType PropertyPreference
     hasPartners ofType Organization
     hasAlliances ofType Organization
     hasCompetitor ofType Organization
     usesServices ofType SWS
     hasTrustedParties ofType TrustedParty
     hasSignedSLAs ofType SLA

//process artifacts

concept ProcessArtefact
     hasConstraints ofType Constraint

concept Process subConceptOf ProcessArtefact
     hasOwner ofType BusinessUser
     hasProcessProfile ofType ProcessProfile
     hasSequence ofType Sequence

concept ProcessFragment subConceptOf ProcessArtefact
     belongsToProcess ofType Process

concept BlockStructure subConceptOf ProcessArtefact

concept Task subConceptOf BlockStructure
     hasGoal ofType Goal
     hasCriticalProperty ofType NFP
```

```
     implementedByTheSameProviderAs ofType Task
     notImplementedByTheSameProviderAs ofType Task
     implementedByTheSameServiceAs ofType Task
     notImplementedByTheSameServiceAs ofType Task
     hasProperty ofType Property
     implementedByTrustedPartyWithTrustAtLeast ofType _float
     implementedByServiceProvider impliesType Organization

concept Sequence subConceptOf BlockStructure
     hasElements ofType BlockStructure

concept Parallel subConceptOf BlockStructure
     hasBranches ofType  (2 *) Branch

concept XOR subConceptOf BlockStructure
     hasBranches ofType  (2 *) ConditionalBranch

concept OR subConceptOf BlockStructure
     hasBranches ofType  (2 *) ConditionalBranch

concept Branch subConceptOf SupportingArtefacts
     hasElements ofType BlockStructure

concept Loop subConceptOf BlockStructure
     hasElements ofType BlockStructure
     hasCounter ofType _integer

// service

concept Service
     hasProfile ofType ServiceProfile
     fulfillsSWSGoal ofType Goal
     hasProvider ofType Organization
     hasSLO ofType SLO
     needsToBeUsedTogetherWith ofType Service
     belongsToBundle ofType Bundle

//actor group

concept Actor
     hasName ofType _string

concept Organization subConceptOf Actor
     hasOrganizationProfile ofType OrganizationProfile

concept BusinessUser subConceptOf Actor
     belongsToOrganization ofType Organization
     hasUserProfile ofType UserProfile
     hasDefaultUserProfile ofType UserProfle

//supporting artifacts

concept SupportingArtefacts

concept Unit subConceptOf SupportingArtefacts

concept TimeUnit subConceptOf Unit

concept SLA subConceptOf SupportingArtefacts
     relatesToService ofType Service
     hasClient ofType Organization
     hasProvider ofType Organization
     hasSLO ofType SLO
     hasDescription ofType _string

concept Bundle subConceptOf SupportingArtefacts
     hasServices ofType Service
     hasDiscountPrice ofType Cost
     hasProvider ofType Actor
     hasBenfits ofType _string

concept Comparator subConceptOf SupportingArtefacts
     value ofType _string

//set of instances

instance miliseconds memberOf TimeUnit
instance seconds memberOf TimeUnit
```

```
instance LongTermProcess memberOf Horizon
     hasName hasValue "Long Term Process"

instance ShortTermProcess memberOf Horizon
     hasName hasValue "Short Term Process"

instance moneyTransfer memberOf PaymentMethod
     value hasValue "moneyTransfer"

instance byCheque memberOf PaymentMethod
     value hasValue "byCheque"
instance Euro memberOf Currency

instance daily memberOf ChargingFrequency
     value hasValue "daily"

instance monthly memberOf ChargingFrequency
     value hasValue "monthly"

instance GreaterThan memberOf Comparator
     value hasValue ">"
```

8.3 Exemplary competency questions

What is the value of a characteristic X of a given service?
What are the constraints connected with the task A?
What preferences has a user towards the given process model?
Is the requirement strict or fuzzy?
What is the type of the property?
What is the character of the property?
What is the changeability?
Which elements does the process consist of?
Which workflow structures are present?
Which service may realize the goal of a certain task?
Who is the provider of the service?
What are the requirements of the organization towards its processes?
What kind of collaboration exists between partners?
Which properties are defined within the SLA?
What are the maximal/minimal/average values of certain property?
What is the unit of the property?
How the given parameter contributes to the overall quality of service?
Which layer the given property belongs to?
Which property should be optimized during selection?
Who are the competitors of an organization?
Are there any constraints assigned to tasks or any other process structure?
Which aggregation method should be used to the given property?
What is the relation between X business property and more technical layer?
What is the charging frequency that may be applied?
What is the payment model used by a service?
What is the trust level assigned to a given partner?
Who are the trusted parties of an organization in question?
What are the preferences of a user towards his processes?
Are there any global constraints assigned to a process?
What is the probability of execution of a given alternative structure?
How many cycles does the loop have?

What are the critical properties for a given property?
What is the goal fulfilled by a given service?
What is a service characteristic?
Who is the provider of the service in relation to our organization?
All services fulfilling certain goal?
What is the horizon of the process?
Is the process owner price sensitive?
Which properties are of interest to a given process model?
What is the organization profile associated with the given business user profile?
What categories of KPIs can be identified?
Which properties constitute performance characteristic of an artefact?
Which property is constrained by the given constraint? What is the desired value?
Which bundle does the service belongs to? With which services? What is the discount value?

8.4 Exemplary WSML queries used to evaluate the information model

1. What is the value of a characteristic X of a given service?
Query for listing all properties assigned to a given service "s1"

```
?x memberOf Property and ?x[value hasValue ?value] and ?x[hasUnit hasValue
?unit] and ?z memberOf ServiceProfile and s1[hasProfile hasValue ?z] and
?z[hasProperty hasValue ?x]
```

Query for listing only cost associated with a given service "s1"

```
?x memberOf Cost and ?x[value hasValue ?value] and ?x[hasUnit hasValue
?unit] and ?z memberOf ServiceProfile and s1[hasProfile hasValue ?z] and
?z[hasProperty hasValue ?x]
```

Query for listing all properties assigned to all services:

```
?x memberOf Property and ?x[value hasValue ?value] and ?x[hasUnit hasValue
?unit] and ?y memberOf Service and ?z memberOf ServiceProfile and
?y[hasProfile hasValue ?z] and ?z[hasProperty hasValue ?x]
```

2. What are the constraints connected with the task A?

```
?x memberOf Constraint and taskA[hasConstraints hasValue ?x]
```

3. What preferences has a user towards the given process model?

```
?x memberOf Preference and ?profile[hasPreference hasValue ?x] and
process[hasProcessProfile hasValue ?profile]
```

4. Is the requirement strict or fuzzy?

```
Requirement1 memberOf ?x and ?x subConceptOf Constraint
```

5. What is the type of the property?; What is the character of the property?; What is the changeability?

```
PropertyX memberOf ?y and ?x subConceptOf Property
```

Or ask whether a given property is a membetOf a certain type. If no result is returned, then it is not.
6. Which elements does the process consist of?

```
?x memberOf Sequence and process1[hasSequence hasValue ?x] and
?x[hasElements hasValue ?y]
```

7. Which workflow structures are present?

```
?x memberOf Sequence and process1[hasSequence hasValue ?x] and
?x[hasElements hasValue ?y] and ?y memberOf ?z and ?z subConceptOf
BlockStructure
```

8. Which services may realize the goal of a certain task?

```
Task1[hasGoal hasValue ?x] and ?z memberOf Service and ?z[fulfillsSWSGoal
hasValue ?x]
```

9. Who is the provider of the service?

```
Service1[hasProvider hasValue ?x]
```

10. What are the requirements of the organization towards its processes?

```
Organization1[hasOrganizationProfile hasValue ?x] and ?x[hasPreferences
hasValue ?z]
```

11. What kind of collaboration exists between partners?

```
Organization1[hasOrganizationProfile hasValue ?x] and ?x[hasCompetitors
hasValue ?z]

Organization1[hasOrganizationProfile hasValue ?x] and ?x[hasAlliances
hasValue ?z]

Organization1[hasOrganizationProfile hasValue ?x] and
?x[hasPartnershasValue ?z]
```

12. Which properties are defined within the SLA?

```
SLA1[hasSLO hasValue ?x] and ?x[hasPropertyValuesGuarantees hasValue ?y]
```

13. What are the maximal/minimal/average values of certain property of a certain service?

```
Service1[hasProfile hasValue ?x] and ?x[hasProperties hasValue ?z] and ?z
memberOf avgDuration
```

14. What is the unit of the property?

```
Property1[hasUnit hasValue ?z]
```

15. How the given parameter contributes to the overall quality of service? See no. 5
16. Which level the given property belongs to?

```
Property1[relatesToLevel hasValue ?z]
```

17. Which aggregation method should be utilised? - See no. 5.
18. Which property should be optimized during selection?

```
Process1[hasProcessProfile hasValue ?x] and ?x[hasPreferencesAssigned
hasValue ?y]
```

19. Who are the competitors of an organization?

```
Organization1[hasOrganizationProfile hasValue ?x] and ?x[hasCompetitor
hasValue ?y]
```

20. Are there any constraints assigned to tasks or any other process structure belonging to a given process?

```
Process1[hasSequence hasValue ?x] and ?x[hasElements hasValue ?y] and
?y[hasConstraints hasValue ?z]
```

21. What is the relation between X business property and more technical layer?

```
businessPropertyX[relatesTo hasValue ?x]

businessPropertyX[relatesToLevel hasValue ?x]

businessPropertyX[hasCalculation hasValue ?x]
```

22. What is the charging frequency that may be applied?

```
?x memberOf ChargingFrequency
```

23. What is the payment model used by a service?

```
Service1[hasProfile hasValue ?x] and ?x[hasProperty hasValue ?z] and ?z
memberOf PaymentModel
```

24. What is the trust level assigned to a given partner?

?z memberOf TrustedParty and ?z[relatesToProvider hasValue provider1] and ?z[hasTrustValue hasValue ?y]

25. Who are the trusted parties of an organization in question?

Organization1[hasOrganizationProfile hasValue ?x] and ?x[hasTrustedParties hasValue ?y]

26. What are the preferences of a user towards his processes?

Process1[hasOwner hasValue ?x] and ?x[hasProcessProfile hasValue ?y] and ?y[hasPreferences hasValue ?z]

27. Are there any global constraints assigned to a process?

Process1[hasProcessProfile hasValue ?x] and ?x[hasConstraints hasValue ?y]

28. What is the probability of execution of a given alternative structure?

Xor1[hasBranches hasValue ?x] and ?x[hasProbability hasValue ?y]

29. How many cycles does the loop have?

Loop1[hasCounter hasValue ?z]

30. What are the critical properties for a given task?

Task1[hasCriticalProperty hasValue ?x]

31. What is the goal fulfilled by a given service?

Service1[fulfillsSWSGoal hasValue ?x]

32. What is a service characteristic?

Service1[hasProfile hasValue ?x]

33.Who is the provider of the service in relation to our organization?

Service1[hasProvider hasValue x?] and Organization1[hasOrganizationProfile hasValue ?y] and ?y[hasCompetitors hasValue ?x]

Service1[hasProvider hasValue x?] and Organization1[hasOrganizationProfile hasValue ?y] and ?y[hasAlliances hasValue ?x]

Service1[hasProvider hasValue x?] and Organization1[hasOrganizationProfile hasValue ?y] and ?y[hasPartners hasValue ?x]

Service1[hasProvider hasValue x?] and Organization1[hasOrganizationProfile hasValue ?y] and ?y[hasTrustedParties hasValue ?z] and ?z[relatesToProvider hasValue ?x]

34. Which services fulfil certain goal?

?x memberOf Service and ?x[fulfillsSWSGoal hasValue certainGoal]

35. What is the horizon of the process?

Process1[hasProcessProfile hasValue ?x] and ?x[hasHorizon hasValue ?y]

36. Is the process owner price sensitive?

Process1[hasOwner hasValue ?z] and ?z[hasDefaultProfile hasValue ?y]

37. Which properties are of interest to a given process model?

Process1[hasProcessProfile hasValue ?x] and ?x[hasConstraints hasValue ?z] and ?z[constraintsValueOf hasValue ?y]

Process1[hasProcessProfile hasValue ?x] and ?x[hasPreferences hasValue ?z] and ?z[optimizeValueOf hasValue ?y]

Process1[hasOwner hasValue ?x] and ?x[hasUserProfile hasValue ?y] and ?y[hasPreferences hasValue ?z] and ?z[optimizeValueOf hasValue ?p]

Process1[hasOwner hasValue ?x] and ?x[belongsToOrganization hasValue ?y] and ?y[hasOrganizationProfile hasValue ?z] and ?z[hasPrefrences hasValue ?p] and ?p[optimizeValueOf hasValue ?property]

38. What is the organization profile associated with the given business user profile?

```
?x memberOf BusinessUser and ?x[hasUserProfile hasValue Profile1] and
?x[belongsToOrganization hasValue ?y] and ?y[hasOrganizationProfile
hasValue ?z]
```

39. What categories of KPIs can be identified?

```
?x subConceptOf KPI
```

40. Which properties constitute performance characteristic of an artefact?

```
?x subConceptOf Performance
```

41. Which property is constrained by the given constraint? What is the desired value?

```
Constraint1[constraintsValueOf hasValue ?z] and
Constraint1[hasConstraintValue hasValue ?y]
```

42. With which services does the given service S1 create a bundle with? What is its price?

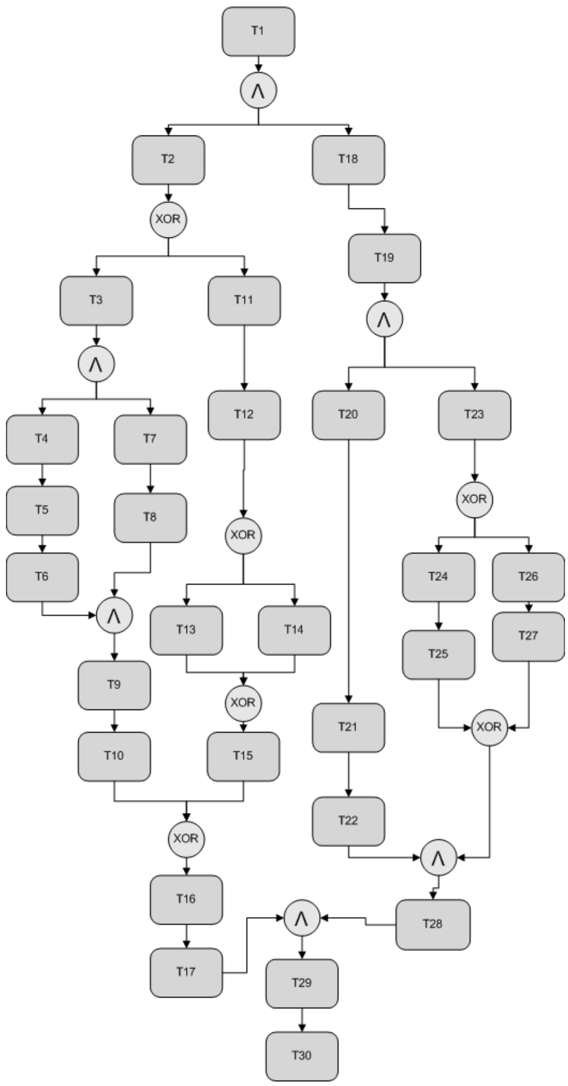
```
S1[belongsToBundle hasValue ?x] and ?x[hasServices hasValue ?y] and not
?y=S1 and ?x[hasDiscountPrice hasValue ?z] and ?z[value hasValue ?g]
```

8.5 VoIP Scenario in WSML – excerpt with the process structure

```
instance VoIPProcess memberOf Process
       hasOwner hasValue VoIPProcessOwner
       hasProcessProfile hasValue VoIPProcess_Profile
       hasSequence hasValue SequenceVoIP

instance TPSA memberOf Organization
       hasName hasValue "Telekomunikacja Polska SA"
       hasOrganizationProfile hasValue TPSAProfile

instance VoIPProcessOwner memberOf BusinessUser
       belongsToOrganization hasValue TPSA
       hasUserProfile hasValue VoIPProcessOwnerProfile

instance SequenceVoIP memberOf Sequence
       hasElements hasValue {Parallel_VoIP_Verification,
XOR_VoIP_After_Verfication}

instance Parallel_VoIP_Verification memberOf Parallel
       hasBranches hasValue {Branch_Technical, Branch_Formal}

instance Branch_Technical memberOf Branch
       hasElements hasValue  VerifyFormalRequirementsTask

instance Branch_Formal memberOf Branch
       hasElements hasValue  VerifyTechnicalRequirementsTask

instance XOR_VoIP_After_Verfication memberOf XOR
       hasBranches hasValue {Branch_Verification_Failed,
Branch_Verification_Succedeed}

instance Branch_Verification_Failed memberOf ConditionalBranch
       hasElements hasValue NotifyCustomerTask
       hasProbability hasValue _float("0.1")

instance Branch_Verification_Succedeed memberOf ConditionalBranch
       hasElements hasValue {CreateOrderTask, Parallel_Prepare_Elements,
SendViaCourierTask, Parallel_Activate, NotifyCustomerTask1}
       hasProbability hasValue _float("0.1")

instance Parallel_Prepare_Elements memberOf Parallel
       hasBranches hasValue {Branch_Agreement, Branch_Hardware}

instance Branch_Agreement memberOf Branch
```

```
        hasElements hasValue PrepareContractTask

instance Branch_Hardware memberOf Branch
        hasElements hasValue XOR_PrepareHardware

instance XOR_PrepareHardware memberOf XOR
        hasBranches hasValue {Branch_Hardware, Branch_NoHardware}

instance Branch_Hardware memberOf ConditionalBranch
        hasElements hasValue PrepareHardwareTask

instance Branch_NoHardware memberOf ConditionalBranch
        hasElements hasValue EmptyTask

instance Parallel_Activate memberOf Parallel
        hasBranches hasValue {Branch_Platform, Branch_Billing}

instance Branch_Platform memberOf Branch
        hasElements hasValue ActivateVoIPPlatformTask

instance Branch_Billing memberOf Branch
        hasElements hasValue ActivateVoIPBillingTask
```

8.6 An exemplary process model used within test cases



8.7 Author's publications

1. ABRAMOWICZ, W., EKELHART, A., FENZ, S., KACZMAREK, M., TJOA, A. M., WEIPPL, E. R. & ZYSKOWSKI, D. (2007) Security Aspects in Semantic Web Services Filtering IN GABRIELE KOTSIS, D. T., ERIC PARDEDE, ISMAIL K. IBRAHIM (Ed.) 9th International Conference on Information Integration and Web-based Applications and Services. Jakarta, Austrian Computer Society.
2. ABRAMOWICZ, W., FILIPOWSKA, A., KACZMAREK, M. & KACZMAREK, T. (2008) A Need for Business Assessment of Semantic Web Services' Applications in Enterprises. Business Process Management Workshops : BPM 2007 International Workshops, BPI, BPD, CBP, ProHealth, RefMod, semantics4ws, Brisbane, Australia, September 24, 2007, Revised Selected Papers. Springer Verlag.
3. ABRAMOWICZ, W., FILIPOWSKA, A., KACZMAREK, M., KACZMAREK, T., KOWALKIEWICZ, M., RUTKOWSKI, W., WIELOCH, K. & ZYSKOWSKI, D. (2006) Service interdependencies: insights into use cases for service compositions. IN DEBENHAM, J. (Ed.) IFIP - Professional Practice in Artificial Intelligence. Springer Verlag.
4. ABRAMOWICZ, W., FILIPOWSKA, A., KACZMAREK, M., PEDRINACI, C., STARZECKA, M. & WALCZAK, A. (2008) Organization Structure Description for the Needs of Semantic Business Process Management. 3rd International Workshop on Semantic Business Process Management. Tenerifa, CEUR WS series.
5. ABRAMOWICZ, W., FILIPOWSKA, A., KACZMAREK, M., STARZECKA, M. & WALCZAK, A. (2008) Semantic Enterprise Description for the Needs of Business Process Automation Proceedings of SemBPM Workshop. Turku, IEEE.
6. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M., PALMA, R. & ZYSKOWSKI, D. (2008) NFP ontology for discovery and sharing Web services in distributed registries. 22nd International Conference on Advanced Information Networking and Applications : Workshops. Gino-wan, IEEE.
7. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2006) Automatic Web Services Interactions -Requirements, Challenges and Limits from the F-WebS System Perspective. IN ABRAHAM, A. & HAN, S. Y. (Eds.) International Conference on Next Generation Web Services Practices. Seoul, Korea, IEEE Press.
8. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2006) Filtering of Semantic Web services with F-WebS system. IN LI, G., LIANG, Y. & RONCHETTI, M. (Eds.) The Semantic Web: ASWC 2006 Workshop. Beijing, Jilin University Press.
9. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2007) Architecture for Web Services Filtering and Clustering. Second International Conference on Internet and Web Applications and Services (ICIW 2007). Mauritius, IEEE Press.
10. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2008) Design of Web Services Filtering and Clustering System. International Journal On Advances in Internet Technology, 1, 30-30.
11. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2008) E-Marketplace for Semantic Web Services. Service-Oriented Computing - ICSOC 2008, 6th International Conference, Sydney, Australia, December 1-5, 2008. Springer Berlin/Heidelberg.
12. ABRAMOWICZ, W., HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2009) Semantic Modelling of Collaborative Business Processes. Proceedings of International Conference on Informaiton, Process and Knowledge Management (eKNOW 2009). Cancun, IEEE.
13. ABRAMOWICZ, W., KACZMAREK, M., KOWALKIEWICZ, M. & ZYSKOWSKI, D. (2005) A survey of QoS computation for Web services Profiling. IN DASCALU, S. (Ed.) 18th International Conference on Computer Applications in Industry and Engineering

(ISCA). Honolulu, USA, ISCA (International Society for Computers and their Applications).

14. ABRAMOWICZ, W., KACZMAREK, M., KOWALKIEWICZ, M. & ZYSKOWSKI, D. (2006) Architecture for Service Profiling. IN MALU CASTELLANOS, J. Y. (Ed.) SCW '06: Proceedings of the IEEE Services Computing Workshops. IEEE Computer Society.
15. ABRAMOWICZ, W., KACZMAREK, M. & ZYSKOWSKI, D. (2006) Duality in Web Services Reliability. IN DINI, P., LORENZ, P., ROMAN, D. & FREIRE, M. (Eds.) Proceedings of the Advanced International Conference on Telecommnications and International Conference on Internet and Web Applications and Services 2006 (AICT/ICIW 2006) Guadeloupe, French Caribbean, IEEE Press.
16. ABRAMOWICZ, W., KACZMAREK, M. & ZYSKOWSKI, D. (2008) Profiling of Web Services to Measure and Verify their Non-Functional Properties. Managing Web Service Quality, Measuring Outcomes and Effectiveness. IGI Publishing.
17. ABRAMOWICZ, W., KACZMAREK, M., ZYSKOWSKI, D., GWIZDALA, J. & GODLEWSKA, A. (2005) Application-oriented Web services filtering. IN ABRAHAM, A., HAN, S. Y., DU, D. H.-C. & PAPRZYCKI, M. (Eds.) International Conference on Next Generation Web Services Practices. Seoul, Korea, IEEE Press.
18. BORN, M., FILIPOWSKA, A., KACZMAREK, M., MARKOVIC, I., STARZECKA, M. & WALCZAK, A. (2008) Business Functions Ontology and its Application in Semantic Business Process Modelling. Proceedings of the 19th Australasian Conference on Information Systems. Christchurch, Australasian Computer Society.
19. FILIPOWSKA, A., KACZMAREK, M., STOLARSKI, P., WIELOCH, K. & ZYSKOWSKI, D. (2007) Kompozycja procesów biznesowych z wykorzystaniem usług sieciowych. Kapitał ludzki i wiedza w gospodarce - wyzwania XXI wieku. Poznań, AE.
20. HANIEWICZ, K., KACZMAREK, M. & ZYSKOWSKI, D. (2008) Semantic Web Services Applications - A Reality Check. Special Issue: Service-Oriented Architectures and Web Services. Wirtschaftsinformatik (WI) 1, 39-46.
21. KACZMAREK, M., FILIPOWSKA, A., ABRAMOWICZ, W. & KACZMAREK, T. (2007) Semantically enhanced Business Process Modelling Notation. IN HEPP, M., HINKELMANN, K., KARAGIANNIS, D. & KLEI, R. (Eds.) Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007) Wien, CEUR WS.
22. KACZMAREK, M., GWIZDAŁA, J. & KLIBER, A. (2006) Usługi sieciowe jako nowy rodzaj usług komercyjnych świadczonych na rynku elektronicznym. Instrumentalne, instytucjonalne i organizacyjne problemy funkcjonowania gospodarki rynkowej. Debiuty Ekonomiczne. Poznań, Wydawnictwo AE.
23. KACZMAREK, M., KACZMAREK, T., VULCU, G., WEBER, I., WETZSTEIN, B., MA, Z., ZYSKOWSKI, D. & AL., E. (2008) D3.5 SUPER Deliverable on Business Process Composition. SUPER Consortium.
24. KACZMAREK, M., ZYSKOWSKI, D., HANIEWICZ, K. & ABRAMOWICZ, W. (2009) SOA, Web services and QoS - perception on the Polish market. Technical Report. Department of Information Systems, Poznan University of Economics.
25. KOWALKIEWICZ, M., LUDWIG, A., KACZMAREK, M. & ZYSKOWSKI, D. (2005) Documented mechanisms for Dynamic Service Profiling and Agreement Life-cycle Management. Adaptive Services Grid Project Deliverables. Potsdam, ASG internal deliverable, FP6-IST-004617.
26. WETZSTEIN, B., MA, Z., FILIPOWSKA, A., KACZMAREK, M., BHIRI, S., LOSADA, S., LOPEZ-COB, J.-M. & CICUREL, L. (2007) Semantic Business Process Management: A Lifecycle Based Requirements Analysis. IN HEPP, M., HINKELMANN, K., KARAGIANNIS, D., KLEIN, R. & STOJANOVIC, N. (Eds.) Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007) Martin Hepp, Knut Hinkelmann, Dimitris Karagiannis, Rüdiger Klei ed. Wien, CEUR-WS.org.

27. FILIPOWSKA, A., KACZMAREK, M., KOWALKIEWICZ, M., ZHOU, X., BORN, M. (2009) Procedures and guidelines for evaluation of BPM methodologies. Business Process Management Journal, Vol. 15, No. 3, pp. 336-357, Emeral Group Publishing.
28. FILIPOWSKA, A., KACZMAREK, M., HEPP, M., MARKOVIC, I., (2009) Organisational Ontology Framework for Semantic Business Process Management, In proceedings Of Business Information Systems, ABRAMOWICZ, W. (Eds), Springer, pp. 1-12, ISBN 978-3-642-01189-4.
29. FILIPOWSKA, A., KACZMAREK, M., STEIN, S. (2009) Semantically Annotated EPC within Semantic Business Process Management. In the proceedings of Business Process Management Workshops, ARDAGNA, D., MECELLA, M., YANG, J. (Eds), Springer, pp. 486-497, ISBN 978-3-642-00327-1.
30. VARGAS-VERA, M., NAGY, M., ZYSKOWSKI, D., HANIEWICZ, K., ABRAMOWICZ, W., KACZMAREK, M. (2009) Challenges on Semantic Web Services. In Handbook of Research on Social Dimensions of Semantic Technologies and Web Services, CUNHA, M.M., OLIVEIRA, E.F., TAVARES, A.J., FERRIERA, L.G. (Eds), Hershey : Information Science Reference, pp. 25-48, ISBN 978-1-60566-650-1
31. ABRAMOWICZ, W., HANIEWICZ, K., HOFMAN, R., KACZMAREK, M., ZYSKOWSKI, D., (2009) Decomposition of SQuaRE – Based Requirements for the Needs of SOA Applications. In Trends in Communication Technologies and Engineering Science, AO, S-I., HUANG, X., WAI, P-K., A. (Eds), Lecture Notes in Electrical Engineering, Springer, pp. 81-94.