Uniwersytet Ekonomiczny w Poznaniu
Wydział Informatyki i Gospodarki Elektronicznej

Web Service Representation and Retrieval Designed for Service Oriented Enterprises

Konstanty Haniewicz

Promotor: prof. dr hab. Witold Abramowicz

Luty 2013

Abstract

The main objective of this thesis is to present a method of description that satisfies a varied set of needs issued by Web service market participants. This method of description is to raise the quality of the whole process of Web service description and retrieval for the sake of organizations following the Service Oriented Architecture paradigm taking into account various issues important from the information economics perspective.

To validate this statement a methodology built upon the Concept-Knowledge theory, Design Science and the traditional research tools was applied.

The research activities were focused on establishing the Key Requirement Aspects from the domain literature supported with a variety of business users willing to participate in informal interviews. The requirements formulated took into account the varying utility of a Web service for its users in the electronic economy setting.

The successful extraction of the Key Requirement Aspects allowed for critical analysis of the available solutions used to describe and retrieve Web service functionality. The critical analysis enabled the author to deduce that there is not a single initiative that could reach a satisfactory level of the fulfilment of Key Requirement Aspects, both individually and in general.

The previously given critique provided the foundation to the designed model that aimed to introduce a set of mechanisms that should minimize negative traits of a number of initiatives and leverage their advantages.

The model introduces the mechanisms that are capable of delivering the desired results. Their capabilities were tested for compliance with the Key Requirement Aspects. The evaluation was performed so that their most crucial qualities could be emphasized in a direct comparison to the alternatives where such direct comparison was possible. The model and its mechanisms were capable of diminishing the potential information asymmetry among various market participants. Therefore, they are an invaluable tool for markets struggling with unwanted obstacles preventing flawless economic exchange.

The above allowed for dissertation's thesis validation. What is more, the research accomplished is a great opportunity for further studies and a first step in a case study of an adoption of the prepared model and its supporting mechanisms in organisations willing to invest in new solutions.

Acknowledgements

I would like to thank a great number of people that were of huge help at various stages of writing this dissertation.

I direct special thanks to dr Monika Kaczmarek for all the time she devoted to talks on the subjects concerning this work and commenting the intermediate results.

I also would like to thank a host of willing interviewees that shared with me their experiences concerning their work experience and knowledge of various organizations and systems used there to manage various information resources.

I would also like to thank the following people: Aleksandra Gawęcka, Olga Zdankiewicz, Olga Nadskakuła, Wojciech Rutkowski, Radosław Ruciński, Łukasz Sosnowski, Tomasz Sierszchuła, Dominik Zyskowski. They provided me with invaluable access to their and their colleagues expertise. What is most important, they patiently withstood a constant stream of inquiries never showing annoyance and always complied with the strangest requests regarding this work on my behalf.

To my parents.

Contents

List of Tables

List of Figures

Nomenclature

C-K Concept-Knowledge Theory

IOPE Inputs, Outputs, Preconditions, Effects

IR Information Retrieval

KRA Key Requirement Aspects

LCA Local Context Anchoring

LCV Locally Controlled Vocabulary

NFP Non-functional properties

OWL Web Ontology Language

QoS Quality of Service

RDF Resource Description Framework

SE Supervising Entity

SOA the Service Oriented Architecture

SU Suborganizational Unit

SWS Semantic Web services

UDDI Universal Description Discovery and Integration

WSDL Web Service Description Language

WSMO Web service Modelling Ontology

Chapter 1

Introduction

1.1 Motivation

Independent of its size, every organisation produces considerable quantities of data in a variety of forms. Documents, audio recordings, video streams, all that is produced, stored and accessed on a daily basis. The attempts to quantify the sheer amount of data produced by the human civilization present results likely to be beyond the grasp of many [1].

The volume of data is a result of, among others, the thriving development of information systems resulting from advances in the computer science field [Hoonlor et al., 2012] that transformed the landscape of enterprises. There is no corporation that is able to function without aid of advanced Information Technology [Lederer and Mendelow, 1988]. The amount of data that is processed every second is exuberant and cannot be fully measured in detail. Yet, there are some estimates such as the one cited below. The study given in (The World's Technological Capacity to Store, Communicate, and Compute Information [Hilbert and Lopez, 2011]) reports that: *[Authors] estimated the world's technological capacity to store, communicate, and compute information, tracking 60 analog and digital technologies during the period from 1986 to 2007. In 2007, humankind was able to store* 2.9×1020 *optimally compressed bytes, communicate almost* 2×1021 *bytes, and carry out* 6.4×1018 *instructions per second on general-purpose computers. General-purpose computing capacity grew at an annual rate of 58%. The world's capacity for bidirectional telecommunication grew at 28% per year, closely followed by the increase in globally stored information (23%).*

[1] `http://www.economist.com/node/15557443`

Humankind's capacity for unidirectional information diffusion through broadcasting channels has experienced comparatively modest annual growth (6%).

To further visualise the sheer size of the volume of data that is to be processed, consider the following excerpt from a report [James E. Shirt and Baru, 2011]:

Three years ago[2]*, the world's 27 million business servers processed 9.57 zettabytes, or 9 570 000 000 000 000 000 000 bytes of information. Researchers at the School of International Relations and Pacific Studies and the San Diego Supercomputer Center at the University of California, San Diego, estimate that the total is equivalent to a 5.6-billion-mile-high stack of books stretching from Earth to Neptune and back to Earth, repeated about 20 times. By 2024, business servers worldwide will annually process the digital equivalent of a stack of books extending more than 4.37 light-years to Alpha Centauri, according to a report compiled by the scientists.*

The cited excerpts underline the overall amount of existing data that has to be processed by Information Systems. One can state that we are beyond a point of return in terms of automation and interoperation of contemporary Information Systems. Without further advancements in these fields optimal decisions cannot be made, not due to the lack of data but its overflow and lack of a confidence that all relevant information was taken into account [Eppler and Mengis, 2003].

What is more, challenged with such numbers, one has to realise that a scheme for a feasible retrieval must be available to any particular data type, both stored and processed. The described plenty resulted in blooming of document retrieval techniques that have matured over last 50 years [Van Rijsbergen, 1979, Baeza-Yates and Ribeiro-Neto, 1999, Manning et al., 2008, Sanderson and Croft, 2012].

The traditional model of data retrieval revolves around a notion of a document and an index. Indices are built to robustly answer a question whether a given term is present in some document or documents. The more advanced the Information Retrieval system is, the more capabilities it has. At the moment of writing of this thesis an implementation of IR system is capable of not only answering the most basic questions of the mentioned above nature but also can retrieve whole phrases, manage spelling errors and variants, classify results, rank the results according to some relevance method and more [Manning et al., 2008].

In this quantity of data, Web services are yet another entity that must be robustly retrieved, yet its retrieval must include additional properties endemic to this particular information asset.

[2] 2007 - author's note

1.1.1 Web services as electronic goods

A Web service is loosely coupled, reusable software component that semantically encapsulates discrete functionality and is distributed and programmatically accessible over standard Internet protocols [Staab et al., 2003].

In essence, as is discussed in later sections in more detail, it is an interface to some functionality, which implementation details are unimportant in contrast to its usability. Standard tools and conventions introduced along with a Web service, provided a method of unified description of various Web service aspects. This was gathered in the Web Service Description Language document (WSDL).

Very soon, the WSDL document became insufficient for various applications envisioned by researchers and industry leaders, mainly due to a fact that Web services become a subsequent reincarnation of technology unifying Information Technology systems [Hansen et al., 2003]. They quickly took over a number of older technologies and became a de facto standard for systems intercommunication [Vinoski, 2002, Yu et al., 2008].

In addition, several new opportunities were observed, such as an ability to compose applications made of Web services that encapsulate functionality without burdening a user with implementation details. What is more, additional extensions such as the automated Web service composition based on Semantic Web services became an important topic in a global research discussion [Traverso and Pistore, 2004, Sirin and Parsia, 2004].

All this resulted in a considerable amount of interest in Web services along with a great number of publications and events devoted to them. One may risk a statement that years 1999 to 2009 were a decade of Web services, where all types of extensions, enhancements and systems using them were presented and discussed. Web services are still an active research area at the time of writing this thesis [Jiang et al., 2012, Feng and Fan, 2012, Lo et al., 2012, Harshavardhanan et al., 2012, Baghdadi, 2012, Tamilarasi and Ramakrishnan, 2012]. There are new angles that were previously left unaddressed and completely new research directions appear.

The actual volume and scale of ready Web services that are fully operational and are used on daily basis is not known. The research community tried to present various estimates, yet in most of the cases they are only applicable to the open Internet [Al-Masri and Mahmoud, 2008, Steinmetz et al., 2009, Hagemann et al., 2007, Song et al., 2007]. One can only think that numerous organizations store and use tens of thousands of Web services.

An architecture based on services helps organisations that implement it become more flexible, more adaptable and manage the cost of various types of their operations [Papazoglou and Heuvel, 2007]. As is given in greater detail in further parts of the dissertation, Service Oriented Architecture is important due to the following traits being a foundation for the above-mentioned benefits [Yu et al., 2008]:

- modularity,
- encapsulation,
- loose coupling,
- separation of concerns,
- reuse,
- composability,
- single implementation.

The task of functionality description is a difficult one not only for the machines, but also for human beings [Geurts, 1997]. Whether some artifact matches one's needs, can be only validated by application of a given artifact to a concrete task one is willing to accomplish.

One of the most important achievements in the human evolution is the language that allows for the communication of abstract ideas so that two different members of the species can comprehend given information in a manner enabling them to identify an act or an object unanimously [Christiansen and Kirby, 2003].

As the abstract concepts such as manhood, courage, love cannot be directly mapped onto physical objects or commonly occurring natural phenomena an elaborate structure aiming at description of the world was built using the language [Gelman and Butterworth, 2005]. What is more, this structure was multiplied several thousand times due to the fact that its builders were scattered around the world where different environments affected their actions. More, it changed over time to accumulate the changes induced by the nature, the technology and the culture.

Taking into account the above, it is surely impossible to fully design and implement a scaffolding that could be used as a ultimate reference to all the systems used around the world. Even a structure that would aim for a single language and only one limited dialect securing that it shall neglect the frivolous nature of language semantics and the fertility of vocabulary, might be a task that cannot be successfully accomplished. Even in the unlikely event of success, the produced structure might be unfathomable to any interested being.

This great complexity is the reason why a number of initiatives tried to present some approximate solution that could be accepted as good-enough in terms of the

overall balance between the impossible to achieve completeness and the usability in the real life.

This dissertation is a reflection of a strong belief that it is better to satisfy a need with some probability of a success than to strive for a precise answer based on the structure covering a huge number of concepts. It is obvious that precise is better than that of some probability, yet in light of the evidence one cannot believe that the prerequisites to achieve the precise can be met.

Therefore the author stands on the position that the most important challenge of an organization willing to adhere to the SOA paradigm is to be able to make the correct choices given that the environment in which it lives is prone to a constant change. This choice is especially important when maintaining and optimizing the crucial business processes enabled by services.

Thus, the main objective of this work, is to provide a well-balanced model capable of providing desired results in a relatively short time. This original model is accompanied by the results provided by the mechanisms supporting it. These results are compared against other available approaches.

1.1.2 Approaches to the role of information in economics

There are two main approaches to information in economics. The first one goes back to the period between early 1960's and 1970's that underlines information on goods and services as a crucial element of market functioning, affecting all of the market participants [Arrow, 1984, Stigler, 1961]. The other one, is treating information as a separate entity that is of interest to economists at the same level of focus as goods and services. The later approach, underlines the utility of any given piece of information and the fact that one has to consider its type, usage and its consumer when discussing it [Allen et al., 1990, Bakos et al., 1999, Freiden et al., 1998].

The crucial difference between the two approaches lays in the role of information in economic decisions stemming from its differently defined nature. In order to provide a graphic example, lets assume that one might agree that both technology for diamond production and the insider information concerning the state of some particular organisation are both examples of information goods. Nevertheless, they are very different in terms of usage and benefits they might yield to interested parties. The later example is one that follows the original role of information in economics. It yields benefits only at a particular moment of time and only for particular people/organizations. The first one is different due to its peculiar characteristics. Technology cannot be in-

validated by its implementation. Every interested party with sufficient resources can use it. It can be stencil for other technologies. It can be refined over time. What is more, it presents new challenges, as to be able to sell or buy the proposed technology market has to propose a system of identification and verification of available information goods. Such systems are non-trivial. Situations where it may be perceived as such, usually do not consider amount of effort covered in specialised training and general education necessary to achieve the existing state of affairs [Eatwell et al., 2000].

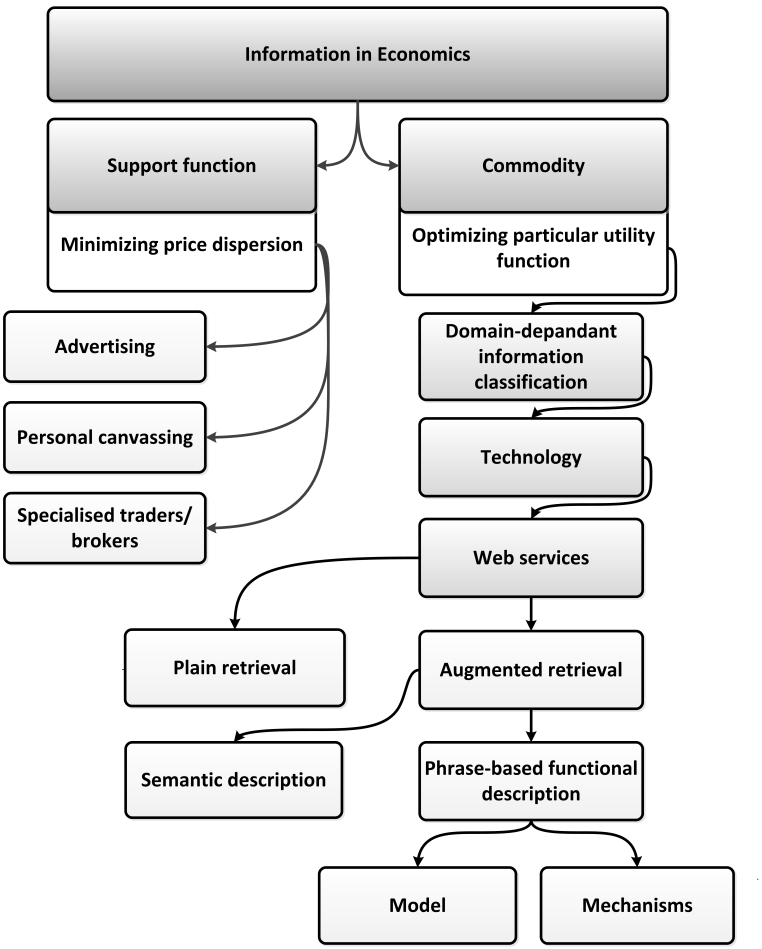

Figure 1.1: Role of information in Economics along with the presented model's place

Classical approach

Information in modern economics is of utmost importance. It hugely influences market participants. Its original importance stems from searching for a price of some desired good. Depending on the nature of the desired good information can pose a

different value for market players. In situations where a good is of high value or when one purchases considerable amount of some good with moderate price, finding the best bargain brings more benefit expressed in monetary savings.

Over half a century ago, George Stigler defined the dispersion of prices as a measure of ignorance in the market [Stigler, 1961]. The greater the dispersion the greater the ignorance. A market participant interested in optimizing his deeds by minimizing the actual price he has to pay for a desired good, will invest his time in getting data on the current situation.

Market participants designed a number of strategies that allow for minimising the price dispersion. Both sides of the supply and demand forces are interested in fighting the price dispersion (as usual, there are exempts to this rule). The most important are [3]:

- personal canvassing,
- advertising,
- specialised traders and brokers,
- domain catalogues.

Ideal situation, that might not be available in any set of circumstances is given by a market with no price dispersion. A certain type of good is valued at the same level in all possible distribution channels. The situation is ideal as participants willing to purchase a given type of good do not feel any pressure to invest time and money to investigate the market. Producers and sellers can focus on minimizing the production costs.

It is possible to view such attempts with the use of an early economic model that considers information as a key resource. This model was proposed by Marin L. Weitzman [Weitzman, 1979] and can be applied under a set of assumptions such as ability to provide a probability distribution of results obtained by choice of concrete solution. It clearly demonstrates that the presence or absence of certain type of information can be crucial factor while making decisions on future actions of some particular business enterprise.

Aside from discussion on viability of such ideal situation, one has to ponder whether identification of any given good that becomes a target for economic search, is as simple as is it is suggested. Majority of basic domain literature on economics suggests that market participants have no problem with identification of goods that they desire [Begg et al., 2008]. One can only wonder whether this is not an over-

[3] the list is compiled based on [Stigler, 1961]

simplification, especially in light of theories breaking with the rationality principle of consumers [Fehr and Tyran, 2005].

Technology as information

Economics as a science analyses human activities concerning various goods and services. It is mainly interested in production, consumption and their distribution among interested parties.

It is believed that official treatment of technology as some specified production set is not enough [Allen et al., 1990]. This definition is given by Debreu [Debreu, 1959] and concerns the products possible to achieve for some given producer. Debreu refers to the commodities being both inputs and outputs of the production process. The definition of commodity requires it to be completely specified in physical, temporal and spatial terms.

As mentioned, information is a specific type of economic good that might be perceived depending on its uses more as a good or more as a service. Some even define information as a third type of entities of interest to economics [Freiden et al., 1998]. Actions that aim to decrease cost of information retrieval positively influence the distribution of information by lowering the overall level of uncertainty. Low level of uncertainty affects positively various processes inside organisations, as they are more apt to apply the best available information. Thus, maximizing their profits and minimizing their costs.

To support their thesis, [Freiden et al., 1998] analyses goods, services and information in a variety of categories. It is obvious that hybrid nature of information allows to support their claims that it is a separate entity.

Experience demonstrates that identification of the desired good can be troublesome in a wide range of cases. Some examples are given below. Current market is plentiful of goods advertised as a butter. Nevertheless, both regular consumers and specialists agree that not everything can be deemed as butter due to various recipes and ingredients included. It is of an importance to a consumer whether vegetable oils were used, what is the percentage of animal fats in the product, what animal species provided the material for production, what is the level of potassium, what are the available pre-packaged serving sizes of the product, etc. The picture seems even more complicated when considering advanced goods that can be represented by cannons. No one can order a cannon without provision of close specification on its various characteristics. When purchasing one, a great many details have to be provided to fit an abstract cannon into concrete materialisation of one's desire. What is the main

target for the cannon operations, what range should it have, what calibre, what type of munitions can be applied, whether it can be mounted on a variety of vehicles on other support structures, how many personnel is used to operate the cannon, how many subsequent shoots can be given before issues connected with overheating occur.

The key idea behind the above examples is the fact that identification of a desired good can be complicated due to various aspects. This complexity is inherent no matter what branch of market one is involved in. The identification also incurs additional costs. Some market participants might not invest in search whether the price of identification is too high.

The situation is especially complex in areas where technology provides considerable number of new abstract entities. These abstract entities might be perceived as modern products that are not directly connected with the physical world, even though that their use can and does affect the real material world. Even the classic approach using a division of economic output into goods and services does not seem to be satisfactory. Thus, information and in broader sense knowledge, have to be perceived as official factors of production.

Information has added complexity in comparison to the original factors of production. This complexity results from the fact that usually one is less interested in particular details and more into a recipe, a process, a procedure to achieve some state. The essence of this recipe is an encryption of the algorithm. This encryption provides its owner with a power to achieve some goal.

As given in Allen [Allen et al., 1990], information goods are very diversified. Basic understanding of information good covers for an entity which might be useful only in a particular situation and in particular time. What is more, more pieces of information on some particular topic don't provide any boost to user's total utility. On the other hand, there are more complicated goods that can be useful time and time again, a good example is a recipe. It can be used a number of times to help produce desired product. Another good example is computer routine, that is an information good, yet using it time and time again doesn't make it obsolete to its user. It can even gain in value by being a cornerstone for a new refined version. Throughout history of mankind, such information good was closely related to a term of technology [Allen, 2000].

All of the above mentioned characteristics and examples lead one to believe that, when it comes to a market of technologies, price can be of secondary importance to those interested in achieving some particular goal. Far more important is their utility function correlated with their personal or organisational objectives.

The utility function of an interested user in achieving some particular goal is a set of technologies that allow for achieving this goal. They might me diversified in terms of important characteristics, such as price (where price can be given for a variety of traits) or time needed to process inputs into desired outputs. This set of technologies is treated as being optimal in terms of user's utility. It is possible to consider another sets that provide partially desired results, but it is difficult to reasonably place them in relation to the optimal set without a scale provided for particular goal. Technologies providing additional results should be contained in the optimal set.

When an optimal set satisfying user's utility function is provided, further operations can be performed in order to manage the already mentioned additional traits. As remarked, there are a number of strategies that can be applied in order to provide the most economic result in given situation.

The choice of the optimal set of technologies is not trivial due to the fact that there is some uncertainty and some cost of the operation. Technology per se, can be easy to discard. Nevertheless, if decision was made so as one technology to be used, some number of resources had to be acquired that might be elusively applicable with it.

As put by Brynjolfsson and Hitt [Brynjolfsson and Hitt, 2000], introduction of technology benefits market operation by decreasing the cost of communication and overall latency of market participants when making buy/sell decisions. Specialised technology is a leverage that when applied leads to innovations inside organisations and higher productivity.

Summary

With the above introduced approaches and their background, one can position Web services which are a particular technology thus information terms of economics.

The intra-organisation dynamics can be modelled as a market of particular goods. The larger the organisation, the more important is to minimise the ineffectiveness resulting from duplication of efforts. As postulated in the literature, the dispersion, here knowledge dispersion, might be fought with by deploying a scheme for advertising particular information goods or designating specialised units to handle the process of finding the necessary information goods.

Both approaches are widely used in modern economy. First of all, majority of computer software is advertised in a variety of channels, so that the product can reach its potential buyers. Second, there is a variety of specialised markets that with a different philosophies allow for aggregating data on available products. These models

can and are deployed on the intra-organisation layer. Nevertheless, price might not be a decisive factor when making a decision on whether to use some solution. It is due a fact, that if organisation as a whole already invested into some solution it is readily available and no additional cost in terms of production is incurred (excluding indirect costs connected with energy consumption and infrastructure maintenance). The role of the price is overtaken by solution's functionality. Only after positive identification of some set of suitable solutions, a decision can be made which should be finally chosen taking into account a variety of other factors.

Reaching again to both, the classic and the information good approaches to information, as an analogy to the price dispersion, one might envision knowledge dispersion and define it as a measure of the number of conflicting descriptions regarding some information good. The conflict can be both understood as a state where information good fulfilling particular need has more than one unique description or a state when single description is used for more than one information good. Hence, all possible efforts should be focused on diminishing its impact on the market. As with the price dispersion, there might be no hope of removing of knowledge dispersion yet the efforts should yield reasonable results in terms of the whole market.

Having dealt with positioning of information in the frames of modern economics, one has to underline that any programming routine or specific implementation embodied as a Web service is nothing more that highly particular information good that has a number of traits that incur specific treatment.

This specific treatment covers for advertising, managing the information dispersion, reducing the level of uncertainty when making decisions concerning production processes and control the cost level. The specific traits are enumerated and discussed in course of this dissertation.

1.1.3 Importance of a modern Web service discovery in electronic economy

Taking into account a total number of sources on the topic of Web service description, the estimates of research community and performed experiments on the open Internet along with informal interviews with IT and business professionals, raise a need for a new method of Web service description.

This need can be questionable when Web services are viewed as independent technology, yet when one is to consider an enormous popularity and importance of Service Oriented Architecture [4][5] it cannot be denied much longer.

As Service Oriented Architecture must be defined, one should consider the following: *Service Oriented Architecture is an architectural paradigm and discipline that may be used to build infrastructures enabling those with needs (consumers) and those with capabilities (providers) to interact via services across disparate domains of technology and ownership. Services act as the core facilitator of electronic data interchanges yet require additional mechanisms in order to function. [Paper, 2007]*

The cited definition is one of several that are used most often. It is noteworthy that Web services understood in the spirit of the WSDL document are one of the many possible ways of realisation of the SOA paradigm. When one is to decide to produce a instantiation catering for his needs, he must include a number of mechanisms that should make this solution durable and resilient to the ever-changing environment. One of the most important is the service retrieval based on some standardised description.

Web services are an example of a highly specialised information good. There is a great need of highly specialised tools and mechanisms that empower users and organisations to make decisions where uncertainty is to be curbed to an acceptable level when dealing with such type of entities. This need is a constant element of all the economic endeavours of participants on any kind of a market. Web service as an information good is yet more complex that data on profitable investments or news of misfortunes that can lead to preemptive actions. Web service is an example of technology, thus its uses are more complicated and circle around the notion of its utility to its prospective adopters. Such complexity and role in economics merits a closer examination that is given in sections 1.1.2 and 2.5.

The research activities and enterprise effort provide a number of main solution groups that try to satisfy its users with robust Web service retrieval. The four initially established groups are:

- solutions using unmodified or slightly changed Universal Description Discovery and Integration (UDDI [Business, 2001]),
- Web portals such as XMethods or eSigma,
- various systems based on the classical Information Retrieval - where a WSDL document is treated as a set of terms,
- semantics-based solutions where additional description techniques are used.

[4] Wintergreen Research report
[5] Gartner Says SOA Is Evolving Beyond Its Traditional Roots

Table 1.1: Initial summary of Web service description and retrieval solutions

technology	ease of use	costs	precision	scalability	time
UDDI	◆	◆	◇	◭	◆
Web portals	◆	◭	◇	◇	◆
Syntax based solutions	◆	◆	◇	◆	◆
Semantics based solutions	◇	◇	◆	◇	◇

Cost is a shorthand for cost of Web service addition into a Web service repository. Time is a shorthand for execution time of a query using a given solution group.

The solutions are evaluated in terms of their suitability for an organisation willing to deploy an infrastructure implementing the SOA paradigm. The earliest attempt to provide an answer on how this available solution groups are copying was an analysis of their traits supported by the feedback obtained from active industry professionals.

The analysis carried by the author and reinforced by the obtained feedback, made it visible that one cannot fully address any solution without extending the set of traits. Among the important traits that had to be included was the notion of ease of use as perceived both by the end user and an organisation as a whole. What is more, another trait that could not be neglected was the cost of Web service incorporation into organisation's repository. Final trait that cannot go unaccounted for was the scalability, as industrial strength repositories cannot brake down under the increasing load of to be processed Web service descriptions [Anadiotis et al., 2009, Pierre et al., 2009, Stephens et al., 2011]. The summary is given in table 1.1. There are solutions that step beyond the boundaries drawn in this initial analysis and they are discussed in later chapters.

The presented summary served as a main motivation for building a proposition for a modern Web service description model. To present this model one has to state the main goals and the thesis of this dissertation.

1.2 Main goals of this dissertation

The dissertation aims at introduction of a novel model for Web service description and retrieval. This model has to redefine a number of strategies used in the most wide-spread solutions so that it better suits the needs of organisations deploying

their infrastructure based on the SOA paradigm. What is more, it closely tends to individual needs of various groups of users inhabiting the addressed organisations. To achieve it, the following main goals were defined:

- Preparation of a Web service description model that combines, rectifies and extends the available Web service description means with simultaneous cost control of the new description. The model addresses both the cost of description preparation and of its retrieval.
- Introduction of a set of mechanisms that work with the designed model and adhere to the requirements defined as a part of the presented model.
- Validation of research prototypes throughout experiments so that the robustness of the model and the necessary supporting mechanisms is truly measured and tested.

The above stated research goals are to be answered by investigating the following thesis: **The modern approach for Web service description and retrieval derived and rectified from the state of the art solutions shall increase quality of the retrieval process in comparison to the available means in concordance with the identified requirements of organizations implementing the Service Oriented Architecture paradigm.**

The main goals of the dissertation can be further detailed by stating a number of specific research goals in form of questions. Providing an answer to the below enlisted questions is intended to help in addressing the main research goals in deeper and more thorough manner. The most important specific research goals are:

- What are the most important requirements for a business users of Web service description and retrieval tools?
- How well the already presented solutions cater for the identified requirements?
- What elements of already available solutions should be used and what elements previously left unaddressed should be introduced into a solution covering the user requirements?
- What is an acceptable level of complexity for a business user in a Web service retrieval model?
- In what manner functionality should be attached to the Web service description?
- Which users' groups should be addressed in the Web service model?
- How does the postulated solution improve the choice agility inside the organization and what other benefits are there?
- What is the description methodology?

- How the solution should address concepts that do not exist in the actual descriptions?
- In what ways should user interact with the solution implementing the model?
- What is the efficiency of the solution and how is it to be measured?
- Is the solution cost effective and how so?
- How well does it scale and under what conditions?
- How to rank multiple results to a given user?

While gradually providing answers for the above questions, a complete model of a modern Web service description emerges. It covers all the most important aspects that were refined from the users' requirements and available technologies.

The functionality description is expressed in a new way that does not need complex description strategies as opposed to models designed and implemented in a spirit of semantic oriented technologies such as OWL-S and WSMO (described in greater detail in 2.3.2. Having prepared the functionality description structure, a method for preparing those adhering to this structure is given along with techniques for automation where it is achievable at a moderate cost. The model also addresses situations that appear when unknown terms are used in the Web service retrieval and when users query repositories foreign to them. This is handled by the introduction of Suborganisation Units (SU) that are represented by namespaces along with Local Context Anchoring that tries to provide a feasible answer leveraging a set of heuristics and data retrieval strategies on the available knowledge resources.

The gathered results are ranked thanks to the mechanism taking into account a variety of variables such as a user status, home Suborganisation Unit, his previous searches and the most popular answers that were classified as the most similar ones to his.

1.3 Methods of research

The methods of research in this thesis are inspired by three main research methodologies. This does not lead to inconsistencies due to the fact that each of the three frameworks delves in a separate tier of generality.

1.3.1 Concept-Knowledge Theory

The core of this work spins around the Concept-Knowledge Theory (further denoted as C-K Theory or just C-K).

As challenges standing before this thesis originate in the analysis of the state of the art approaches for Web service description and retrieval, one can emphasize that the covered material amounts to a specification (understood as in [Hatchuel and Weil, 2008]).

In addition, this specification is inherently bipolar due to the complexity of the problem of functionality description. The specification gathered from the literature review and inquiries among Information Technology and business practitioners dealing with Web services and other technologies conforming to the general manner of Web services operation (examples are given and explained in the later part of this work) made possible an initiation of work on mapping it to a design solution.

Following [Hatchuel and Weil, 2008], the design solution induces a number of previously unknown objects that could not be foreseen in the beginning of the design process. Yet, with an advancement of the process a new body of knowledge is generated that allows for a confirmation of the existence of the previously unknown objects. A definition of design presented in [Hatchuel and Weil, 2008]:

Design is a reasoning activity which starts with a concept (an undecidable proposition regarding existing knowledge) about a partially unknown object x and attempts to expand it into other concepts and/or new knowledge. Among the knowledge generated by this expansion, certain new propositions can be selected as new definitions (designs) of x and/or of new objects.

Initial research activities revolved around establishing a desired nature of a solution that should be acceptable and preferable by a specific group of the target users. There was a number of attributes that had to be gathered and confronted with the available body of knowledge. The final solution is the final step in the process of multiple bidirectional transitions between so-called C-space and K-space. The finished solution enriches knowledge resources. The overall flow of the transgressions between Concept-space and Knowledge-space is given in Figure 1.2.

1.3.2 Design Science in Information Science

The second tier of generality is domain of design science understood as in [Hevner et al., 2004]. It is a framework that allows for managing the inseparable nature of design science and behavioural science research.

This is more specialised approach that equips researcher in a number of guidelines that make it possible to produce a high quality output. As mentioned in [Ondrus and Pigneur, 2009], there is a general trend for close examination of the design product,

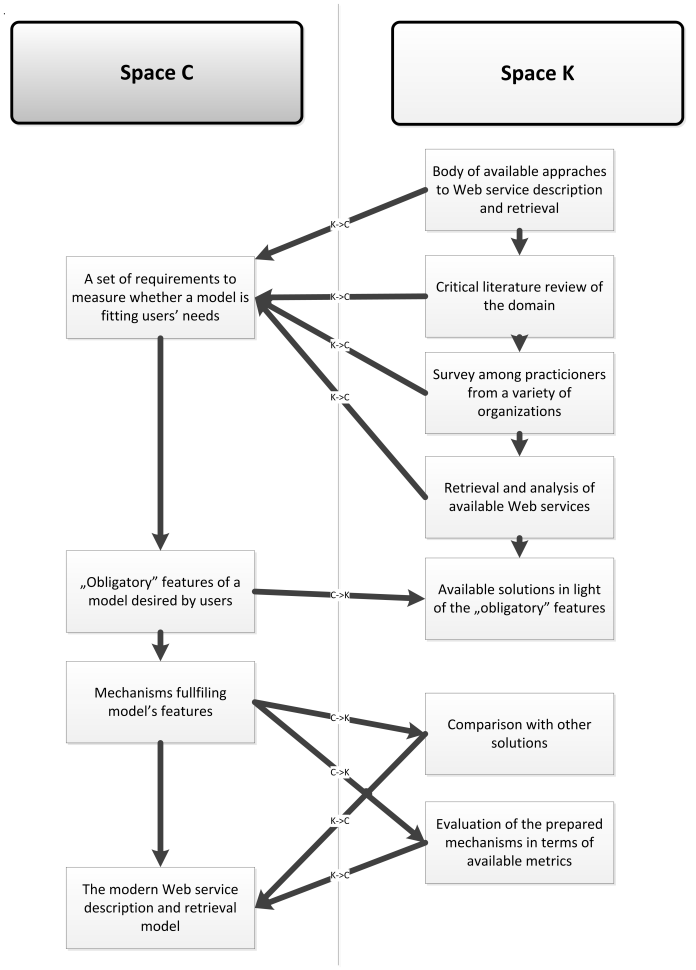

Figure 1.2: Flow between Concept and Knowledge spaces for the dissertation topic of interest

the design process, and the design environment in design science, thus leaving the questions of designed reasoning to be answered elsewhere.

The guidelines of Design Science Research are gathered in table 1.2.

The mentioned guidelines not only allow for quality assurance, but also help organize the research activities in a well ordered sequence of steps leading to the effective solution.

1.3.3 Additional research methods

The third tier is composed of the canonical tools of research such as abduction, deduction, induction and inference supported by results of experiments, data gathered from opinion panels participants and measures that allow for comparing obtained

Table 1.2: Guidelines for Design Science Research from [Hevner et al., 2004]

Guideline	Description
Guideline 1: Design as an Artifact	There is a number of artifacts being a result of work on this thesis. As mentioned earlier those are constructs, models and methods.
Guideline 2: Problem relevance	Functionality definition for Web service retrieval is an important problem in the domain of Service Oriented Architecture. As the available solutions lack important features needed to successfully operate considerable repositories in an effective manner where level of cost is managed, the proposed model is a possible solution. More, it breaks with a number of schemes considering the method of functionality description so it is once more accessible to every type of user.
Guideline 3: Design evaluation	The validation of the model and mechanisms was conducted. The key aspects of the desired solution were chosen as validation criteria. Due to the lack of resemblance of the model to the mainstream approaches a direct benchmark was not possible in all of the key aspects. Nevertheless, validation was prepared both in the form of as a qualitative and quantitative analysis depending on the feasibility of application in given test scenario.
Guideline 4: Research contributions	As mentioned the proposed model varies a lot in comparison to the mainstream approaches available in the domain. The key aspects were created as a plane of comparison that enables one to observe that there are no similar solutions. Thus, the postulated model is a novel approach unprecedented in the domain in its presented here form and scope.
Guideline 5: Research rigour	In order to adhere to this guideline, all research activities took into account available best practices and standards. Those were followed as far as their scope allowed for it. What is more, various regarded viable concepts and methods with established reputation were used.
Guideline 6: Design as a search process	All the research activities were driven by a constant review of already crafted artifacts. This review led to a number of improvements that helped to raise the quality level of the already available artifacts. Constant discussions among peers, presentation of artifacts and input from professionals served as driving force of the research.
Guideline 7: Communication of research	The artifacts being the result of the research are communicated in form of various publications. The publications demonstrate different artifacts being at different levels of advancement.

research results to competing approaches. The use of enumerated tools is in lockstep with the above-referenced guidelines and provides measures to secure research rigour and the validity of artifacts [Lakatos, 1978].

1.3.4 Summary of the most important research artifacts

Table 1.3: Summary of the most important artifacts obtained throughout research

Artifacts	Description
Constructs	In order to clarify the discussion in this dissertation a number of items had to be addressed. First of all, most important principles governing the domain of interest were given. More, terminology and vocabulary of the domain was presented along with the discussion on various concepts that had to be taken into account while developing models that should find it as an application ground. These were amassed in chapters 2 and 3.
Models	Using means provided in earlier chapters, chapter 4 introduces a general model for the modern Web service retrieval. It is introduced as a general overview. Further, the most important aspects of the model are regarded with reference to the domain.
Methods	Chapter 5 focuses on various features of the model that were addressed in a number of mechanisms. The mechanisms are presented in greater detail along with the demonstration of their functioning and steps necessary to obtain the desired level of results.
Instantiation	It is covered by chapter 6 devoted to the validation of the model along with its mechanisms. Both cover the key aspects concerning the Web service retrieval defined at the beginning of this dissertation.

Among various research artifacts described in this thesis the most important are those that present the highest level of added value in terms of novelty. The model of functional description is an example of such an artifact. While based on already established technologies thanks to the in-depth analysis of status quo of the domain it offers considerable extensions in terms of scope and features.

Methods supporting the functioning of the model are another set of artifacts produced by the research described in this thesis. The prototype being the proof of concept is also an important artifact as it allows for measuring the performance

of proposed model in terms of syntactic measures prepared for this task and overall user quality evaluation based on the opinion panels.

Of importance, are measures designed to capture efficiency of main artifacts and series of experiments that were conducted to answer important research questions concerning direction of the model's evolution. The general overview is available in table 1.3.

1.3.5 Summary of applied methods

The author found that organizing the methodology in three tier structure proved to be very effective in terms of research clarity leading to the specific organization of the whole research process. Harnessing the inherently complex process of constant referral between C-Space and K-space provided this work with an axis which at a certain point allowed for stating the satisfaction with the design.

In addition, guidelines provided by Design Science methodology coupled with tools of scientific method proved to be invaluable in the process of research quality assurance.

1.4 Dissertation organization

The dissertation is organised as follows:

- Chapter 1 – Introduction – Necessary elements on thesis, research goals, research questions and research methodology.
- Part I
  - Chapter 2 – Service Oriented Architecture – Introduction of the necessary background concepts and technologies vital for the Service Oriented Architecture and Web services and specialised Information Retrieval. The definition of the key aspects to be applied to Web service description and retrieval. The discussion on the importance of retrieval and description as key elements in optimization of the decision process.
  - Chapter 3 – Web service description – critical analysis and summary of available initiatives covering Web service description and retrieval according to the previously defined key aspects.
- Part II

  - Chapter 4 – Model – Formulation of the proposed model along with introduction of mechanisms supporting its realisation.
  - Chapter 5 – Mechanisms – Description and exemplification of means formulated in the model along with the important research leading to the presented state of affairs.
- Part III
  - Chapter 6 – Validation – A set of experiments aiming at capturing the effects of instantiating of model and its mechanisms in a number of scenarios that shall demonstrate its performance measured against the previously defined key aspects of Web service description and retrieval.
  - Chapter 7 – Conclusions – Summary of the most important research artifacts along with a discussion on its applicability in real world scenarios of Service Oriented Architecture enterprises.

1.5 Summary

Subsequent chapters broaden the topic of Web service, Service Oriented Architecture and available description solutions so that the proposed model has a full grounding in the author's domain of interest. A special attention is given to the functional aspects of a Web service, motivation on Service Oriented Architecture and use cases allowing for clear alignment of SOA with electronic marketplaces. As electronic marketplaces are directly connected with core interest of economics, which is choice and its criteria, a more in-depth discussion is given.

Part I

Economical and technical perspective of Web service description and retrieval - State of the art

Chapter 2

Service Architecture

2.1 Overview

This chapter concentrates on a presentation of the crucial concepts constituting a background and reference for this dissertation in terms of current state of the affairs in the Web service description, key concepts and technologies used in conjunction with Web services.

The full list of the most important topics addressed in this chapter is given:

- Service Oriented Architecture,
- Web services,
- Semantic Web services,
- specialised Information Retrieval,
- economics of choice and,
- the most important knowledge representation techniques.

As emphasized in the previous chapter, making the correct choice is of utmost importance for an organization functioning in a buoyant and a highly competitive environment. Works such as [Costa et al., 2009] emphasize this struggle for efficiency. Observing that currently emphasis is upon optimization of a infrastructure's usage in terms of electricity and carbon emission [Masanet et al., 2011], one might suppose that the next step will be a general optimization of costs of used software. With publications such as those [Sun et al., 2011] it is still more probable.

The structure of the chapter aims at a clear top-down oriented presentation of the above-enlisted ideas. The presentation begins with those being the most general, and it is accomplished by the presentation of the ones that specify various matters in a very detailed manner crucial for the understanding of this work.

All of the enlisted concepts are necessary in order to fully apprehend the potential of service driven enterprises.

2.2 Service Oriented Architecture

This section focuses on providing a concise yet thorough picture of SOA. It includes definitions, applications and supporting mechanisms. The most important role of this picture is to demonstrate the sheer scale of the influence that SOA, once implemented, has on an organisation. What is more, use of any kind of services demands a special attention from the policy makers and technology enablers.

2.2.1 Importance of flexibility

Service Oriented Architecture can be viewed as a trend among enterprises to prepare their infrastructure in such a manner that enables change and adaptation to the buoyant environment. One has to remember that service orientation is not only technology centered, it also addresses the philosophical underpinnings of the whole organisation model and its functioning as an entity rising to achieve some goal, at the same time being capable of changing in order to reach this goal in an efficient manner [Jensen, 1998].

2.2.2 Characteristics of Service Oriented Architecture

Service Oriented Architecture is perceived as an another step in the Information Age Evolution [Cuadrado et al., 2008]. After initial IT infrastructure based on mainframe, subsequently replaced by client-server solutions, SOA is based on loosely-coupled building elements hosted in a cloud computing environment [Mell and Grance, 2009], which quickly becomes an enterprise standard.

There is no commonly accepted definition of Service Oriented Architecture. Those present in the domain publications are very broad and not always fully overlapping. The common denominator for all the available definitions or descriptions is the presence of a service.

Therefore, any solution to be perceived as SOA is a collection of services. This collection is based on the following characteristics [Services and Architecture, 2001,Papazoglou and Heuvel, 2007]: loose-coupling, transparency of service location, independence on protocols. Of utmost importance, is the difference between a service

and a component or a subsystem. It lays in the independence of a single service from other services available in a given service pool.

Additionally, a service apart from encapsulating and abstracting some actual routines is to be seen as an actual element of a business workflow. All services are manifested by their description. It is necessary as it enables users to find them and place them in their current workflows to achieve desired results.

Benefits of SOA could be summarized as a synergy between two coexistent architectures of every organization, the business architecture and the IT infrastructure. When the goal of joining the two above mentioned elements is realised, SOA promises following benefits to an organization that should implement it (assembled based on: [Bhiri et al., 2009, Liu et al., 2009, Papageorgiou et al., 2010, Werth et al., 2006]):

- increased revenues,
- more adaptive business model,
- decreased costs,
- shorter business cycles,
- organization's integration,
- decreased levels of business risk.

A corollary from the above is that SOA is empowering organizations to achieve more with less effort in terms of various resources.

In case of SOA, the operating environment for services is the Enterprise Service Bus [Chappell, 2004]. It enables services to be coupled together, so that after a number of such operations a desired workflow is ready to process business tasks. Its additional capabilities make it suitable for monitoring and reporting of internal state of the whole organization's infrastructure. Additionally, it acts as a tool to restrict access to some services.

Traditionally, three types of interested parties are involved when SOA is considered (originating from [Services and Architecture, 2001]). Namely: a service provider, a service (broker) registry, a service requester.

The first and the third party are traditionally important from the organisation's point of view as they represent supply and demand for services. The middle element is nonetheless crucial in situations where the number of providers, requesters and services is considerable. As will become apparent in later parts of this work, the traditional SOA model should be viewed in a slightly different manner.

This different manner highlights the shift of focus to the service repository, as without one that can handle a lot of services in timely, cost-effective and scalable

way, there is low probability of achieving a durable success for any organisation implementing SOA.

The service registry is defined by [Papazoglou and Heuvel, 2007] as an intermediary that is interposed between service requesters and service providers. It maintains an index of the available service providers and it is capable of adding a value to its registry of application service providers by provision of additional information about their services (quality, terms of use, etc.).

None of these extensions seem to satisfy needs expressed by users[1] [Benson et al., 2006]. One can argue that constant increase in SOA's adoption proves that available methods are enough. Yet, one cannot prove whether introduction of more profiled solution would not boost SOA's adoption as well as coverage of its promised benefits.

This trail of thought, induced a lot of interest from the research field. First, as a place were Semantic Web services[2] could be employed to achieve greater flexibility and easier configuration, then as a tool for abstracting various business processes [Haller et al., 2005b, Bhiri et al., 2009, Mahmoud and Gomez, 2008].

Service Oriented Architecture was introduced to possibly interested parties as a set of practices and principles rather than fully formalized rules to be followed in order to be compliant. A key idea behind SOA is building IT infrastructure from entities that provide some functionality and can be repeatedly applied across an organization. These entities should fulfil some well defined action with an additional constraint of using no other entities for accomplishing this action[3]. More complex workflows can be built by composition of the entities already made available to a user.

2.3 Web service technologies

Web services are treated in this work not as a technical mechanism for communication across networks, but as entities that allow for capturing a type of contract carried out by underlining code so that a greater focus can be set on what is possible to achieve by identifying a functionality of each and every operation from any given Web service.

Therefore, technical description of Web service, protocols allowing for their operation and various extensions are kept to the absolute minimum.

[1] The ones given in the cited work focus mainly on: a lack of explicit data tying, a lack of a notion of dynamic service data and flawed search model. Discussion there is specifically on grid environment, yet it is perceived that those observation still hold in the SOA paradigm environment.

[2] Semantic Web services are Web services enhanced by specialised descriptions allowing for an automation of their discovery, composition and execution [Cabral et al., 2004]

[3] A user should not be entangled in any specific implementation details. He is to use the element as he sees fit, having an unrestricted freedom from artificial constraints (excluding security).

2.3.1 Web services - an overview

Web services were introduced to researchers and industry in 1999. From the beginning, this technology was heavily promoted both by research and industry [Alonso, 2003]. Web Services were presented as a tool that shall make it possible to homogenize IT infrastructure built with different systems crafted in a variety of programming languages often perceived as legacy ones. The key element to Web Service is a description of its attributes. The description was standardised in the Web Service Definition Language (the name was changed from Web Service Description Language in version 1.1 of specification [Christensen et al., 2001]).

Every WSDL document describes a service in detail, focusing on the following elements:

- available operations,
- used data types,
- method of access to operations,
- mapping among data types and messages used in operations.

A WSDL document is encoded with XML[4], thus every section is implemented as a node with a set of child nodes that further describe the parent. Web Service Description Language has 4 versions, starting with initial one premiering in September 2000. There ware two minor versions, WSDL 1.1 and 1.2 before WSDL 2.0 become World Wide Web Consortium's recommendation in 2007. Version 1.2 was renamed to 2.0 as it contained many significant changes in comparison to the previous one.

WSDL in all its versions allows a developer to store information on technical functionality of Web service's operations in documentation nodes of document. For Web services deployed via programming environments such as Microsoft Visual Studio [6] or Eclipse [7], these nodes are used in automatic generation of web pages presenting them to the interested parties. An examples obtained from the Internet is presented in Figure 2.1[8].

[4] *The Extensible Markup Language (XML) is a subset of Standard Generalized Markup Language*[5] *that is completely described in this document. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML.* - `http://www.w3.org/TR/REC-xml/`

[6] `http://www.microsoft.com/visualstudio/en-us`

[7] `http://www.eclipse.org/`

[8] Source: `https://demo.oqanalyst.com/OQAHS/Services/exportimportservice.asmx`

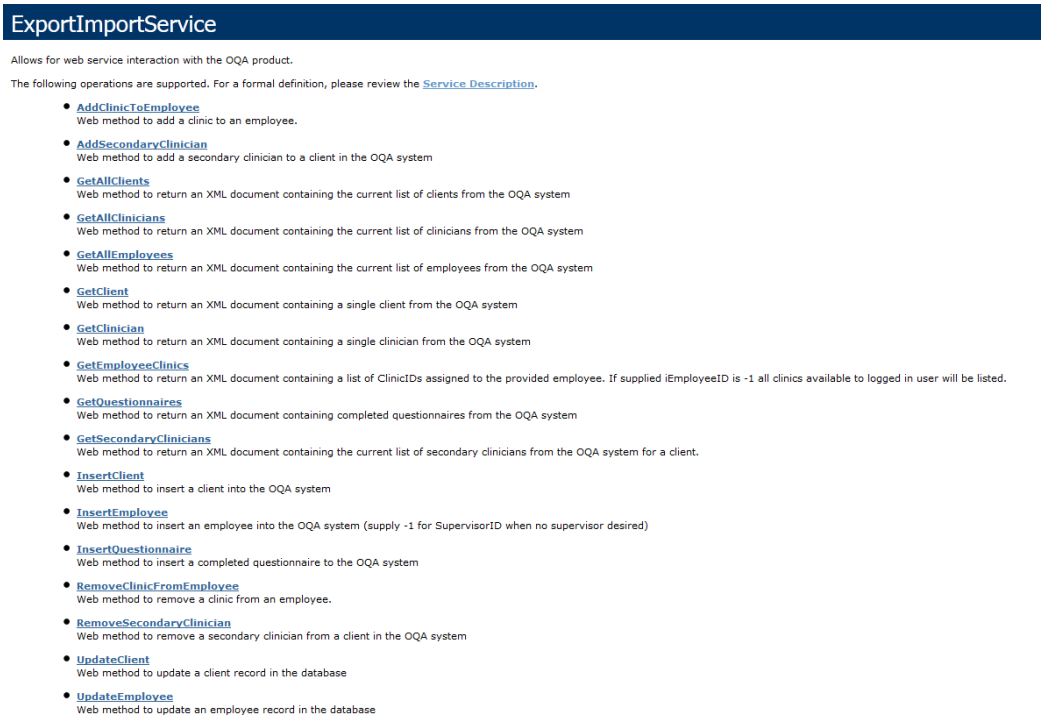

Figure 2.1: An example of Web service representation to a user willing to browse for details without a need of execution when deployed with ASMX technology

The form and quality of presentation depends on the effort invested by a developer during a Web service deployment and quality of documentation nodes inside the particular WSDL document.

While possibly scarce, presented data usually surpass those served as a part of the public UDDI [Business, 2001] (now mostly inoperable), excluding the data on a business function and the Web service publishing organization [Oasis, 2004]. WSDL became standard for defining Web services and all major software vendors employ it in their products [Nezval and Bartolo, 2011]. As standard documentation means of WSDL were found insufficient very soon, WSDL became a basis for a number of extensions. These extensions were mainly focused on realizing the idea of Semantic Web[9] that would allow computer programs to use Web service's functionality to a greater extent and in an automated manner. The detailed discussion on the most important extensions is given later.

[9]As given by the author of the term: "The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation." [Berners-Lee et al., 2001]

2.3.2 Semantic Web services

This section is focused on the presentation of the most important initiatives in the domain of Semantic Web services. Their importance is a result of the level of adoption in various areas and the availability of tools and projects using them as mechanisms enabling the projects' visions.

The selected semantic annotation technologies

As opposed to a particular solution this section is concerned with the most important semantic description technologies. Every technology described proposes a method for extending a WSDL document with additional data that allows for queries resulting in more precise results.

No matter the choice, there is always a cost of propagation of the necessary knowledge representation structures. Taking into account reviewed technologies, the most common knowledge representation is an ontology allowing for logic based reasoning [Guizzardi, 2006].

The reviewed technologies convey semantic data in a manner particular for them, and what is more, they differ in extent of logic expressions that can be used in query processing. The actual query processing is performed by a reasoner [Gardiner et al., 2006], ontology is, as mentioned, a knowledge representation structure[10].

To better visualize the idea of ontology building and reasoning an example is given in the final subsection. Apart from demonstration of the most important features, the issues emphasized in the initial research phase should be brought to the reader's attention in greater detail.

OWL-S

This initiative uses the Web Ontology Language (OWL)[11] to describe a so-called upper ontology used to describe Web services so that a precise Web service retrieval is possible. It is the oldest initiative recognized by the World Wide Web Consortium (W3C) and it is a direct descendant of technology first known as DAML Service Ontology (DAML-S) realised with DAML+OIL description language. This language was built so that it allowed for reasoning [Calvanese, 1996] on elements described with it.

[10] Depending on the complexity and the support for automated reasoning a very wide spectrum of entities can be deemed an ontology, from a simple catalogue to a set of general logical constraints [Smith and Welty, 2001].

[11] `http://www.w3.org/TR/owl-features/`

This ontology is organized in referral to the three major aspects of a Web service, the Profile, the Process Model and the the Grounding.

There are various variants of the technology, where differences stem from the level of coverage of various aspects of the Description Logics [Baader and Sattler, 2001] and resulting performance while processing queries. A more detailed discussion is given in [Klusch et al., 2009].

The underlying language of OWL-S was improved by introduction of features allowing for new methods of ontology representation [Hitzler et al., 2009]. This change improved the situation as it introduced a previously missing layer of flexibility.

Overall adoption of the technology was very high, a number of tools and projects thrived around it.

WSMO

The Web service Modelling Ontology [Roman et al., 2005] was introduced in 2005 with a goal to make the following tasks at least semi-automated: discovery, selection, composition, mediation, execution, monitoring.

The given list is not a complete catalogue of the targeted tasks, yet as one can see it is very broad and when reviewed it shall promise a comprehensive attendance to user needs. One must emphasize that, there are various versions of WSMO, that have the same origins as OWL-S variants.

In addition, one must clearly establish that the comprehensive attendance to user needs in terms of a Web service description and handling is achieved not only by WSMO but also with its Web Services Execution Environment (WSMX) [Haller et al., 2005a], which is the technology that leverages various aspects of Web services described with entities taken from WSMO based ontologies.

For a Web service user, the intricacies of the Web service description with WSMO concepts are very similar to the ones necessary in OWL-S. The important extension missing in OWL-S is a facility of the WSMO Goal. *...WSMO Goal is a specification of a resulting situation for which several plans and executions may exist and where there is a rigid structure containing a domain ontology, a mediator, a capability and an interface* [Toma et al., 2006]

This is very important as a Web service is given a purpose, the obvious deficiency of the WSMO goal solution is its opaqueness to a user as all the concepts and supporting elements that have some impact on the goal must be considered.

Additional insight that has to be given, is the price of comprehensible description. When a Web service is not only treated as an interface extended with data types

accepted, and access methods, but an element that can be positioned in a process, all the interactions, preconditions, postconditions, inputs and outputs (IOPEs) must be described in terms provided by the ontology describing the part of the world. All of the above is inducing an additional cost, and this cost can be unbounded as comprehensive description can grow without limits as more and more aspects are to be included to sustain its effectiveness. These remarks are addressed to both WSMO and OWL-S. The situation is far more complicated with WSMO as it covers far more aspects than OWL-S.

The issue of the growth of the complexity in time of a system that caters for users' needs is not new [Heylighen, 2004]. Addition of yet another layer provided by heavy weight description language with close to impenetrable syntax cannot help to reverse this trend or even balance it.

Hence, very early a set of alternatives was available advertising some different approaches. Nevertheless, the success of the technology was considerable, especially when it shall be measured with a number of paneuropean initiatives and projects. The available tools and supporting projects are not as rich and as popular as those surrounding OWL-S [Cardoso, 2007].

SWRL

A Semantic Web Rule Language introduced in [Horrocks et al., 2004] extends features of ontologies described in OWL by rules encoded in RuleML [Boley, 2006]. Motivation for this is the re-usability of once crafted rules, enhancing ontologies by introduction of a new feature boosting its expressiveness and facilitating the whole rule crafting process with well defined semantics of the RuleML.

The advertised expressiveness finds application while performing complicated queries which take into account a number of concepts which are in interdependence according to the ontology they originate from.

SWRL provides more power to a user with good grasp of the technology. Hence, it does not cut the complexity, therefore a cost of description and a learning curve remain high. SWRL's success is to be considered as moderate in comparison to OWL-S and WSMO, yet it is actively supported by a number of ongoing projects.

WSDL-S

The Web Service Semantics (WSDL-S) [Akkiraju et al., 2005] was first significant effort to decouple a WSDL document and its semantic extensions. The biggest ad-

vantage was its focus on facilitating work of those handling the description process. The technology allowed for describing WSDL documents with any viable technology such as OWL, WSML or UML. It therefore, was the first step to reuse already available descriptions.

The adoption of this technology was moderate especially in the light of introduction of SA-WSDL. Nevertheless, there are successful real life deployments based on WSDL-S [Herrmann et al., 2007].

SAWSDL

Semantic Annotations for WSDL and XML Schema [Kopecký et al., 2007] is a technology originating from the efforts of researchers clustered around WSM* family of technologies. It reached a level of World Wide Web Consortium Candidate Recommendation in 2007. Currently it is used as a vehicle for a semantic description in a number of research projects.

The idea motivating the SAWSDL's introduction is somewhat similar to the one behind WSDL-S: to decouple well known, recognized and covered WSDL documents from semantic annotations that might be overcomplicated as perceived by a user. In addition, this decoupling encourages to semantically annotate using any of the available technologies.

As an addition to the technology, a set of tools was made available to achieve the promised results. The most important are SAWSDL4J[12], WSMO Studio[13] and Radiant[14]. The two later are based on Eclipse, the first one is a Application Programming Interface for Java.

Adoption of this technology was far broader than WSDL-S, mainly thanks to support of WSM* community. It is a step towards making annotation easier and more accessible. Automatic mappings are possible only when a substantial effort is spent on preparation of rules that achieve it. Yet, one time cost of preparation might be an important incentive to invest in this technology. There is a number of projects using SAWSDL and its derivatives [Klusch and Kapahnke, 2008].

Semantic Web services - a discussion

Idea of addition of semantic extensions to WSDL documents is only slightly older that the Web service technology. In publication [McIlraith et al., 2001], authors outlined a

[12] `http://sawsdl4j.sourceforge.net/`
[13] `http://www.wsmostudio.org/`
[14] `http://lsdis.cs.uga.edu/projects/meteor-s/downloads/index.php?page=1`

mechanism of extending technical description with OIL and related ones [Van Harmelen and Horrocks, 2000]. The early results were very promising. Additional extensions made it possible for machines to reason on the purpose of processed annotated documents, what affected positively the value of the precision measure. What was more important for the authors, boost of the precision's value allowed for the introduction of automatic composition of Web services based on their functionality.

A good view on the whole domain is presented in [Maigre, 2010] where one can find results of a survey of tools for automatic service composition. The author of the referenced work established that there are 21 tools in total. Nevertheless, none of them meets all aspects that were chosen to test whether one is working with a tool that truly automates Web service composition. More interesting is the fact, that there are six distinct types of available service description base technologies:

- state chart [Fu et al., 2005],
- flow graph [Hecht and Ullman, 1974],
- Business Process Execution Language and derivatives [Jordan and Alves, 2007],
- OWL-S (successor of: DAML-S) [Antoniou and Van Harmelen, 2004],
- mashup [Alam et al., 2010],
- Web Service Modelling Language [Bruijn, 2005].

Every description method, save the mashup, imposes a certain cost on its user, that must be taken into consideration when deciding to employ it.

Disadvantages of Semantic Web services

A decision to not include traditional semantic annotation languages such as technologies operating on OWL-S [Antoniou and Van Harmelen, 2004], WSMO [Roman et al., 2005], WSMO-lite [Vitvar et al., 2008], SWRL [Horrocks et al., 2004], WSML [Bruijn, 2005], SWSF [Battle et al., 2005] pure RDF[15] [Needleman, 2001] or Microformats [Khare and Çelik, 2006], is a key feature that allows to store a large number of Web services and retrieve it in a robust manner.

This is due to the fact that each of the mentioned technologies except from Microformats is a fully fledge ontology system, which is designed to give answers based on an ability to infer facts. As is visible in conclusions of the literature review (given in chapter 3, this is one of the core issues that are decisive for a solution sharing

[15]RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. - `http://www.w3.org/RDF/`

the best traits of the evaluated solutions. Reasoning and the semantic description provide results, yet consume considerable amounts of time in order to present a viable answer.

As an ontology is a model of a world it has to convey a sufficient quantity of information on the world it models. There is an ever-occurring problem of the right scope of an ontology, of how many concepts are necessary to obtain a model that is sufficient for purposes of its users. More, there should be some framework for future extensions and changes which are inevitable throughout life of an organization [Peroni et al., 2008].

It takes time and effort to build a good ontology that can be used further in any of the fully-fledged semantic oriented solutions. A detailed description on performance of semantic annotation oriented solutions is given in 3.6.2.

2.3.3 Advantages of Web services to Service Oriented Architecture

First and foremost, Web services are well suited for integration of heterogeneous systems and interoperability among them [Pasley, 2005]. Integration and data interchange is possible both inside organizations, and breaching their boundaries in order to reach closer and farther stakeholders.

An additional benefit of Web services, is the ability to encapsulate some functionality. In similitude to a basic building block, a functionality manifested in Web services can be used in any business process that is in the need of it.

This encapsulation, when aided with a proper markup, allows for savings. Savings are possible from the two sources. First one allows for lack of unnecessary implementation of already present functionality. It opens new pools of functionality available from competing suppliers that should compete with each other with price and quality of their offerings. The second source of savings is possible as long as some reliable resource pools for processing power are sustained. Fierce competition of cloud computing systems promise plenty of resources at competitive prices along with high degree of reliability and data security.

Being able to encapsulate and potentially cut costs, another aspect of Web services that is important in SOA is the ability to monitor the use of every single Web service [Lund et al., 2007].

With this, new key performance indicators can be built showing which organizational divisions consume which services. Managerial decisions on which IT investments are the most important can be taken with greater conviction.

Another important aspect of Web services used in SOA is the ability to dynamically exchange Web services in workflows. Always when there is a new Web service that complies with interfaces and encapsulated functionality of the old one, it can be exchanged with no visible impact on workflows [Basu et al., 2008].

Aside from the already mentioned aspects, Web services can be offered to interested parties in order to maximize returns on investments.

2.4 Information Retrieval technologies and its impact on information asymmetry

This work references classic Information Retrieval techniques many a time. In order to legitimately do that, one must refer to a definition of the Information Retrieval. It is always entwined with a system in order to point to the automated nature of the whole domain.

One of the earliest available definitions is given by Lancaster: *An information retrieval system does not inform (i.e., change the knowledge of) the user on the subject of his inquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request* [Lancaster, 1979]

A one that is presented by Chowdhury, 35 years later does not alter its core message, but extends it by functional aspects: *An information retrieval system is designed to analyse, process and store sources of information and retrieve those that match a particular user's requirements. A bewildering range of techniques is now available to the information professional attempting to achieve this goal.* [Chowdhury, 2004]

Information Retrieval (referenced as IR) applies a number of techniques in order to satisfy user's requirements. These techniques are very diversified, yet the most important and the most recognized in the IR context is managing various types of data referring to an entity that is known as a document. IR techniques allow to index terms and other entities in any given document. Hence, they provide means for further retrieval of any given document by cross referencing a set of retrieved terms with documents they originate from.

Entities that cannot be treated as a stream of terms, in order to be retrievable must be described with metadata. Metadata is: *Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource. Metadata is often called data about data or information about information.* [met, 2004] A structure of any particular metadata entry is dependable only by the means and requirements of the IR system constructors.

The two most important means to benchmark a robustness of IR system are precision and recall. For more detailed description of the other measures one might consult one of the following sources [Manning et al., 2008, Lancaster, 1979, Baeza-Yates and Ribeiro-Neto, 1999].

Precision measures the number of relevant documents retrieved in the general result set. The relevancy is based on user's information need [Wilson, 2006]. Recall measures how many relevant terms were retrieved from the corpus of all the relevant documents in some actual result set.

2.4.1 Knowledge representation and its role in IR

In order to allow for a high level of effectiveness and manageable level of the cost of the whole description and retrieval process, an overview of a wide range of available resources that can be used as auxiliary tools, must be given. This overview concentrates on resources that are rich in data and easily accessible for any interested party. These resources cover various categorisation methods, available categorisation technologies, taxonomies, ontologies and data sources.

Knowledge and categorisation resources

As mentioned there is a number of resources that store knowledge. By many Wordnet [Miller and Fellbaum, 2007] is a good example of commonly available resource that can be used as a reference ontology for organizing data and enabling reasoning on what is a generalisation of a concept. Unfortunately, Wordnet cannot be used for this task in its natural form as it is designed to be sense complete not a resource that would allow for robust disambiguation. Without additional extensions and reworking of it structure it might introduce more ambiguity than clarity. Nevertheless, due to the fact that it stores well over 155287 thousand unique terms (as on 10.07.2012) it is well equipped to demonstrate closest neighbourhood of a markup term.

Another important resource is the CYC project [16]. This resource is also difficult to completely apply in the pursue of satisfactory solution as it imposes on its user steep learning curve which is not facilitated with abundance of references and documentation. Even so, it contains, in its full commercial version, over 300000 concepts and offer 3 million facts relating to stored concepts (as on 13.06.2012).

Yet another important resource is Suggested Upper Merged Ontology (SUMO) [Niles and Pease,] [17] and its domain ontologies. Similarly to CYC, apart from terms SUMO and support ontologies consist not only of terms but also of axioms that allow for inference. The overall number of concepts in all SUMO group ontologies is only 20000 and additional 70000 axioms. It is comparable with the number of concepts available in one of the CYC subprojects - OpenCYC. The relative low number of concepts is organized in smaller ontologies, which results in a categorization edge to be easily used when presenting marker terms in context. As another advantage, SUMO group terms were mapped manually to Wordnet synsets what boosts the overall potential of this ontology group.

The decision to use one of the mentioned knowledge representation resources deems one to agree on relation building strategy employed by its builders. As mentioned earlier this shall not be the best solution in highly specialised organisations using language suiting its particular needs. The easiest demonstration is trying to query available solutions on terms such as GalaxyTab or MeeGo. As these are respectively a phone model and an operation system one cannot say that they are irrelevant. The point is to underline the enormous task of importing hundreds of thousands of terms into ontologies and organizing them in flexible manner.

Anyone interested in knowledge representation shall review the Wikipedia[18] as one of the most important effort that provides a free description and framework for categorisation of multiple domains in multiple languages. Thanks to the efforts of researchers [Auer et al., 2007] knowledge stored in the Wikipedia and accessible mainly by people, got structured and interfaces were provided so that it can be queried in a manner similar to a database. The effort was named as DBpedia[19].

The sheer size and scope of the Wikipedia allows to map a huge number of concepts originating from a great number of domains. Its constant growth and refinement of its contents positions it as the most important resource for knowledge representation.

[16] http://www.cyc.com/
[17] http://www.ontologyportal.org/
[18] http://en.wikipedia.org/wiki/Main_Page
[19] http://dbpedia.org/About

Nevertheless, one has to be aware that there are claims of inaccuracies or blatant errors in this body of knowledge [Denning et al., 2005, Korsgaard and Jensen, 2009, Messner and South, 2011]. Many of the problems with the content are not solely connected to the Wikipedia itself and occur in other resources commonly accepted as trustworthy [Giles, 2005], yet some are typically a result of the nature of this resource [HOLMAN RECTOR, 2008]. When one is to compare advantages and disadvantages of this resource, one has to admit that the problems with inaccuracy or errors can be accepted due to the enormous advantage of scope and speed on inclusion of new concepts. In addition there are mechanisms that shall establish a level o trust to any resource that shall provide data on demand.

Apart from the previously given resources, the Internet enables to use a wide variety of thesauri, dictionaries and taxonomies. These are very important from the perspective of disambiguation and retrieval of similar or alternative terms.

Categorisation efforts such as the Open Directory[20], large software repositories such as Sourceforge [21] and socially driven initiatives such as Freebase[22] and Quora[23] provide an important background on variety of topics an serve as a source of relations that can be used in functionality retrieval.

2.4.2 A brief overview of the most important IR techniques

To begin with, one must one more time describe the situation with which the modern IR is dealing. Every IR system works with a body of documents which differ in particular details. There must by a stage where analysis of these resources is accomplished. Its main objective is to prepare structures that are used to resolve queries.

The most known structure of this type is a inverted index which provides data on which term occurs in which documents. More robust solutions rely heavily on this approach while abstracting from the original way of representation and storage [Baeza-Yates and Ribeiro-Neto, 1999]. Such extensions allow for construction of indices that can be distributed among a number of phisical/virtual machines. What is more, the great effort is invested into building indices that dynamically react to the changes in the body of indexed documents [Manning et al., 2008]. Moreover, dynamic and distributed indices have to be further optimized by a number of techniques so

[20] `http://www.dmoz.org`
[21] `http://sourceforge.net/`
[22] `http://www.freebase.com/`
[23] `http://www.quora.com/`

that it is feasible to store the constantly growing in size indices in memory of the machines delegated to hosting them [Anh and Moffat, 2005].

Apart from various optimizations and enhancements on the structure responsible for indexing the body of documents being the interest of a given IR system, there are additional techniques relevant to the model of IR system. The most important of those is the preprocessing of the data that has to be indexed. The most important stages that are enumerated as a standard ones are [Manning et al., 2008]:

- tokenization,
- removal of stop terms/words/phrases,
- normalization,
- stemming and lemmatization.

Despite very advanced stage of these techniques, there is a constant research in these fields mainly due, but not exclusively, thanks to efforts of various non-english initiatives [Ingason et al., 2008, Korenius et al., 2004, Roth et al., 2008].

Yet another important element of an IR system is a capability to score and rank results based on their perceived value to a end user. The approaches to this notion vary, yet all IR systems try to maximise the perceived usability in this domain. The most important approach to the matter is present in the PageRank [Page et al., 1999]. It is to this moment the most known and the most influential idea that is mimicked by all the major players in the search industry. Obviously, it is not always applicable due to the nature of resources to be indexed. Without ranking and scoring, an end user has to manually review resources in order to decide which is the best for him. Ranking and scoring of documents provide a suggestion based on some heuristics of content analysis that are tuned to reflect the possible similarity of user provided query and content of a resource.

The last important aspect of an IR system is its broadly defined interface. This is a layer that is responsible for Human-Computer Interaction. It helps to deliver user query by provision of a number of mechanism, such as input suggestions or recently issued search phrases. Here it presents the computed results in a manner that suits user best. Support visualisation of results, their clustering and other form of aggregation are also handled through the interface [24].

[24] An interesting example of a IR system with a set of discussed traits is available here:`http://search.carrot2.org/stable/search`

2.5 Economical aspects of Web services and SOA

The section covers most important aspects of Web service description and retrieval in light of the economics. The most important aspect of these, is the facilitation of the decision making when it comes to choosing the best suited services to a given task. Redefined service description enables various groups of users to back their decisions with broader range of important data. Thus, good description should minimize the risk of suboptimal choice.

All of the above is set into perspective of Service Oriented Architecture, and the necessary reorientation of the organization functioning to adapt to this paradigm.

2.5.1 Economics background of Web service description

In order to achieve the key goals of this work, one must address a number of issues that enable the decision support optimization [Geoffrion and Maturana, 1995]. In order to achieve a desired state of the postulated model, a set of supporting models and mechanisms is presented in this work. In order to make the presented models work there has to be a change in philosophy of defining and accessing knowledge on functionality. This change in philosophy touches a number of issues concerning organization's operations on a daily basis. What is more important, it concerns the way organization's members think of the artifacts produced in order to achieve business objectives.

The mentioned change of philosophy is noticeable, due to the role of artifacts, which convey the functionality that is a cornerstone of Service Oriented Architecture ([Erl, 2005]). The benefits of SOA are enumerated and discussed earlier. The important issue is that the detail level of building blocks, which were decided to deliver the ability to reach business objectives, is different from the one that is associated with particular algorithms and data structures encoded in some programming language. These building blocks might be referred to with a variety of names. It is a result of lax definition of the building blocks, yet an overwhelming majority of interested in SOA had decided to reference them by a term of a service.

The services are different and the difference is emphasised, because they abstract more than code. They abstract pure functionality. Their interfaces are a contract that states what can be done but does not state the means of accomplishment. They were designed to serve as black boxes that encapsulate all the details of execution. All these

is a value already, yet the real benefit is the possibility to account of functionality available to an organisation[25].

It is true that one can browse code repositories, accompanied or enriched by documentation. It is true that in every organisation, there are dedicated professionals that maintain and develop subsystems enabling the organization to function.

Nevertheless, there seem to be no good idea for a service functionality description (a broad review of available options is given in the subsequent chapter). By a good idea, or in this context one that addresses the Key Requirement Aspects 2.6, one is to understand a solution that presents functionality in terms understandable for wide range of users. This wide range begins with regular business users that have to establish whether given functionality is available in an organisation, through service developers that need more information on service capabilities or its performance, and finishing with executive users that are interested in additional traits associated with particular services and concerning terms and agreements bonded with them.

The functionality of services, apart from actual implementation and design decisions, can be viewed as an economic model. Information on resources, their availability, their quantity and quality is the key to certain actions that aggregated can be viewed as a market where supply meets demand. The novelty of the situation here is the weakened role of price as the most important factor in all business dealings. One may argue that there are other important qualities such as guarantees, support or availability of skilled professionals to tackle the load of work with some solution. Yet, in general this are all price related issues, as they can be mapped on particular numbers representing monetary value to the decision maker. All of the mentioned aspects can be weighted by a cost of replacement when the guarantee expires or a cost of support when it was not bought and an event occurs that must be managed with its help [Heal, 1999]. The functionality itself is obviously a result of some business objective, set so that a profit could be made.

Nevertheless, demand for some particular type of functionality cannot be expressed as monetary value. A method must be present that enables to describe desired functionality or present the inventory of the ones already delivered by systems governed by an organization.

Hence, so much emphasis is put on this topic, many a time not directly expressed and overshadowed by technical goals of automated application building or resource indexing. Here, a model and methods that make it possible to implement are presented.

[25]More thorough technical description is given in section 2.3.1

Information on service resources is one of the possible sources of competitive advantage ([Shin, 1999]). Information in general and information on functionality in particular is a crucial element of market. As in [Bakos, 1998] the market has three core functions:

- matching buyers and sellers,
- facilitation of transactions,
- provision of institutional infrastructure.

The functionality description is a domain that is interwoven with all of the enumerated functions. A desirable description shall easily present features of a service, make it comparable to other services, make the logistics of a service usage swift and make it possible to monitor behaviour of a service throughout requests its lifetime.

As more and more of everyday business operations is handled with computer systems the eminence of applications handling them grows. The IT systems are quick to grow when successful and usually the growth bereaves them from agility and ease of use as with new functionality, new layers of complexity are added ([Merali, 2006]). To remedy that, a number of solutions is applied ranging from using dedicated professionals to maintain systems for suitable remuneration, to reimplementation of system functionality as a completely new entity.

One shall imagine, that the status quo is, to say the least, inconvenient for any organisation and there must be some course of action that could stop or restraint the above-mentioned situation. Transparency and openness in construction of tools that shall serve the organization's community may serve this goal well.

One has to remember Conway's Law [Conway, 1968] in conjunction with the postulate of changing philosophy of functionality description.

> Any organization that designs a system will produce a design whose structure is a copy of the organization's communications structure.

This is a very important observation as it highlights the fact that the solution is dependent on an organization and its final form will resemble it closely. Change is acceptable but it should not be introduced in a revolutionary manner. It is envisioned that it shall adhere to the culture of the organization and facilitate basic functions not introduce obstacles.

2.5.2 Economic impact of SOA on organization's functioning

A decision on implementing an organization-wide Service Oriented Architecture is accompanied with considerable effort in terms of time and financial expenses [Choi

et al., 2010]. There are no ready solutions for organizations that span multiple continents and their staff accounts to tens of thousands.

It is a result of uniqueness of every organization, the manner which it handles its own products and how it manages workflows. Therefore, all that can be advised to small or medium entities is unapplicable besides the general outlines and goals to be reached while implementing the SOA paradigm.

To accomplish implementation of SOA, an organisation is in need of definition of many new behaviours concerning [Papazoglou et al., 2007]:

- registry of available services storing data on means of accessing service and its state,
- service performance monitoring,
- data on service consumption across organization,
- details on which service has what contracts attached to it,
- details on responsibility for every service.

In essence SOA implementation is more than changing the IT infrastructure components, as it exerts considerable changes on organization's functioning. The scope of this changes is dependent on the profile of organisation, the closer it is to IT technologies the bigger the exerted impact. SOA , as was and will be discussed, has many to offer in terms of revenue. Yet this revenue might only partially come from the organization's core business. Leveraging SOA means that every Suborganizational Unit has access to data of other Suborganizational Units. Further, there is no obstacle except motives and possibilities, to encourage collaboration of external entities to produce and supply vital functionality to the benefit of an organisation. What is most important, the additional functionality created by a new platform of interacting services can encourage an introduction of new services which were not envisioned in the initial plan. Thus, SOA is the enabler of access to previously untapped resources.

The greatest examples of this philosophy are beyond SOA itself. Microsoft has become dominating force on the personal computer market thanks to allowing access to the infrastructure with which it is building its most successful products. The difference laying in the fact that Microsoft allows access to libraries and APIs is irrelevant at this point. It shares its resources to gain potential on the market due to the fact that making its platform successful, it makes successful its products.

The similar scenario was implemented in Apple with the advent of AppStore and huge success of the software available there, written with the tools provided by the company. It is not know whether Apple portable products would be that successful

without AppStore, yet the fate of other solutions may allow for speculating that this was vital choice.

The most eminent example of successful transition from pre-SOA architecture to a SOA one is the Amazon. Once an organization dedicated to online retail, now a leader in dedicated services aimed at cloud computing. Thanks to internal changes the technology enabling retail business was transformed into an aggregate of services that could be easily applicable outside the organisation.

In manner of years, Amazon opened for itself brand new market that made it possible to widen and diversify its revenue streams. More, it encouraged other companies to follow its example and open access to their infrastructure with services monetizing something that by many was thought to be irrecoverable investment[26].

It has not been decided, whether there is a single best method to implementing SOA in terms of technology supporting it. An organisation can achieve it be employing any of available ones such as HTTP based (Web sevices implemented with SOAP, REST services), Corba, xml-rpc or any other even custom designed one. Nevertheless, all of them need a unified solution that can describe their functionality in a way that shall allow for grasping the intent of its constructors. This requirement is of utmost importance, as even the simplest documentation must provide the goal of every described artifact in order to be a useful resource.

Services being the building blocks of SOA hide a lot of dependencies that software handling them must take into account. This is tremendous advantage, as when a team wants something done in its workflow it has just to invoke a service and the results are returned. Obviously, the complexity is not gone, yet it is hidden from those in charge of fulfilling business objectives who can achieve them with greater flexibility.

2.6 New economically driven requirements for the Web service description and retrieval

The technologies and trends surveyed along with the preliminary research work, aimed at answering whether the state-of-the-art solutions do meet the requirements of their users, it was observed that a new set of indicators must be used to provide a correct answer. The correctness of answer should reflect the amount of change that Service Oriented Architecture and its implementations undergone over the course of the last

[26]Source `http://preibusch.de/documents/PreibuschS_FleckensteinM_Amazon.pdf`

decade. Moreover, it should include various factors omitted previously, due to its not technical nature, which are of importance to the users.

The actual list was compiled as a result of the research on the four groups of solutions delivering Web service retrieval and description capabilities 1.1. The following requirements were obtained thanks to experts' feedback while discussing the available options (covered in section 1.1.3):

- ease of use,
- description cost,
- precision,
- scalability,
- execution time.

To better asses the potential of available solutions and prepare a scaffolding for a comprehensive evaluation that leads to quantifiable results, a list of redefined requirements based on the initial ones was prepared by the author of the dissertation:

- **effectiveness** - a standard measure of a precision and a recall is used when a solution allows for it, if not applicable, effectiveness is perceived as the ability to cater for a need expressed by a user in a format provided by the solution. Solutions based on fully-fledged semantic annotation technologies always fetch items described by users when available, if not, an inference mechanisms tries to substitute with artifacts most similar. Therefore, one has a perfect precision dependent on design decisions of knowledge representation structures;
- **cost** - an effort that should be spent for a proper description with an envisioned technology. In addition the cost covers also time spent on learning necessary description techniques and a prognosis on a time-wise performance of analyzed solution. This blend of properties penalizes solutions that require a lot of effort in preparation for production and use. In addition, excessive time consumed of query matching phase is also reflected negatively;
- **scalability** - how soon and to what degree a performance should drop along with an increase in a number of handled descriptions. This measure should promote solutions that capable of handling millions of descriptions simultaneously;
- **scope** - any important additions to a baseline of a functionality description offered by a WSDL document in terms of an identification of vital, not previously addressed areas of importance to a user. Any extension of description past functional description of Web service operations such as business key performance

measures [Marr et al., 2004] and different than the developer's perspective on Web services is reflected as an increase in value of this measure;

- **purpose statement** - whether reviewed initiative allows for stating the purpose of an entity. This is understood as a possibility to express a goal of an artifact in question so that it is clear what it does to any interested user.

As one can easily see, the overlapping qualities were merged (as in case of cost, ease of use and execution time), precision was generalised into effectiveness as it now covers more than a standard IR measure. In addition purpose statement and scope were added to clearly demonstrate whether given solution is capable of bringing up the level of satisfaction while being used in any organisation, thanks to more features and clear functional definition.

Equipped with this set of refined requirements one can attempt a critical evaluation of available solutions described in the domain literature.

Chapter 3

Existing approaches to Web services description and discovery and their evaluation using the economically driven aspects

3.1 Introduction

As stated before, the Web service description is a very important subject for any fully-fledged initiative that wants to leverage all possible advantages envisioned in the domain literature (given in section 2.2.2). A description in question has to address demands set forth by various stakeholders. A particular description strategy is usually connected to a set of mechanisms allowing for a discovery of entities described according to this particular strategy. Therefore, each description effort is related along with the proposed discovery mechanisms, to provide the fullest possible view.

As most often, a success of Web services is connected with the Service Oriented Architecture [Haase and Nagl, 2008], a natural thing is that the Web service description has to reflect business users' needs. Many a time, this is not aligned with solutions proposed by researchers due to a mismatch in a level of the quality of retrieval process and a cost of achieving it. There is a considerable number of solutions that try to revamp the Web service description so that it is more useful in various envisioned applications.

These solutions might be divided into:

- those that try to maximize functionality of already available standards - covered in the WSDL based retrieval section,
- those that reinvent Web service description - covered in a separate named section,
- those that present an amalgamated approach - covered in a hybrid solutions section.

This classification, while coarse grained, is a good reference for further discussions on the solutions available in the literature, as it encapsulates the most important available trends.

When one would like to make himself familiar with a fine-grained taxonomy of the Web service description, he should refer to [D'Mello and Ananthanarayana, 2010]. This is an important work as it analyzes and enumerates approaches developed and presented over the last decade. It is not complete as it does not explicitly address technologies such as OWL-S ([Martin et al., 2007]) and WSMO ([Roman et al., 2005]), which are the core artifacts of the most important trend prevalent in the last decade, which is a semantic annotation started in the McIlraith's work [McIlraith et al., 2001].

As discussed earlier, both technologies offer a lot in terms of enhancements and additions to WSDL documents. Nevertheless, it is believed that they fail to provide a good ratio [Kungas and Dumas, 2009] of an effort spent on a description to its effectiveness (measured in terms of precision and recall). Further analysis was given in section 2.3.2.

The approaches that include Quality of Service parameters (defined for Web services in [Conti et al., 2002] and in broader view in [Hutchison et al., 2001]), those that present the ability to link the Web service description with Service Level Agreements, and those that implement a method to deliver multiple views of the same Web service for different interested parties, are of special interest to this analysis, due to a higher impact on their applicability in the Service Oriented Architecture environment.

It has to be noted that, all remarks towards one of the enumerated criteria take into account a corpus of Web services gathered as a part of research for this dissertation. The corpus covers over 50.000 WSDL documents that were retrieved from the Internet and analyzed to obtain a complete picture of a structure and peculiarities of the real world Web services (the research was started in 2008 and was continued to 2011). Especially, a number of offered operations and all the available documentation was closely examined as these two characteristics are of a tremendous importance to anyone designing a model that should stand a chance to any applicability whatsoever.

The findings from the literature review are summarized at the end of this section along with the derived conclusions that serve as a cornerstone of a postulated model. These conclusions, reinforced with data originating from the discussions with business practitioners, allowed for defining a model that is not detached from the real-life objectives.

The solutions could be compared to a model of Web service description and retrieval depicted in Figure 3.1. It contains a number of components, which implemented could provide more thorough coverage for the Key Requirement Aspects. The components were developed as a summary for KRA and additional issues signalled in two first chapters.

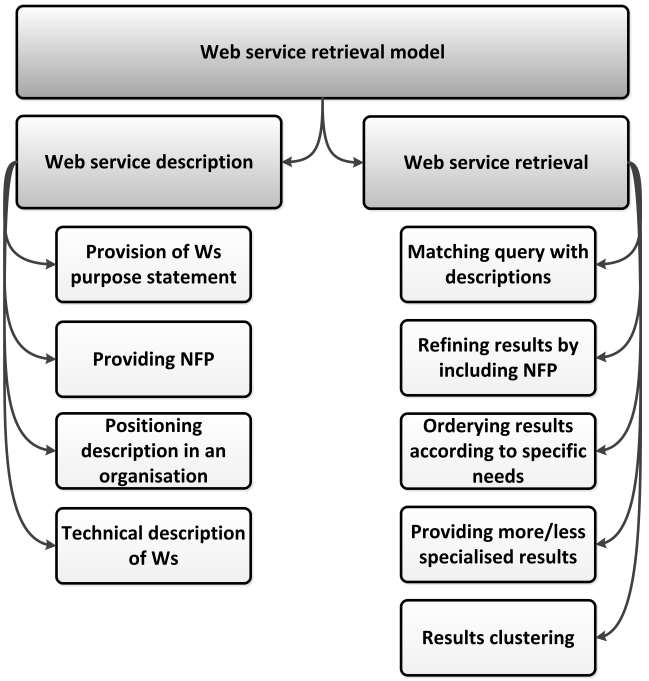

Figure 3.1: The model of Web service description and retrieval

The subsequent sections cover the three main groups of description efforts.

3.2 A WSDL based retrieval

This section focuses on the solutions that do not reach far beyond a transformation or a refinement of the base WSDL documents. As a description is seldom a main topic of the analyzed solutions, a discussion is often enriched with a broader perspective on a whole retrieval process, which is of a great importance to the goals of this work.

The key characteristic of each described initiative is its strong connection with a WSDL document that represents a Web service to the outer world. Nevertheless, some of the solutions include special techniques that aim at detachment of actual terms used within a query with those used in a WSDL document.

The solutions chosen originated from fairly large corpus of works devoted to the topic. From over 30, 12 were chosen to be reviewed more closely and assessed making use of the Key Requirement Aspects as given in 2.6. Those, being the most crucial to the discussion, are briefly analysed here, wherein the rest is only addressed in the table summarising the whole section. Similar actions are taken in the following sections.

3.2.1 A critical review of the chosen solutions

A number of ways was devised to extend usability of WSDL documents. One of the most important examples is an extraction of keyword data out of the WSDL documents and its transformation in order to achieve better results in retrieval. As an example, one can study [Zhou et al., 2008], where the authors decided to narrow possible computations to the most significant parts of a description. Of interest to the reader is how a borderline for relevance measure is created. The authors decided to focus on a frequency of mutual word relations. To some point these resembles an extraction of hub vertices from a graph [Blondel et al., 2004]. The authors decided then to overcome an issue of a keyword mismatch in a user request and a service description by extending their mechanism with an ontology [Sowa, 2000] that enables them to process keywords as concepts from the used ontology. This is promising, yet the authors do not provide an evaluation and experiments on the service corpora. Without that, one cannot speculate on efficiency and a description cost especially that presented method uses OWL-DL.

The need for grasping a Web service's purpose led to a work of [Wang et al., 2010]. Authors decided to enrich scant information available in a WSDL document by incorporating plausible external data. This approach led earlier to good results with photos and pictures obtained from the Internet [Lew et al., 2006]. A Web service neighborhood proved to contain enough data to improve retrieval results based on the classic Information Retrieval methods. One has to express that even the visible improvement cannot act as a remedy to the basic challenge of a definition of a Web service's purpose.

The work given in [Srivastava et al., 2007], presents Web services from the asset perspective. The discussion highlights the complexity of a Web service and proposes to observe it from a variety of angles implemented as a set of views. These views allow noticing the way an asset is stored, in what relations it is with other assets, a history of its changes, a catalogue of its permutations, information on usage across organization, standards it implements and finally, its purpose represented by a solution view that references its use in an application, thus, defining its function. These views, according to the authors allow for remedying some of the problems of a Web service retrieval with a satisfactory level of precision. What is more, the authors present experiment results that seem to be promising and where views are implemented without the use of semantic description languages.

The scalability challenge drives initiatives such as [Ma et al., 2008]. This work comes from a traditional approach to a Web service retrieval as it works with descriptions based on terms found in WSDL documents. It differs from other approaches as it addresses a scalability challenge from the very beginning of a retrieval process. The authors introduced a partition of overall body of Web services into separate clusters. This is an example of a divide and conquer approach. A query is matched across all available clusters and the best match allows for second phase of retrieval process. It uses technique of a Singular Value Decomposition [Klema and Laub, 1980] to index all Web services from the chosen cluster. After that, a regular retrieval is performed. As mentioned, this work does not surpass available solutions due to employment of generally known techniques, yet it is important as scalability issues are neglected in most of the available works. The authors performed experiments on a test corpus of Web services that consisted of 240 Web services. One cannot present experimental results based on this number of Web services and demonstrate great trust in it, as the number of Web services should be much larger. Application of the envisioned technique on a mock corpus of general documents does not change this corollary as the other corpus was roughly four times more numerous.

The trend to facilitate adding semantic data is visible in many published works. This promotes arguments of high cost of semantic extensions and need of reducing them in addition to the automation of whole process. An important example of already presented, yet enhanced techniques is [Espinoza and Mena, 2007]. As in other referenced works, the authors decided to retrieve all possible data from the Web service description and then use it as semantic descriptors. A novelty stems from observation that extracted terms being transformed into Semantic keywords may be ambiguous. The authors gave a method for disambiguation that uses a set

of domain ontologies. This allows making an informed decision on what exact sense should be connected with a keyword. In addition, the authors believe that a user assistance during the whole described process is necessary due to various problems with a knowledge representation. In general, the given solution is an attempt to further automate extension of the Web service description with elements allowing for a more precise retrieval. The authors are conscious of challenges ahead of any attempts of further automation. Nevertheless, they do not address the issue of obtaining and handling the knowledge base expressed as various ontologies. There are no reports on performance of their solution, what would be a great advantage to anyone that would like to follow their line of argumentation.

3.2.2 Summary

Table 3.1: Summary of the WSDL based retrieval solutions.

	criteria:				
	effectiveness	cost	scalability	scope	purpose
solutions:					
[Zhou et al., 2008]	◑	◑	◇	◇	◇
[Bruno et al., 2005]	◆	◑	◆	◇	◇
[Wang et al., 2010]	◑	◑	◑	◇	◇
[Srivastava et al., 2007]	◑	◇	◑	◆	◇
[Qin et al., 2010]	◇	◆	◆	◇	◇
[Ma et al., 2008]	◑	◆	◆	◇	◇
[Liang et al., 2009]	◑	◆	◑	◇	◇
[Espinoza and Mena, 2007]	◑	◑	◇	◇	◇
[Gao et al., 2009]	◑	◆	◑	◇	◇
[Khdour and Fasli, 2010]	◑	◇	◑	◇	◇
[Karimpour and Taghiyareh, 2009]	◑	◆	◑	◇	◇
[Iqbal et al., 2008]	◑	◇	◇	◆	◆

Solutions are evaluated towards the fulfillment of the Key Requirement Aspects with three marks. A black marker stands for a good level of support for given aspect, a half black marker stands for partial support, and an empty marker stands for lack of support.

Data presented in table 3.1 aims to give an overview on available solutions that base their functioning on an almost raw WSDL document processing. It is easy to

notice that effectiveness is low for most of the reviewed solutions. This is obvious as it is not possible to discover a true purpose of a Web service based only on terms that can be retrieved from a WSDL document.

There is no shared platform that supports building of Web service descriptions that use common terminology for common operations or which can provide a relation detection with some credibility. There are attempts to overcome this, yet they are insufficient and technologies proposed to achieve it, either cannot deliver the promised change or need some human intervention.

The above, directly influences cost level of the presented solutions. A cost is understood as an extra effort spent on the external description and time spent on applying it to the Web service repository elements and learning how to use it. Thus, one can say that the cost is rather low, as in most of the cases there is no extra description and no extra effort is needed to use most of the solutions.

Few of the described solutions were prepared as solutions that could handle many Web services at once. Yet, due to the fact that the level of additional complexity is rather low and well tested and widely used techniques were applied, one can assume that majority of the reviewed solutions is scalable. Solutions that introduce clustering methods are the best examples of design with scalability in mind. They are prepared to handle a high number of Web services due to the use of cluster representatives that visibly cut down on the amount of comparisons needed to present a possible match for a query.

Almost every solution from this group fails at addressing the two very important aspects of scope and purpose statement. They fail miserably mainly because of a lack of enhancements or redefinition of the Web service description. The description is in most of the cases equivalent to the set of terms obtained from the WSDL documents, thus functionality can be only guessed as far as a service designer chose corresponding wording with actual functionality. All but one of the solutions, fail to address a scope of Web services, therefore they are treated as artifacts solely useful to developers.

3.3 Redefined Web service description

This section focuses on the solutions that enhance the Web service description by addition of external data that is to facilitate retrieval or make the description more feature-rich. These extensions may be viewed as a description redefinition, as operations on the enhancements are the core of the reviewed solutions. In essence, all semantic annotation initiatives belong to this category. Possible exclusion from this

category may be a result of integration of approaches stemming from the traditional IR and any others with the semantic annotation, which blurs the border between this category and the subsequent one.

3.3.1 A critical review of the chosen solutions

One of the attempts to address the state of affairs in Web service description is [Verma and Sheth, 2007]. It is of significance as the author emphasizes the requirements, which are to be accounted for a purpose statement, when Web services are to be used in the Service Oriented Architecture. Three major approaches were enumerated, namely an agreement of Web service provider with its clients that is to be prior to any Web service usage, an external textual description that is left for processing to end users and finally an annotation based on external models expressed with controlled dictionaries, taxonomies or ontologies. The cited work promotes research efforts by authors of METEOR-S [Patil et al., 2004]. It takes into account previously mentioned challenges of functional semantics and more, as it addresses also data semantics, non functional semantics and execution semantics. The key element of the presented work is a summary of the advantages of SAWSDL that enables to extend WSDL documents by the functional and data semantics. It achieves this by enabling the end user to make a decision as to the means of semantic annotation. This is an important change as, no longer one has to use complex description languages that are costly in adoption, both in terms of learning curve and tool development. Unfortunately, promised service agnostics with aim to result in a greater flexibility, achieves it by introduction of high heterogeneity and additional costs of production of various mappings between different descriptions. Thus, the presented solution solves trivial test cases remarkably well, but it is felt that it might not perform well in real-life situations.

The work [Rocco et al., 2005] presents the shortcomings of, available at the time, Web service discovery solutions and proposes a novel approach for the Web service description. It takes as a base a Service Class Description (SCD), which is a complex entity that binds definitions for types, flow graph [Hecht and Ullman, 1974] and a set of templates used to test whether a Web service fits to a Service Class in question. SCDs were chosen to leverage specific information distinguishing a set of Web services from the available corpus. Results of performed experiments proved that the approach yields good results. The approach presented is novel as it is one of the first noticing the recurrently highlighted challenges and not employing mainstream semantic description languages in favor of flow graphs. Even so, flow graphs have

to be crafted manually by system designers and be constantly updated whenever a new type of Web services has to be processed. There are no comparative studies demonstrating whether crafting of flow graphs is more or less complex task than a description using a specialized semantic description language.

Among the earliest attempts to rationalize use of Web services annotated with semantics one has to recall the work of [Zhang and Li, 2005]. Rationalization is understood as a partition of semantic description into segments describing different aspects of a Web service such as its core functionality, its Non-functional Properties and a special domain of interest. This allowed for a better performance, as a general functional description could be brief and domain specific ontology could avoid inclusion of redundant concepts. One can highlight the fact that this is one of the earliest visible efforts to harness complexity of semantic description by quitting with a monolithic design. As mentioned this boosted efficiency and facilitated work of those only interested in a particular domain.

A purpose of Web services was presented as a type of contract in [Tosic and Pagurek, 2005]. The idea of the contract presented there is inspiring as it is one of the earliest attempts to demonstrate the complex perception of Web service per se. The authors outlined three types of contracts that in general concur with other research teams trying to encompass the perceived Web service's complexity. Types of outlined contracts are functionality, quality and infrastructure. The cited work also presents findings from the research on available solutions using Web services as elements of abstraction construction. The authors summarized available solutions as non sufficient. None of them provided all features necessary from their point of view. What is of interest, only OWL-S was classified as a language that can fulfill a functional contract in all of the aspects devised by the authors (syntax functionality, behaviour functionality, synchronization functionality and composition functionality). The authors advise use of a mix of surveyed languages to achieve full coverage of desired contracts. As their main focus was on Quality of Services, the functional contract was not discussed in great detail and there was no discussion on performance issues of the proposed approach.

The purpose of [Colucci et al., 2003] is to present a reader with a general use case of Description Logic [Nardi and Brachman, 2003] and how it can be applied to a process of service matchmaking. This is very important as one can tell whether a Web service being searched for can be located in a repository. As other works delving into this matter, the authors focused on features and capabilities of their proposed solution and did not discuss its cost.

Semantic annotations were introduced during era of dominant use of UDDI as the Web service registries. It was early observed that UDDI suffered from a lack of many vital features. Early efforts of semantic annotations focused on devising new ways of extending a service representation along with sidestepping functionality barriers imposed by UDDI. On the other hand, efforts such as presented in [Colgrave et al., 2004] introduce more evolutionary attitude to service repositories. The cited work is especially important as it introduces ability to plug-in external service providers into a proprietary repository of an organization. This is a key feature of this work as it shows the importance of a repository to be a kind of a market rather than a simple container for organization's Web services. Other functional extensions followed. One of them was ability to express Quality of Service of Web Services in a manner proposed in [Zhou et al., 2005]. First works on extending UDDI with semantics and its possible impact on Web service retrieval are [Colgrave et al., 2004] and [Paolucci et al., 2002]. Several efforts such as [Treiber and Dustdar, 2007] focus more on the technology of access and distribution of Web services rather than on changes in a manner of their description.

One of the most important works in the domain of Web service description and discovery is [Plebani and Pernici, 2009]. This work presents the URBE system. It is one that uses primary WSDL documents but it can also make use of data provided by semantic annotations. URBE makes use of SAWSDL which leverages OWL-S as an ontology description technology. The primary service description focuses on a decomposition of names of WSDL elements where tokenization techniques are employed to present the system with a set of terms that later are used to perform reasoning with the use of the general ontology. This work presumes that a sufficient source of data for the general ontology is data stored in Wordnet. The authors underline the fact that ambiguity in case of this choice is an important challenge and when the domain is unknown for a given set of Web services, it is very difficult to resolve. The secondary measure taken by the authors is processing data types founded in Web services. This action allows for a quick classification of inputs and outputs in case of the simple data types. Complex data types are matched with custom prepared heuristics taking into account trade-offs between precision and cost measured in the processing time. Tertiary, the authors present a mechanism that allows for quantifying overall similarity of Web services taking into account all of their properties. This can be extended by additional annotations. There semantics hooked in with SAWSDL is treated as high fidelity data as there is no risk of ambiguity. In general, URBE presents an interesting approach that steps beyond simple IR techniques

and employees broad analysis of WSDL documents. It can be potentially applied to large repositories. Having in mind that project is still not finished, one can point out that it does not address the issue of service purpose. Its main advantages come from automatic processing of WSDL documents that does not state the purpose explicitly. The possible semantic annotation expects ready ontology expressed in OWL-S, thus quality of results is dependant on ability to prepare this ontology and annotate available services with it. This is especially relevant as, Wordnet chosen as a data source for their implementation is great aide, but shall not be trusted due to a number of problems with its structure. [Andrikopoulos and Plebani, 2011] is continuation of their effort where actual results are presented and compared to other approaches. The results are encouraging, yet without further extensions the presented times are unacceptable for business users.

In majority of the reviewed works, there is no reference to an actual cost of Web service description. This can be troubling especially, when one is to argue that Web services enhanced by a semantic annotation are to further revolutionize SOA. In [Kungas and Dumas, 2009] its authors use this argument to present an alternative approach to Web service description where no ontologies or controlled vocabulary is required. The authors decided on the bottom-up description building taking into account the most promising data first. This allowed for a construction of ontology concepts where there was none in the first place. Their method proved to give good results. However, the repository used by the authors consisted of only 1000 operations what cannot be widely recognized as large enough to draw conclusions. Further, the method was applied and built using operations coming from Web services centered on a set of related objectives. Thus, one can reason that it was far easier to tune the presented method to come up with good results. This does not negate its usefulness, yet it hampers its use in a situation where a far more varied body of Web services is in need of annotation. Moreover, the process of annotation is only semi automatic, thus, an expert user must be present at the annotation stage to fill in missing concepts.

There are various related issues concerning the Web service description. One of them is signalled before enhancing Web services with data on their quality referenced as Quality of Service (QoS). A significant reference in this matter is [Kritikos and Plexousakis, 2009]. Of most importance is a list of benefits that a Web service gains from inclusion of a QoS description. These benefits can be in a part applied to Web service applications in SOA. To begin with, the Quality of Service extensions allows for improvements in Business Process Modeling [Hommes and Reijswoud, 1999] as there are various vital characteristics ready at a design time. Further, when there

are several Web services that fulfill the same purpose, one can choose the one with best characteristics as for the undertaken task. In addition, the definition of the QoS allows for monitoring of Web service performance in terms of designed characteristics. Lastly, stating what QoS are connected to a Web service and what is their value one can foster a set of backup strategies allowing one to protect oneself from the buoyant environment to some degree. The authors, proposed to define the QoS as a set of concepts related within an ontology. This is a commonly agreed solution, and it does not hamper performance as much due to a relatively simple structure of the QoS. One can risk arguing, that the QoS in their most basic form is a set of tags marking various operations where a scale of possible values is attached. The cited publication is valuable, as it demonstrates merits of the QoS for Web services especially when they are to be a cornerstone of SOA.

3.3.2 Summary

As visible in table 3.2 one can easily observe that the overall level of support for effectiveness is definitely higher than in the case of the previously reviewed solution group. Unfortunately, when confronted with the level of cost, which is also high, one might infer a direct correlation between these two aspects. One of the interesting approaches addressing this issue is given in [Stollberg et al., 2007], an idea of cache introduction is very plausible yet anyone implementing it must take into account additional execution time costs of its construction and updates.

Due to the fact that a high number of solutions was based on fully-fledged semantic solutions, a great number of Web service descriptions simply cannot be processed due to the complexity of the matchmaking mechanisms. Solutions that employ a divide and conquer approach have encouraging results and serve as a good example for future implementation efforts.

There is a slightly greater impact on the scope of the reviewed solutions. This can be seen as a grater effort to include the Service of Quality metrics and in some cases additional data gathered as user feedback.

The most visible change in comparison with the previously reviewed solutions is the surge in the level of support for the purpose statement. It is significantly higher and there are even solutions that allow for stating explicitly what is a goal of given Web service or its operation. Nevertheless, most of the surge is explained by the qualities of underlying semantic technologies, which to some degree support the idea of a purpose attached to a Web service.

Table 3.2: Summary of a WSDL based retrieval solutions

	criteria:				
	effectiveness	cost	scalability	scope	purpose
solutions:					
[Verma and Sheth, 2007]	◆	◇	◇	◑	◑
[Rocco et al., 2005]	◆	◑	◑	◇	◑
[Ren and Xu, 2008]	◑	◇	◑	◇	◑
[Zhang and Li, 2005]	◆	◇	◇	◇	◑
[Sriharee, 2006]	◆	◇	◑	◇	◑
[Mao and Le, 2009]	◑	◇	◇	◇	◇
[Ye and Zhang, 2006]	◑	◑	◑	◇	◆
[Wu, 2009]	◆	◇	◇	◑	◑
[Tosic and Pagurek, 2005]	◑	◇	◇	◑	◑
[Colucci et al., 2003]	◆	◇	◇	◇	◇
[Lee et al., 2004]	◆	◇	◇	◇	◇
[Radetzki and Cremers, 2006]	◆	◇	◑	◇	◑
[Prazeres et al., 2009]	◆	◇	◇	◇	◑
[Plebani and Pernici, 2009]	◆	◑	◑	◇	◇
[Paliwal et al., 2007]	◑	◑	◆	◇	◇
[Nayak and Lee, 2007]	◑	◇	◑	◇	◇
[Luo et al., 2006]	◆	◆	◑	◇	◇
[Lee et al., 2007]	◆	◑	◑	◇	◇
[Kungas and Dumas, 2009]	◑	◑	◇	◇	◇
[Kritikos and Plexousakis, 2009]	◑	◑	◑	◑	◇
[Averbakh et al., 2009]	◆	◑	◇	◑	◇
[Celik and Elci, 2006]	◆	◇	◇	◇	◇

Solutions are evaluated towards five aspects with three marks. A black marker stands for a good level of support for given aspect, a half black marker stands for partial support, and an empty marker stands for lack of support.

3.4 Hybrid solutions

This section covers solutions not easily assignable to any of the previous categories. There are important observations and traits that should not be underestimated in the study of subject.

3.4.1 A critical review of the chosen solutions

There are important approaches that do not repeatedly reuse already available technologies in the Web service description. An example is [Kuster and Konig-Ries, 2007b] where DIANE Service Description (DSD) is presented and discussed. The authors differentiated themselves from other technologies by equipping DSD with the specialized ontology language that aims at a creating a lightweight description of Web services. Lightweight is also to be understood as efficient to process. It was decided that DSD should start with a general object oriented approach and extend it by four significant, both for users and the system, characteristics. First one is capturing an effect of Web service invocation. Thus, there is a way to determine its purpose. The effect is captured as a state that maps on an actual instance from the available ontology. Second one takes the notion of an effect of Web service invocation a step further and enables descriptions to contain a set of effects for every Web service. This is due to the fact that various Web services can have a number of effects depending on their inputs. The previous observation leads to third extension that allows selecting a desired state depending on a variant that one is interested in. Envisioned sets gain a feature of being configurable with variables. Forth extension covers evaluating elements that use fuzzy sets to represent varying preferences and come up with a reasonable answer that fits previously recorded preferences. These extensions were proved to work well while applied to SWS-Challenge[1]. It was emphasized that features provided by DSD, especially its treatment of requirements allow for the most complete discovery approach. One has to notice that inclusion of Web service purpose statement and ability to efficiently describe varying requirements is very rare, yet the essential property of available description approaches. Nevertheless, all of these features were achieved by breaking with de facto standards for Web service description and investing considerable amount in preparation of necessary ontologies and facilities allowing for capturing changes of state in any of envisioned possible situations. While this breaking with de facto standards might be perceived as an issue, one might argue that this is the right way as it introduces a new flexibility that might be a deal breaker in the Web service description domain. Further efforts to boost the solutions usability are presented in [Kuster and Konig-Ries, 2007a].

Almost general agreement on extending Web service descriptions with semantic annotations expressed in OWL, WSML or RDF is sometimes extended by introducing differing hybrid solutions. Such is the case with [Kona et al., 2006] where a Universal

[1] `http://sws-challenge.org/wiki/index.php/Main_Page`

Service-Semantics Description Language (USDL) is introduced. USDL's features are a mix of semantic description technologies, a WSDL document and ontologies built upon Wordnet [Miller and Fellbaum, 2007]. The authors decided to adapt Wordnet to the Web service description with application of OWL Wordnet ontology. It was argued that this ontology balances between being coarse-grained on the one hand and universal on the other hand. One is made understand that USDL is a formal language for service documentation. This is achieved by describing a Web service in reference to one global ontology. Extensions are performed with addition of predefined tags to a WSDL document. A comparison to other initiatives is given, yet it is fairly inconclusive as the authors state that USDL is superior to already introduced solutions as it does not allow for under-specification of Web services. On the other hand, it is argued that USDL can be a companion description language to OWL-S. One has to emphasize that although using a global ontology is a bold idea the problems arising from this decision can be overwhelming as Web services differ a lot and have various domain peculiarities. Further, Wordnet is a great resource, but it has many problems with over-specification and under-specification of some domains that result in extremely unstable results when applied to Web services greatly varying in domain of application. Lastly, USDL does not address performance and cost issues as it mixes in semantic technologies in the description blend and requires a manual Web service description's extension.

Due to lack of common consensus for a manner of how to convey a semantic description concerning Web services a number of propositions were given in [Hausmann et al., 2004], which described an approach for conveying added semantic value with visual representation derived from Unified Modeling Language [Schewe, 2001] and formal rules. The approach in general, did not succeed to become a standard, yet motivation behind it, which was to overcome conceptual complexity of Web services for various interested parties, was noteworthy.

A broader approach towards services is presented in [Cardoso et al., 2010]. The authors highlight a need arising from not only the SOA paradigm but also the vision of Internet of Services (IoS [Buxmann, 2009]). It is argued that, what concurs with previous observations, available technologies of Web service description aim at providing a distributed infrastructure for supporting business operations. The key to realizing the higher level IoS vision is building additional layers that would allow treating Web services as business services oriented on delivery of added value to business users. With all these, a presentation of Unified Service Description Language ($USDL^2$) is given. Analysis performed by the authors led them to several conclusions concern-

ing business services and their representation in USDL^2. All conclusions are drawn while taking into account a shift of perspective from IT infrastructure one, towards enterprise one. They are formulated as a set of sections in a general model addressing those features that are especially lacking in traditional IT infrastructure oriented approaches. The general model of USDL^2 joins three views on a Web service under one initiative, namely business view, technical view and operational view. Thus, a Web service represented in the USDL^2 shall store data on participants and they role in a Web service life cycle, provision and consumption. There are also means of expressing the Service Level Agreements (SLA [Maclaren et al., 1995]) linking it directly to Web service operations and its general functioning. A curious aspect, very novel in the Web service description, is an extension with data that can support marketing actions. Marketing issues are covered by sections devoted to pricing, documentation and certification. As noted by the authors, a Web service must consider various legal aspects of its invocation. There is a section that describes terms of use of given Web service. Synergy matters of a Web service use, led to inclusion of bundling description. This section describes how various aspects change when a business user decides to opt in use of other Web services offered in conjunction with a given Web service. The final conclusion reflects the constant need of change and adaptation, as one is not able to foresee all description needs. It is achieved by extension mechanisms. One has to be aware of the fact that, aspects addressing operations description, interaction, classification, functionality and those associated with Web service as a part of some workflow are described in the operational part of the USDL^2 model. What is interesting, this part is the first attempt at incorporating important features usually left to other external ways of description. A technical part focuses on technologies used to achieve business and enterprise objectives. As is clear from the above given description, the referenced work has greatly shifted focus towards enterprise needs. USDL^2 strives for being a truly universal description tool by joint use of the following elements: name, textual description, keywords and ontological concepts. The authors argue that it is necessary, as without this set of features a description cannot be truly universal. The vision presented in the cited work is a significant attempt to make IoS happen. One shall miss detailed examples or experiments that would allow him to better understand mechanisms envisioned. A comparison with already available technologies would also be a greatly beneficial to the idea itself.

The Web service description could be defined much broader when stating a purpose of a Web service in question. An example is work [Jin and Liu, 2006] where introduction of semantic description led to a proposal of building a description that

would allow for discovering the best use of some Web service. This is achieved by categorizing Web services with use of domain ontologies. The authors remarked that a true purpose (meaning) of Web service can be known to a user only when he is able to determine how a given Web service interacts with other Web services and what effects are available thanks to this interactions. Upon this remark, the authors developed a set of prerequisites towards a structure and scope of necessary ontologies that enabled them to express their idea. These ontologies allow for presenting a Web service as a triple defined by a Web service context, a set of its interactions and a set of scenarios where a Web service would be engaged. The method given by the authors can be viewed as interesting extension to the research area. Nevertheless, the method cannot cover general case of Web service description due to the varying characteristics of Web services in general. The authors, for their experiments, chose a domain that is rich with interactions and investigating states being their outcome really suit their method. Moreover, a number of additional semantic descriptions is considerable and its structure is not obvious for non-expert users. Therefore, the lack of ability to cover a general case and high cost of description prevents one from implementation of presented method.

Treating Web service as a Representational State Transfer (ReST [Fielding, 2000]) resource led to works such as [Karimpour and Taghiyareh, 2009]. The authors decided to augment the Web service description by annotation with concepts from Wordnet. In addition, as Web services are treated as ReST resources authors were interested only in input and output parameters. A Web service description is extended manually through presented tools. When Web service descriptions are ready, they are matched with user requests by means of Wordnet semantic distance facilities. It is a low cost operation in comparison with reasoning done by specialized software. Experiments done by the authors showed very good results in terms of recall and precision. Unfortunately, one has too few details to decide whether experiments performed are trustworthy, as there was not enough data concerning queries, in terms of their structure and length, and used Web services' corpus. As the approach presented is a bold attempt in boosting performance of Web service discovery, one has to emphasize description deficiencies. First of all, Wordnet as already remarked, has a number of problems with granularity of its concept structure. More, synsets can contain concepts that are ambiguous when an exact synset name is not specified. Not stating the purpose of a Web service is yet another deficiency. Representing only inputs and outputs, even in augmented manner can disqualify the reviewed method from any compositional challenges.

An example addressing some earlier observed deficiencies is given in [Iqbal et al., 2008]. This work is centered on four main semantic aspects that were defined during analysis of a Web service discovery. These four aspects are functional semantics, data semantics, non-functional semantics and behavioural semantics. The most important aspect is the functional one, as it strictly defines which Web services do not satisfy user needs expressed in a query. Data semantic and non-functional semantics are useful in terms of tuning result set, where some characteristics might be more favorable than others, especially in terms of data type relations and various properties that usually are covered by the QoS. The mentioned two aspects also tend to be of secondary importance to the matching process what might be in contrary to user's view on the matter. Finally, behaviour semantics tries to capture a Web service interaction with others and formulate, which actions are allowed under what conditions. The authors observed that any successful Web service discovery has to start with obtaining user's need. This need has to be formalized in a way that makes it possible to match expressed need across available Web services. The authors used SPARQL [Perez et al., 2006] for defining user needs and purpose of Web services. Web services are described in SAWSDL [Kopecký et al., 2007], which as remarked, is agnostic of particular semantic technology. Solution presented by authors is indeed a flexible one, and allows adjusting itself to user needs. However, one has to have in mind a fact that this flexibility comes at a certain price. All the functionality is based on semantic features that need a lot of computing power to work reasonably. What is more, semantic solutions do not scale well with addition of new Web services and are expensive in terms of user time both for actual annotation and for training. These undermine real world applications in SOA oriented enterprises.

What is very interesting, there are approaches totally neglecting semantic annotations implemented with technologies such as WSMO, OWL-S or basic RDF. The most complete example of the Web service description with a custom model not using enumerated above technologies is [Haidar and Abdallah, 2009]. A whole discussion is started by stating a question on the purpose of Web services. It is then restated as a question on what are key aspects of producing a good abstraction for a Web service. One has to emphasize, that the authors also highlighted a simple fact, missed by many, that a Web service is an abstraction itself. When one of definitions of Web services is to be recalled, it is a software system designed to provide interoperability among remote machines. More, it does not implement features essential for enabling it, thus it's an abstraction. Taking into account these observations, the authors presented a solution that uses state-based formalisms. This state formalism is tested

with SOAP implementation and test results indicate that the solution is functional. The greatest issue, that one is bound to notice is that chosen Web services are rather simplistic. A lot of the effort has to be invested into this type of description. Without enterprise scale tools this might not be achievable outside the research community. Nevertheless, this is important work as it effectively counters arguments for Semantic Web services being the only available solution for the Web service retrieval.

Trends being an outcome of Web 2.0 paradigm also affected the Web service description and retrieval. A study of applying tags to Web services to improve the retrieval process was given in [Chukmol et al., 2008]. The authors closely examine possible strategies for annotating Web services with tags both as a single term or a phrase. They motivated it by lacking textual data for Web services and weak alternatives from Semantic Web services due to high cost both for users being in need to process complicated formalisms and for providers being obliged to prepare them. The authors proposed two models for retrieval, first based on keywords and second using free text annotation. Both provided techniques were explicitly given with discussion on merits and angles of future research expansion. One has to acknowledge that author's proposal is very stimulating as it allows for stating a Web service purpose with a tag. As authors have enumerated, this solution is not without flaws. Yet, flaws such as tag reputation, quality and various malicious attacks aimed at the system cannot be fully solved at technological level of interested community [Gupta et al., 2010]. What is more, the authors do emphasize that the tag based solution for a Web service description can achieve success in communities that evaluate Web services to be tagged. This narrows spectrum of applications to specialized and high-cultured ensembles organized around high quality software projects. Nevertheless, this is a valuable strategy that should surely be investigated while preparing any solution supporting Web service description and retrieval.

3.4.2 Summary

Table 3.3 summarizes the solutions reviewed in this section. As one easily can notice, the level of support for different traits is more evenly distributed than the two before reviewed groups.

The cost and effectiveness are both lower than in solutions in the previous group, yet scope and purpose definition is the most important differentiating aspect of this group. Especially the purpose statement is to be recognized as first class citizen of

Table 3.3: Summary of a Hybrid Web service description solutions

solutions:	criteria: effectiveness	cost	scalability	scope	purpose
[Kuster and Konig-Ries, 2007b]	◑	◑	◇	◆	◆
[Kona et al., 2006]	◑	◇	◇	◑	◑
[Shafiq et al., 2010]	◑	◑	◑	◑	◇
[Hausmann et al., 2004]	◑	◑	◇	◇	◇
[Cardoso et al., 2010]	◑	◑	◑	◆	◑
[Huang et al., 2005]	◑	◑	◇	◑	◑
[Jin and Liu, 2006]	◆	◇	◇	◑	◆
[Jin et al., 2007]	◑	◆	◆	◇	◇
[Hu et al., 2009]	◇	◆	◇	◆	◑
[Haidar and Abdallah, 2009]	◑	◇	◑	◇	◆
[Bravo et al., 2008]	◑	◇	◇	◇	◑
[Chukmol et al., 2008]	◑	◆	◑	◑	◑

Solutions are evaluated in five aspects with three marks. A black marker stands for a good level of support for the given aspect, a half black marker stands for partial support, and an empty marker stands for lack of support.

the Web search description. The scope of solutions is also considerably broader than before.

3.5 Other efforts considering the Web service description

Before giving the overall statistics obtained from the quantitative and qualitative analysis, there are some efforts visible in the literature review that could not be included in any of the reviewed groups due to their nature. They are presenting experiments, views and methods rather than complete solutions that could be assessed with reference to the enumerated Key Requirement Aspects.

The Web service description usually defers a discussion on perception of Web services. This, many a time ignored, topic is addressed in [Galle et al., 2008]. This work targeted the topic of how different groups of people would like to interact with

the Web service description. Surprisingly, few groups are interested in unabridged WSDL documents as they do not propose a uniform way of extracting key aspects of various possible views. The authors proposed a set of roles such as Web service architects, Web service engineers, requirement engineers, domain experts and managers that are interested in different data presented in specific manner. Possible range of presentation layers includes spatial projections using uniform representation patterns, glossaries summarizing operations available and functional descriptions usable for composition and a purpose deduction. The idea envisioned by the authors is depicted with an exemplary Web service that is presented in various ways along with a discussion which way suits what role the most. In general, this input is valuable, as it underlines often forgotten aspect of Web service description. One can argue that some important roles were omitted and one example is not sufficient for the whole argument. Yet, the key message cannot be left uninitiated in any work addressing a Web service description.

Out of many publications available on the Web service discovery, [Crasso et al., 2010] is an important input into the discussion concerning the Web service description. It differs from other referenced material as it does not propose a brand new approach for the description. It enumerates the most important problems found during an inspection of Web services gathered from the Internet. It enables one to easily notice why many proposed methods for the Web service description cannot deliver proposed performance in terms of precision and recall. First of all, the authors after analysis, presented the most common problems found in surveyed material. The most crucial one was that names used in description of various parts of WSDL document were ambiguous. The ambiguity stemmed from too high a level of generality that although common in names used does not add any information that would allow for conclusion for those interested in discovery of given Web service purpose. Similar, yet slightly less occurring problem was a lack of comments or their low quality. Solutions, that would like to base their functionality on comments, shall inevitably fail in more than 50% of Web services, taking into account data provided by the authors. There were several other problems connected to the structure of the WSDL documents, data types used and fault propagation strategies. Overall, the authors surveyed 391 Web services gathered from the Internet. Their quality is much below satisfying level due to enumerated problems. To test whether authors' observations were accurate, they decided to apply a set of modifications based on their observations. As a result, they proved that not only WSDL documents became more compact, but also already proposed solutions for retrieval yield better effects. The cited work is valuable to

anyone dealing with the Web service description. One would only hope that a similar approach is at some point in time available for Web services used inside corporations. There is no way to ascertain that their global quality is considerably better than those publicly available. General lessons are to be drawn from the authors input both by developers and architects of retrieval solutions.

There are very few works that try to summarize efforts of the Web service discovery. One of them is [D'Mello and Ananthanarayana, 2009] that presented a prototype of solution that surpasses UDDI in efficiency of Web service discovery. This work is a study with a detailed description of one possible approach to store and retrieve Web services. It's completed with an experiment on tiny corpus of Web services. Arguments made by the authors and their solution have their merits. Nevertheless the most important value added is observation made as to the types of available Web service descriptions. The types enumerated are:

- those based on syntax,
- those based on behaviour,
- and those using semantics.

Every use of a traditional retrieval technology no matter what features are implemented and what strategies to better manage number of processed Web services fall into a syntactic type of the Web service description. Approaches focusing on the extraction of the interaction characteristics among Web services are classified as behaviour based description. Finally, the Web service description extended by additional information such as formalism codified in one of available semantic description languages or simple addition of a tag falls into category of semantic description. This categorization seems to fully cover available solutions, but one might argue that there is an obvious lack of category covering a purpose of given Web service and its operations. Of course, the purpose of a Web service is highly elusive as what matters the most to business users and developers is an operation fulfilling their particular needs.

The authors revised and extended their categorization effort in [D'Mello and Ananthanarayana, 2010]. This work focuses specifically on more granular categorization on both Web service discovery architectures and Web service discovery techniques. This work is an important overview of approaches available in literature. The previous work was restructured and main division line between approaches is driven with functionality description based methods and those that focus on non-functional description methods. The authors conclude their presentation of built taxonomy with postulates on Web service description objectives.

First of all, they would like to present a Web service and request for it in a natural form. This should be augmented with inclusion of all important domain objects and explicit declaration of features and available actions that can be performed. Second, the authors proposed using action nouns in formal descriptions in order to better match postulated enhancements on semantic framework. Third, the authors proposed to present all the available actions and additional participating objects in sequential manner as this shall help in the retrieval process when a kind of ranking of results is necessary. Fourth and last, all these should be represented in a compact manner to avoid introduction of inefficiency in the retrieval process. All these conclusions were made under assumption of a crucial role of the Web service description and retrieval in regard to overall success of Web services.

3.6 Query performance of semantic solutions and IR based solutions

The effectiveness of any proposed model has to be benchmarked against other leading solutions available. This is a non-trivial objective as the scope of each of the solutions can be very different. Mote, the technology driving these solutions can differ greatly. Therefore, the two most important groups have to be surveyed:

- classic Information Retrieval based systems - treating Web services as text documents,
- semantic oriented technologies - those equipped with a special set of extensions, usually in a form of a language that allows for reasoning with the available data.

3.6.1 Information Retrieval based solutions

Classic IR based techniques are the fastest ones in terms of pure execution speed [Ounis et al.,]. This is mainly due to the fact, that choosing a document as a feasible one is a matter of checkup in a previously prepared index. The implementations prepared by the biggest Web search enterprises demonstrate that thousands of millions of documents can be covered in fraction of a single second [Brin and Page, 1998].

Without scientific breakthrough and considerable funds one cannot compete with the already available solutions offered by various vendors and communities.

The list[2] of the most important initiatives based on [Middleton and Baeza-yates, 2007]: ht://Dig, Indri, IXE, Lucene, MG4J, Swish-E, Swish++, Terrier, XMLSearch, Zettair.

Results provided in [Middleton and Baeza-yates, 2007] demonstrate that average search time of 2.7 GB of test data was well below 32 milliseconds in 2007. Since the referenced work, new projects of note have arrived accomplishing better results.

Most notable addition to the above list are Sphinx[3], Solr[4] and Xapian[5].

The most up-to-date available benchamrks[6] demonstrate that it is possible to obtain query results in well under 20 milliseconds for approximately 3 GB corpus of documents and below 110 milliseconds for 10 GB corpus. This is achievable on typical workstation equipped in 4 core server equipped with 16 GB or RAM.

Even better results are available to companies such as Google and Microsoft where specialised teams deploy optimized solutions on the specially prepared hardware.

3.6.2 Semantic

Semantic oriented technologies are a varied ensemble. The overview of the most important was given in 2.3.2. Majority of the enumerated and described approaches is driven by a reasoner. It is crucial element in order to provide a query result.

As previously reported, this special programme is capable of deriving new facts based on those provided in form of previously predefined ones, many a time referred to as axioms. The need for formal description was outlined in [Horrocks and Tessaris, 2002].

There is a number of competing reasoners, both open source and proprietary. They vary greatly in terms of performance, which is dependant on the task. In a number of publications it was established that there is no single reasoner that performs equally well in a number of tasks devised to measure the performance. One of the most important works that aim to synthesize the answer on the performance of reasoning is [Thakker et al., 2010].

[2] The full list of sources for the enumerated search engines is given in the referenced work.
[3] `http://sphinxsearch.com/`
[4] `http://lucene.apache.org/solr/`
[5] `http://xapian.org/`
[6] `http://notes.jschutz.net/2011/03/sphinx-search-engine-comparative-benchmarks/`

The most notable reasoners are: Pellet[7], KAON2[8], RacerPro[9], Jena[10], FaCT++[11], HermiT[12].

The referenced work focuses on solutions that can be applied to measure performance of queries with the most important benchmarks. One has to talk about a solution, because a reasoner is only an element of total infrastructure. Domain refers to the complete solutions available to load, process, optimize, cache and return a query result as a store. Therefore, all following discussion will be centred around the notion of a solution.

There are a number of benchmarks comparing performance of various characteristics of semantic enabled solutions.

The first one of those is University Ontology Benchmark [Ma et al., 2006]. Unfortunately, not only this benchmark is perceived as a faulty one but so are others in terms defined in [Weithöner et al., 2006] (explicitly brought up Lehigh University Benchmark LUBM [Guo et al., 2004] and those focused on application of SPARQL functionality). Not all of the enumerated nine requirements available in referenced criticism must be agreed upon, but it is essential to notice that many a time the most basic ones are neglected.

Having established this, one can view the numeric results provided by [Thakker et al., 2010] as a reasonably safe start point to further inquiry.

Thakker et al. [Thakker et al., 2010] have established that the two most robust solutions are BigOWLIM [13] and Allegrograph [14]. Both of those are more than standalone reasoner, they are a complete solutions that could be deemed as semantic repositories. They were tested with the mentioned LUBM benchmark [Guo et al., 2005]. The ontology used for the full description of the organization modelled for the benchmark is relatively simple. Yet, number of individuals and units multiplied by the total number organizations modelled in the biggest test scenario results in over 20 thousand million explicit statements.

These statements are understood as both given and computed data originating from the test repository. In order to answer the also provided queries, the whole dataset must be imported and preprocessed. The process in the biggest test case

[7] http://clarkparsia.com/pellet/
[8] http://kaon2.semanticweb.org/
[9] http://www.franz.com/agraph/racer/
[10] http://jena.apache.org/documentation/inference/index.html
[11] http://owl.man.ac.uk/factplusplus/
[12] http://hermit-reasoner.com/
[13] http://www.ontotext.com/owlim/editions
[14] http://www.franz.com/agraph/allegrograph/

lasts over twelve days. There is no data provided by the software manufacturer on the query time for the biggest test case, by the one processable on a desktop-class computer (1.8 thousand million statements, loaded in 14.4 hours) results in an average query completion time of 199 seconds [15]. This average represents a number of queries that both run extremely long due to its complexity level and those that are completed under 25 milliseconds.

The most obvious issue is the time that is spent on loading and preprocessing the test cases. The benchmark was not prepared in a form that takes into account updates to the ontology driving the tests. Dynamic addition of entities annotated with the used ontology was also not addressed by the benchmark designers. Used quotation for workstation set at 2000$ can buy at the beginning of 2012 a computer equipped with a single processor that can have up to 8 cores each in hyperthreading mode. This allows for parallelised execution in 16 processing streams.

Other important insights, that have to be included in this place, are provided in [Stegmaier et al., 2009] and [Faye et al., 2012]. There, the Berlin SPARQL Benchmark (BSBM [Bizer and Schultz, 2009]) is used to measure robustness on a number of semantic solutions. As the name of the benchmark used suggested, queries prepared as an element of the test case focus on SPARQL, which is one of the most popular solutions when querying RDF stores. SPARQL[16] is the official recommendation of WWW Consortium[17] and was prepared to handle queries to a number of different data sources.

Nevertheless, the original concept behind SPARQL is handling of RDF data. Other data sources are addressed through middleware that is responsible for the necessary transformations from some source to the desired format. The flexibility of this language was tested in several experiments [Sirin and Parsia, 2007, Kollia et al., 2011].

The most valuable of the RDF-based solutions is given in [Stegmaier et al., 2009]. It covers a wide range of solutions both open-source and proprietary. The tests were performed over repositories built from 100000, 1000000 and 5000000 RDF triples. The results are extremely diverse, what is important is the fact that none of the tested solutions could handle all of the queries in comparable time. The fastest computed queries for the largest dataset where no time-outs occurred took as little as 100 milliseconds but with other queries the execution time spiked to over a minute.

[15] http://www.ontotext.com/owlim/benchmark-results/lubm
[16] http://www.w3.org/TR/rdf-sparql-query/
[17] http://www.w3.org/

Even more up-to-date data is provided in [Morsey et al., 2011]. This work focused on introduction of new benchmark that increases a total number of properties and classes used in test cases along with emphasis on test data and test queries being real. There are over 153 million RDF triples in the full benchmark test case. As in previous benchmarks the query performance is extremely varied. The fastest solution from the tested was capable of handling over 200 queries per second. The same overall fastest solution could handle only a fraction of query per second for specific test queries. What is more, the structure of queries proposed in [Morsey et al., 2011] is different from those in BSBM and LUBM benchmarks. Due to their real world origin, they can be perceived as less complex than those crafted synthetically.

Due to the advances in technology, different methodologies and scope of these solutions results in a situation where they cannot be compared on one to one basis. Yet, it is possible to summarise the situation in the following points:

- capability to produce query results depends on the robustness of preprocessing the data in the test dataset,
- preprocessing does not eliminate a situation in which some queries yield results after thousands of seconds,
- the number of statements handled by current solutions is well above limit of 20 thousand million, and the newest versions of the surveyed mechanism strive for 100 thousand million statements[18],
- the number of RDF triples tested in the available literature is beyond 25 million (benchmark's leader is newly built solution Stardog[19],
- the systems capable to handle the maximums reported above exceed well beyond what is a current standard of a workstation[20] or regular servers[21], tests performed by producers of AllegroGraph software were carried on a system equipped with 32 cores and 1 TB of RAM,
- query processing time is strictly dependent on its complexity, the geometric average for the best solution in the SP^2Bench SPARQL Performance Benchmark [22] as given in [Schmidt et al., 2009] is 8.96 seconds (while arithmetic average is over 870 seconds) where test scenario considered 5 million RDF triples, the previously mentioned Stardog achieved better results as it reached geometric

[18] `http://www.franz.com/agraph/allegrograph/agraph_benchmarks.lhtml`

[19] `http://stardog.com/` - it was excluded from the previous comparisons as it did not exist then

[20] at the beginning of 2012 a typical workstation could provide up to 8 hyper-threaded cores in budget of 2000$ with 16 GB or RAM

[21] as in previous footnote, 2 8 core processors with 64 GB RAM in budget of 10000$

[22] `http://dbis.informatik.uni-freiburg.de/index.php?project=SP2B`

average of 0.11 seconds (ignoring one timeout) and 4.49 seconds arithmetic average (also ignoring one timeout), both tests were on the same benchmark test scenario with additional difference of hardware used (2 vs. 8 cores, 3 vs. 8 GB or RAM and different CPU architectures Intel Core Duo vs Intel i7),

- statement and RDF triple load time is a considerable issue, every technical problem or system crash incurs a penalty starting with 20 minutes for 100 million RDF triples and 338 hours for 100 thousand million statements.

3.7 Conclusions from the literature review

The reviewed literature allows for conclusions concerning state of the Web service description and discovery. In general, for over 10 years there is no consensus whether Web services should be enhanced with semantic annotations or some other method that increases effectiveness of retrieval. The solutions proposed range from a simple retrieval based on processed keywords available in WSDL documents to those that not only use technologies such as WSMO and OWL-S for description of basic functionality but also expression of other, desired by users, features.

The most prominent trend in the Web service description is the semantic annotation. Various initiatives discuss merits on application of metadata in a form of ontology encrypted in one of the most popular semantic annotation technologies. The greatest benefit of description with an ontology is ability to drastically increase precision and recall of retrieval process. It is achieved by unprecedented power to express a model of world. One can prepare a description of a world in such details he sees fit. Unfortunately, this rebounds at level of complexity for both the world description builders and the model end users.

First of all, the process of designing an ontology reflecting some piece of world consumes time and assumes some level of skill in its in builder. Using enterprise-scale tools to leverage the task does not solve issues completely, especially that these tools are not common and their market penetration is low [Cardoso, 2007].

Second, end users are not interested in extra complexity. They would like to obtain simple answers to queries whether some artifact exists. Once more, systems that hide complexity from an end user are scare, but even when interface is designed to be helpful, an end user has to embrace a language of the presented model in order to use it [Jansen et al., 2007].

Third, the semantic description was envisioned to provide answers on concepts originating from some given ontology. It was not designed to do that fast. Processing

time of some ontologies may be measured in tens of seconds ([Prazeres et al., 2009, Stollberg et al., 2007, Zhang et al., 2009, Chu-Carroll and Prager, 2007, Jónsson et al., 2006]). Some queries may be processed in seconds. These results are unacceptable for an end user [Nah, 2004].

On the other end of the spectrum, there are description methods that delve into terms used in WSDL documents. In its most naive form they propose indexing of all available terms and presenting results as ranked list of WSDL documents that match query terms. More advanced solutions allow for preprocessing of terms in order to filter out possible synonyms and ambiguities. It was proven that it yields better results than the naive approach. Using the traditional retrieval can be perceived as an iterative process as previous improvements gain recognition and are included in the later works. Thus, when filtering of synonyms, and partial disambiguation of terms became established technique, it was enriched by attempts to build an ontology of concepts available in corpus of Web services accessible to researchers. This was an important step, as it allowed for further gain in effectiveness and in retrieval time. As a drawback, one has to highlight that from this point, any retrieval had to be aided by a human operator to some extent.

A few researchers decided to describe Web services in alternative manner. The most interesting method devised is capturing a Web service as a patter of states and transitions between them. Flow graphs enabled to perceive what one can do with a given Web service as one is presented with a list of possibilities. One has to underline that these attempts although tempting, cannot relieve Web service descriptions builders and end users from effort spent on learning how to efficiently model a Web service and later retrieve it.

Another solution from the alternative category, is a complete redefinition of technology that describes a Web service. It is realized by a new model that allows for expressing Web service with a new language in order to achieve goals stated by its authors. Other is employing description tools usually applied to very different artifacts, an example of such misplaced technology for Web service description is the Unified Modeling Language.

One last subcategory are hybrid approaches that do not focus on Web service description technology per se, but on its scope. This is the most varied category by far. The common denominator for its members is inclusion of features that are ignored by previous categories. As Quality of Service and Non-functional Properties are handled in some solutions (WSMO recognizes NFPs, it realizes some of the QoS with them) distinctive features come from recognition and addressing issues such

as multiple perspective of Web service, Service Level Agreement bound to specific operations and Web services, trust and ability to resolve fuzzily stated requirements.

It is crucial to highlight the most important conclusion drawn from the literature review: *the technology in which Web services are described is of secondary importance.* The key issue is the right spectrum of features. There is no single solution present that could address the most important demands.

To put this statement into a perspective let's review the coverage of various aspects by the presented Web service description efforts. The data is given in table 3.4. The reviewed works were selected from the body of over 100 that were gathered from the relevant conference proceedings and journals. Tables summarize 46 solutions divided into 3 groups closely reviewed in the preceding sections.

The overview presented in table 3.4 allows for composing a conclusion that none of the groups perform well in overall coverage of the defined aspects of the Web service description. Nevertheless, it serves as a good tool to provide a direction for a solution that could perform significantly better. It shall be achieved by inclusion of traits well covered by the reviewed groups and balancing conflicts resulting from orthogonal nature of some aspects (such as cost and effectiveness).

Table 3.4: Summary of the aspect level coverage by reviewed groups.

		criteria:				
		effectiveness	cost	scalability	scope	purpose
groups:	**WSDL based**	50.00%	58.33%	50.00%	16.67%	8.33%
	Redefined	81.81%	22.72%	27.27%	11.36%	25.00%
	Hybrid	50.00%	45.83%	25.00%	45.83%	50.00%

Data is obtained from tables 3.1, 3.2 and 3.3. Markers are translated in a following manner: a full black marker is given a value of 1, half black 0.5 and empty marker becomes 0. Then an average is computed for each group.

Concrete ideas and trends that should be included in a solution that could perform better than the available range of solutions are given below.

First of all, one would need a purpose statement for every Web service. This is realized by a few solutions, one of them is WSMO which allows for Web service goals. Nevertheless, these service goals are expressed as capabilities addressing every element from IOPE quadruple. This cannot be a universal solution as a business user is not interested in the preconditions and the postconditions (effects). He is interested

in finding a Web service that brings concrete results and he wants to find it without extra effort in analyzing ontology interdependencies. It was observed that tagging systems, based on some taxonomy are of great interest as they aid to address these needs. A Web service that is categorized not with some unrelated terms, but with terms coming from the business user's environment.

Thus, the Web service description shall take into account its context. Without it, it is yet another abstraction layer that can find application possibly for developers when it is properly documented. By the Web service context, one can understand its application in an organization. Why was it prepared, and in what terms it was documented. Finally, how it was classified by its builders.

More, as stated in [Cardoso et al., 2010], a Web service can mean different things to different users. A business user would like to acquire a building block that can be employed into his business process and enable him to produce added value. A developer would like to be able to locate Web services that can implement some desired functionality in order to save him from unnecessary work that could be invested elsewhere. An architect of organization's system would like to audit a state of affairs and quickly asses whether some functionality is under or over represented and act upon this data accordingly. An external contractor would like to quickly check whether he can introduce some functionality so that it can find application in an organization and thus bring him revenue.

A good Web service description (good in terms of defined KRA and support for different user perspectives) should allow for multidimensional tagging with a number of taxonomies. One cannot believe that neither these taxonomies nor ontologies should be built automatically [Gacitua et al., 2007,Brewster et al., 2009]. The process can be aided by traversal of available documentation and additional input from users and accessible resources with a good reputation in terms of quality and reliability. Ultimately, every taxonomy must be prepared by a skilled user. Yet, when prepared for some compact area it should be still easily comprehensible for users unlike oversized ontologies striving for depicting domain exhaustively.

Abstracting from the defined categories, one can allow oneself to draw a dichotomy of the Web service description. And it should be neither based on the technology nor on the attitude towards functional or non-functional features. Let us assume that a Web service description should be divided into two groups:

- The first focuses on attempts to describe Web services making use of traces of purpose stored in documentation, retrieved with various algorithms trying to

make sense of names used in a WSDL document or any other method that uses a Web service invocation context as a clue.

- The second group shall represent all those solutions that are not WSDL centric. Whether they provide extensions with WSMO, OWL-S, SWRL [Horrocks et al., 2004], pure RDF, DSD or USDL is not important, as they key value is representing Web service in a broader environment.

Semantic Web services were designed to make it available for software agents to operate on them. As for year 2011 this is not the crucial objective. The research community is far more interested in initiatives such as Internet of Services or Software as a Service [Sun et al., 2007]. Thus, once more, a user, a human being is in center of these initiatives. A person is not interested in complex scripture-like semantic languages, he is not interested in flow graphs and transition algebra. A user would like to be presented with a solution that makes it easy to narrow his search task to a manageable set of feasible Web services that comply with his needs both in functional and non-functional terms.

Part II

The multi-perspective utility driven model for the Web service description and retrieval for the modern electronic economy

Chapter 4

The proposed model for the Web service discovery

4.1 Chapter outline

Taking into account the presented Key Requirement Aspects (given in section 2.6) and the critical analysis of the available solutions as given in the domain literature review and the most important technologies that support their majority, the novel model designed for this thesis is outlined here. This proposed model is specially designed to include traits not emphasized enough in other available approaches. It includes results of author's experiments obtained on the self-assembled corpus of Web services originating from the open Internet. Thus, it is closely related to the real-world data. In addition it was built using the data provided by the professionals originating from various organisations. They varied greatly in terms of their function in their organisations (project managers, programmers, consultants, administrators, business analysts, executives) and the profile of their organisations[1]. It is designed to be flexible enough for very different groups of users. It aims at balancing traits that are perceived as orthogonal in nature (such as pure execution speed versus expressiveness and effectiveness). The proposed model presented takes into account not only description but also discovery of the Web services. Therefore, it is a comprehensive approach ready for adoption in organisations implementing the SOA paradigm.

The proposed model itself can be viewed as a proposition that changes the focus from the purely technological one, to a one that does not alienate business and ex-

[1]The thorough review is given in section 6.1.1

ecutive users. The whole model can become a valuable addition to other initiatives such as electronic marketplaces [Cheng et al., 2007].

Obviously, the instantiation of the proposed model could provide a backbone for more advanced business scenarios, as having the description and the retrieval process covered with this model, other important issues can be addressed.

Table 4.1: Summary of measures used to address the KRA by the presented model

aspect:	measures[2]:
effectiveness	$\alpha\beta\gamma$ phrase, NFPs, Local Context Anchoring, Source reputation ranking
cost	Semi-automatic shortlisting, $\alpha\beta\gamma$ phrase, Locally Controlled Vocabulary
scalability	$\alpha\beta\gamma$ phrase, Source reputation ranking, caching strategies
scope	NFPs, dashboard, aggregation and categorization, namespaces/ SU
purpose statement	$\alpha\beta\gamma$ phrase

As clearly visible nearly all of the Key Requirement Aspects reference the $\alpha\beta\gamma$ phrases. It is crucial to note, that all of the model features and qualities stem from inclusion of the phrases as the model aims at balancing various important features derived from the traditional Information Retrieval techniques and semantic annotation languages.

In order to fully discuss the proposed model a following structure is given:

- Introduction to the model in terms of the important background challenges and requirements;
- Description of the most important model's notions and mechanisms;
- Model's application scenario.

Such a structure was chosen, as it is impossible to introduce the proposed model without providing a detailed account of model features in conjunction to the requirements gathered, differences to existing solutions and addressing a number of requirements being special cases and derivatives of the initial five Key Requirement Aspects.

4.2 Motivation for the proposed model and its general premise

In order to realize a vision of flexible and robust architecture based on Web services one has to design a set of features that are a definite necessity for prospective users. Flexibility of the Web service retrieval depends on its description that should capture both technological and functionality features. WSDL documents proved to give a good baseline for a technological description. They convey data on possible actions and means of accessing them in a standardised manner.

What was already emphasized, a WSDL document lacks a clear description of service functionality. The reviewed initiatives, such as OWL-S [Antoniou and Van Harmelen, 2004], WSMO [Roman et al., 2005] and SAWSDL [Kopecký et al., 2007] decided to invest heavily in a clear identification of every operation by appending semantic metadata (full range is given in section 2.3.2). This identification is based on a reference to a common ontology or a set of ontologies integrated for the sake of a given application. This enhancement resulted in highly precise results, especially in comparison with solutions based on traditional Information Retrieval methods.

Yet, along with beneficial effects, a lack of scalability and performance issues became apparent in instantiations applied to a task of making a choice out of thousands of Web services and tens of thousands of Web service operations. Moreover, available solutions do not lessen negative effects of information asymmetry [Hadar and Fox, 2009] on the market.

The work presented in [Bashir et al., 2010] defines seven main limitations of current solutions that are crucial to address in order to handle a growing number of Web services and their dynamic discovery. From the original list given by the authors of referenced publication, the most important issue is the lack of mechanisms that would change the way of retrieval of Web services in the context of current user's need. The context is to be understood as a variable set of requirements both functional and non-functional expressed in a variety of manners.

Market leaders, such as Microsoft, IBM, SAP, provide customers with tools enabling them to implement their own infrastructure as Software as a Service. Further, a broader use of Web services and SOA opens new ways of software design and implementation. All these open a chance for appearance of the Web service markets [Barros and Dumas, 2006, Conte et al., 2010].

Every market needs a robust mechanism for a representation and a retrieval of its goods. A market for Web services cannot be fully functional without tailored solutions that would enable its users to retrieve Web services that match a desired functionality along with some additional constraints derived from a business environment.

This cannot be understated as the Web services, being an abstraction for encapsulated programming routines, could not thrive if one is unable to locate them within boundaries of any given market [Reichwald et al., 2002, Tamm and Wünsche, 2003].

The overall premise of the model was defined as fulfillment of the Key Requirement Aspects to the fullest degree. As is presented later, it is constrained by several tradeoffs at various levels. The source of constraints is discussed, and motivation for the actual choices is given. Such general premise is derived from the dissertation's thesis and research goals and questions accompanying it.

The more specific premise has to be formulated so that it relates the dissertation's thesis and the research goals. The answer for the detailed research questions formulated in section 1.2 is realised by introduction of the following elements to the postulated model:

- the Sub-organisational Units with authorities on description of IT assets,
- the Local Controlled Vocabularies,
- the cost-effective purpose statement structure,
- the description extension with Non-functional Properties,
- the categorization and aggregation of Web service resources based on user provided labels,
- the mechanisms overcoming SU boundaries when users venture beyond their original SUs.

The above list is a list of the artifacts that were produced throughout the research activities carried out for the purpose of validation of this work's thesis. They are derived from the in-depth analysis of the needs of an organisation leveraging Web services in the Service Oriented Architecture paradigm.

4.3 Web services in organization's environment

This section is focused on a presentation of a number of important challenges that are directly connected with the Web service description and retrieval in general. As these general relations provide the motivation behind various features of the postulated model, the examples and a broader discussion must be given.

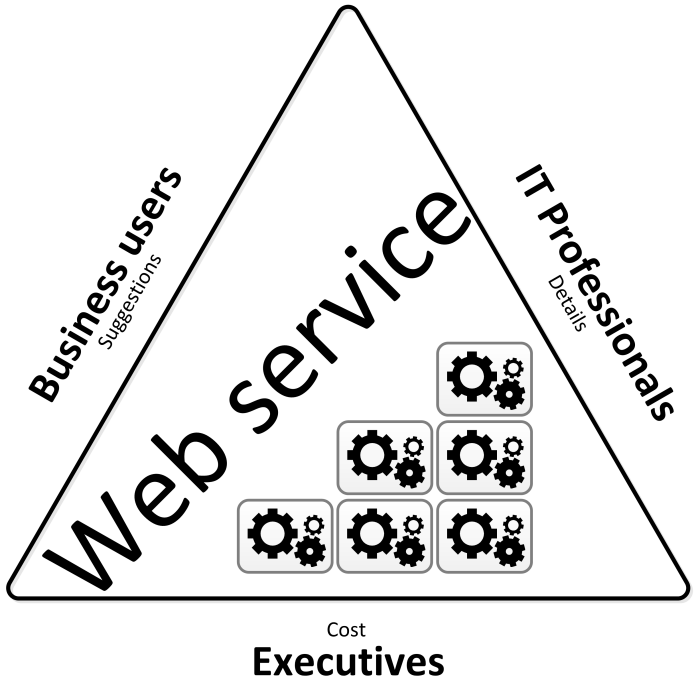

Figure 4.1: Different perspective of a Web service

4.3.1 Users and Web services

As there is no single solution for functional description that is to be understood by the advanced users such as most of the developers and people that are not interested in technical details such as the business users, it was felt that a desired solution has to include important compromises to make it available to any user [Günther et al., 2007].

The users interested in Web services originate from the following three groups:

- Executive users,
- Business users,
- IT users.

Each of these groups has different objectives when dealing with Web services. Executive users are a group that is the most interested in overall picture of Web service usage, as thanks to it they are able to make better decisions. They are interested in a cost optimization on various levels and dependencies among various projects and processes inside the organization.

Business users are oriented on searching for ideas and available implementations that can be used in reaching goals given to them as a result of activities they are carrying out. Moreover, they are interested in a quick access to a set of non-functional

characteristics that can provide information on suitability of a Web service operation being the subject of examination.

IT users have possibly the broadest interest in a Web service. They want to be provided with various details, usable in crafting of new Web services. These cover a number of details on who was the Web service prepared for, who prepared it, what is a context of a WSDL document, whether upgrading a Web service can break Service Level Agreements and where the Web service is currently deployed.

The situation is depicted in figure 4.1.

4.3.2 The common plane of Web service description - purpose description

So far, it was established that for a Web service treated as an interface, the description available in Web Service Description Language is a specialized type of metadata, which is designed to be usable to other software in order to generate a gateway to some remote functionality.

In addition, one has to always remember that, the most important parts of a Web service, from the perspective of one planning to apply it to one's particular goal, are its operations. Each Web service operation describes some actual functionality. This functionality does not need to be associated with other Web service operations. Nevertheless, due to packaging in a single Web service, there is some relation. The most common example of this kind of relation, is a situation when a Web service operation is using a single system in order to perform advertised functionality that is common for other operations bundled in the Web service in question. Other kind of this relation is one that makes one Web service operation a necessary step in invocation of others. Let us assume that an authorisation procedure can be an example of such a situation.

Taking into account the above, one can easily see what was previously stated, that available means for description might be unsatisfactory, even in the simplest of possible scenarios. Thus, as one should avoid yet another standard of description, a need to extend available means of description has to be emphasized at this point.

It was natural observed by others and solutions such as SAWSDL referenced in 2.3.2 were designed and implemented.

Extension per se, gives one nothing, yet by inserting a number of to be postulated description data elements, it becomes a mean to bring desired functionality that

addresses lacks of two most prominent solutions in the Web service description as reviewed in chapter 3.

An important issue in describing the postulated model's context, is the realisation that for a long time Web services and their semantic versions were thought of as either building blocks for business processes that should become a new functionality as a result of some automated composition procedures or as independent entities stored in some kind of repository where their functionality could be reached by a free text query. While both described cases are viable and many a time they might be the only relevant ones, it was noted that a Web service described in a WSDL document can be something completely different to different stakeholders interested in the same organization.

To exemplify, let us consider a possible scenario when both the traditional Information Retrieval techniques and the advanced semantic annotation might fail as a retrieval tool. The simplest test for a success is a verification of availability of some desired functionality in organizations repository. It is not obvious that at all times there will be a successful answer ready for a user. By a successful answer one has to understood a result that with 100% certainty informs of either functionality existence or its non-existence.

Whether any of solutions succeeds or fails is a result of matching the terms used by actual Web service developers or people, which provided requirements for the Web services with the terms used by the one issuing the query.

One has to remember that every department, project team or developer has a slightly different vocabulary and the same process can be mapped in his mind to a different set of terms. As this is fairly obvious for a long time, organizations tried to manage it by issuing various policies and guidelines that shall result in unambiguous description of different entities [Francez, 1982, Tzagarakis et al., 2000, Government, 2005]. The semantic solutions can also fail due to simple fact of lack of entry into knowledge representation structures for the actual used terms. The best inference engine cannot produce an insightful result without data on relations between queried term and concepts held in the ontology or semantic network.

All the groups enlisted in the previous section need Web services for their specific purposes. Hence, they have to successfully identify a Web service operation or a whole Web service. As given in the first part of this work, available initiatives are varied and propose a whole range of features that benefit various interests. There are many interesting ideas that range from using linguistic patterns, through flow graphs,

lightweight semantic annotations to complex solutions including many different techniques.

Nevertheless, one can definitely state that there is not a single initiative that provides a straightforward means of stating the purpose of a Web service operation. The purpose that is the most important notion when somebody is trying to retrieve Web service operations that suit his goal. One can discuss cost, compare NFPs [Ortiz et al., 2005] or monitor the performance when he is capable of finding Web services that provide the functionality he desires.

It is the utmost goal of Web service description, the effective retrieval (also in terms of the cost) of Web services providing some functionality. As given in section 2.6, the effectiveness and the cost investigate how much one must spent in times of organization's time and effort to come up with a list of Web services satisfying his need. The state of the art analysis, demonstrate that there is a tradeoff between effectiveness and cost. The cheaper the solution in terms of time and effort, the less effective it is. Yet, the absolute precision, can lead to unfathomable constructs that are not acceptable by its potential users.

Hence, the model proposed in this work is driven by the need of straightforwardness. It uses additional description layer, but this layer's complexity is kept to minimum. It was designed to provide a purpose description with maximum relation to the concepts and terminology used in the organisation. The syntax of the description is brief and avoids clutter. It reflects many successful initiatives such as SQL and RDF, but trades expressiveness of advanced constructions for effectiveness of execution in a number of usage scenarios. It dissolves the responsibility of the maximum effectiveness on a number of supporting mechanisms.

The introduction of model's outline is proceeded by its constraints in relation to which it was designed and the introduction of a number of crucial to the model's existence notions that are examined in detail afterwards.

4.3.3 Constraints for the proposed model

With the Web services in the SOA paradigm one has to deliver a set of constraints and agreements, which organize the proposed model. First of all, one has to underline the fact that each Web service has to have its goal or purpose defined. This purpose is to be defined with terms. Atomicity of the purpose is at the level of a single operation. Each operation does one thing well, those that do not comply due to multiple goals must be reworked, usually by partition into a set of more specialised ones.

The above agreement is a first step to organize the available functionality in an organisation. Such an agreement does not deviate for usual practices inside organisations that on a regular basis introduce order into their assets. The standard plan for accounts in financial departments, the common structure of employee datasets, the code production policies are examples of the already applied agreements. The standardisation is at times induced by regulations to achieve various goals.

The premise of the agreement is that by homogenization of description, one should gain more than he could loose due to drop in expression flexibility.

Having established the first agreement a list of other constraints and agreements concerning the Web service operation description follows:

- Every Web service operation does one and only one function well. It is preferred that the operations are mapped with systems and processes that they interact with on the level of purpose. This aims at clear definition what is being done to what entity with additional information. The idea is later explained in depth as it is a reason for phrase structured descriptions.
- The presence of Web service operation description is mandated at every level of its production, from the initial design, through its implementation and to its deployment in business processes.
- Every group responsible for introduction of the new Web services and Web service operations must prepare a catalogue of vocabulary and phrases for use in the Web service description. This vocabulary has to be rich enough in order to make it possible to describe both coarse grained and specialised functionalities. When a unique unambiguous term is already used a compound one is allowed. Terms, and compound terms have to refer to organizational communication standards and context established terminology.
- Every Web service operation must be available not only through a functional description and technical one prepared in a WSDL document, but also through a number of auxiliary access channels. These focus on a Web service operation usage, its involvement in various projects, business processes, Suborganisational Units and teams. In addition, data on Quality of Service parameters and Service Level Agreements should lead to a catalogue of Web service operations characterised with desired level of parameters. Lastly, data on retrieval results has to be connected with the choice made by the user responsible for retrieval.
- Available Local Controlled Vocabulary catalogues are separate, yet there have to be means to provide a viable mapping between them. This ensures the ability

to locate valuable resources that were prepared and deployed by various teams across organisation.

4.4 The model's core notions

Taking into account the already accomplished research there is a number of traits that a model of Web service should include. Especially that a comprehensive model has to provide mechanisms for an effective discovery of Web services, taking into account a wide range of requirements defined by various groups of users. From the precision point of view the most important thing is to get Web services that match users' requests. As mentioned before, semantic languages allow for that, yet they incur additional cost. This cost is a combination of time invested into a training and a description of every Web service. The later is additionally disturbing as it is a common practice that every code artefact should be documented what allows for further maintenance.

One could think of a perfect situation, where Web services expressed as WSDL documents were annotated at the same time they are documented. What is more, a further optimization should directly force anyone documenting and annotating WSDL documents include a number of words from some controlled vocabulary. This vocabulary might be either a thesaurus or a semantic net or a flat list of important terms. As inferring is out of the scope of this work, either of mentioned structures offer a good combination of efficiency (in terms of speed) and transparency for the end user. Annotation should be performed so that each atomic element of a Web service must be annotated with words from the common vocabulary. As emphasized earlier, the atomic entity is a Web service operation.

It is easy to notice that a sum of all terms used to describe each operation of a Web service might not yield a coherent picture of its functionality. Alas, this is a very common situation in Web services gathered from the open Internet. One might believe that the situation is different in organizations that possess a strict set of rules that codify handling of various types of artifacts. This might or might not be true. Many a time Web services found in organizations were built to integrate some heterogeneous systems. At the level of general purpose, all operations can be described as facilitating some data interchange. When one is to step deeper, to the level of a developer, this picture might become invalidated as there is a number of different systems that constitute this integration Web service. It makes user want to know, which systems are used exactly, because it matters from his point of view.

When a given Web service integrates four separate systems it addresses four separate issues. Thus, every operation needs to be annotated separately as a Web service might not be perceived as a homogenous entity.

Definition 1 (Local Controlled Vocabulary)**.** *Taking into account, the above stated facts, one has to introduce a notion of Local Controlled Vocabulary. This, depending on the needs of the organisation can be a thesaurus, a semantic network or a flat term list that is used to manage the task of description of Web service operations in the context of some part of an organization.*

As organizations might be of arbitrary size, a one centrally managed vocabulary might be counterproductive due to a number of bureaucratic obstacles, therefore a federated structure of vocabularies is postulated. This postulate introduces some caveats as there is a risk of repetition of some terms. To overcome this problem a well known solution of namespaces is to be used. Every Local Controlled Vocabulary must be described in terms of a unit or a group that is responsible for it. While searching for an operation by a set of words, results will be grouped by their namespaces. This is an interesting feature, as after retrieval a user can quickly inspect results and notice whether a hit is stored in a namespace that posses some interest for him.

Definition 2 (Namespace)**.** *A unique identifier assigned to a single Sub-organizational Unit in order to differentiate possibly ambiguous terms originating in the Sub-organizational Unit's area of authority from areas owned by other Sub-organizational Units.*

Definition 3 (Sub-organizational Unit)**.** *Sub-organizational Unit is a minimal part of organization's structure that has a complete authority on its area in terms of description of assets such as Web service operations with the terms pertaining to the Local Controlled Vocabulary built to manage the assets in this area.*

The obvious advantages of this scheme over formal descriptions (implemented with one of the semantic description languages) is its straightforwardness for a user. There is no learning curve that descends deeper than absorption of vocabulary that is used everyday in his closest environment.

When a user is to venture beyond the boundaries of his everyday interests he is aided with a list of namespaces mapping other Sub-organizational Units on their software artifacts. This is important, as when one is to break with tedious procedures he is more likely to consult a search service in order to answer a question of whether

something similar exists in a organisation. One may think of this as a utility that helps avoid suboptimal decisions by providing necessary data.

The cost side of the envisioned solution, at this point, is covered by intra organizational agreement on documenting Web services in a specific manner and honouring organizational policies.

More, a Local Controlled Vocabulary must be assembled to make it work. Cost of this operation should be manageable as even when started organization-wide, the actual expenditure is at a unit level. One is to assume that the unit size is limited.

There can be important exceptions from the above scenario. When an organization has one unit producing all the Web services for the rest of the units, compartmentalization should be devised in such a manner that it reflects actual demands across the organization. Developer teams responsible for software writing should include terms found in documents resulting from requirement analysis included in Local Controlled Vocabulary in addition with documenting resulting Web services and their operations.

Additional effort should be recuperated when additional elements of systems are requested and when new team members are taking over maintenance tasks of some specific project. As previously highlighted all investment in extra description should enable quicker and more confident decisions when a request for new functionality arrives.

Local Controlled Vocabularies should be most effective if and only if organization implementing them agrees to give a complete control on an atomic namespace to a single management. Preferably a single person should wield the control over the vocabulary as there is no risk of conflicts. The management role is of utmost importance because it has the ability to define the structure of queries and decide what terms are acceptable in a given namespace. This allows for further features such as clustering of Web services based on their functionality described with the LCV [Cattuto et al., 2007]. This bears close relationships to semantic networks as structures for representing a language [Fellbaum, 1999]. Instead of describing entities in terms of subtypes and instances, one is interested in relations such as more general, more specific or vogue equivalent.

4.5 Formalization of the proposed model

Let us consider a quadruple $< \alpha, \beta, \gamma, \nu >$ where

- α denotes an element of the A set that contains every possible α-phrase element,

- β denotes an element of the B set that contains every possible β-phrase element,
- γ denotes a subset of the Γ set that contains every possible γ-phrase element,
- ν denotes an element of the N set that contains defined Non-functional Parameters with some specified values for these parameters.

It stores information of functionality of a Web service operation. It is its functional phrase-based description. A single Web service operation described with a given quadruple is referred to as σ. The repository of Web service operation descriptions is denoted as Σ.

The sets A, B, Γ and N are the domains for the respective elements of the tuple that forms a concrete Web service operation description. When giving a valid description α and β hold exactly one element from their respective domains. γ can hold multiple elements from its domain. The ν element of the description is a vector of pairs: NFP and its value. The NFPs are defined specifically for interested organisation along with the metric for the value. It is believed that all σ available in the global repository of Web service operation descriptions Σ should have ν phrases of equal length containing single set of NFP-value pairs. This allows for meaningful comparison across the organisation.

The searching for a Web service operation description deals with the two most important elements. The sought Web service operation description σ and the available repository of Web service operation descriptions denoted as Σ.

The result of matching sought σ against the Σ is a result of the following operation: $\sigma^A \cap \sigma^B \cap \sigma^\Gamma$. The result is the set of all the Web service operations satisfying phrase constraints of the sought Web service operation, it is denoted as Σ^σ. Σ^σ must be checked against the ν set so that a final answer set can be computed ($\Sigma^{\sigma\nu}$). This set can contain 0 or multiple elements.

It contains 0 elements when a $\sigma^A \cap \sigma^B \cap \sigma^\Gamma = \emptyset$. It can be also empty when there is no $\sigma^\nu \sim \nu_i$ (nu_i is representing every individual ν set from N). The similarity is used instead of equality, as some parameters can be tolerated with values higher or lower than the one given. The situation is a result of the fact that nature of the parameters is determined by a organisation's needs. $\Sigma^{\sigma\nu}$ is non empty when the intersection of the sets of chosen terms is non empty and the similarity between the NFP vectors holds.

The building of Σ begins with division of the whole organisation into sovereign Sub-organisational Units that build their particular fragments of A, B and Γ. Their are responsible for incorporating commonly used terms into a framework so that it can be used in further ventures. One of the aides in this objective are standard measures

applied in IR, such as Term Frequency and Inverted Document Frequency [Ramos et al., 2003]. Application of those, results in generation of $AB\Gamma$ set. $AB\Gamma$ is a directly proceeding human-managed building of individual fragments of A, B and Γ sets that are made distinct by incorporation of unique namespace connected with sovereign SU.

The set containing the results obtained in the main scenario - $\Sigma^{\sigma\nu}$, is a basis of further refinement. This refinement determines the actual order in which given propositions are presented to a user.

The most important strategy that affects the order of the presented results, is taking into account the overall popularity of given Web service operation described as σ. The more the given σ is chosen by users or the more it is used in a variety of projects the higher it is being displayed in the overall ranking. The whole transformation is denoted as a *ranking*.

Other strategy is the use of information on proximity of a user base SU and namespaces in which Web service operation descriptions are found. It was decided that there is additional bonus for those that are more related in terms of organisational structure to the interested user. The whole transformation is denoted as $proximity\_evaluation$.

Therefore, user is presented with a result of application of both transformations over the $\Sigma^{\sigma\nu}$. The presentation ready set is denoted as $\Sigma\Sigma$ ($\Sigma\Sigma = proximity\_evaluation(ranking(\Sigma_{\sigma\nu}))$).

The above-given apply to a scenario when a user is searching for Web service operation with full knowledge of the structure of Web service operation description and actual terms used in the respective phrases. When he cannot act so, there is a must for a mechanism that maps user provided input onto the available resources.

Such mechanism is provided by Local Context Anchoring. It is a transformation that changes a set of unidentified terms into a set of those that can be matched with some actual Web service operation description available in the Σ. Additional hardship for such a transformation is lack of data on the role of the individually provided terms (in a sense of positioning it as an α, β or γ phrase). This set of unidentified terms is denoted as x. Therefore, $lca(x) \rightarrow \Sigma$. This transformation is not deterministic, as there is some probability that $lca(x) \rightarrow \emptyset$. Therefore, it is a heuristic transformation.

LCA can operate only thanks to a number of resources that allow for finding relations among members of the x and anyone of A, B and Γ. The resources available have different value. What is more, not all relations can be deemed as useful. Thus,

all resources must be ranked according to their reputation for an organisation and sensible guidelines for relation approval must be used.

4.6 Additional mechanisms catering for specific group's interests

The Local Controlled Vocabularies and division into a number of Sub-organizational Units with exclusive authority on naming schemes are very important for the success of the whole model.

4.6.1 Mapping the business goals with Web service assets

Business users might be interested in Web services as they are implementing a business process and want to check the repository for possible functionalities that should be inside it. Without extra description, be it semantic annotation, or good documentation they are not able to state whether some desired functionality is already present.

This is important to notice as two way communication between high level business users and low level IT professionals consumes a considerable amount of time thus invokes an extra unnecessary cost [Ferreira et al., 2011]. It is far better to present a business user with a tool that will let him know what functionalities are currently supported thus, enabling him to shortlist those of potential interest to his task or project. Equipped in that shortlist he can confirm positively or negatively whether these Web services can satisfy his needs. Additionally, these shortlists allow IT professional not to play guessing game, but confront a business requirement stated as a goal and Web services that were listed as possible solutions. This can be of high importance in shortening the time necessary for the right decision. Seeing the shortlist allows for answering whether there are really irrelevant services in the list, one or many fits, and the recommendation which to use follows or simply a request for building a new Web service is made.

This request is tightly connected to the business need thus, greatly influences the cost optimization processes, as it draws a line between a Web service and a business task. The postulated shortlist becomes a ranking of LCV based Web service operation descriptions. Thus, Local Controlled Vocabularies and Sub-organizational Units managing the LCVs are crucial to the whole enterprise.

The solution implementing the postulated model allows for mapping of subsystems used in the organization to the business needs what allows for optimizing of the organization functioning. Imagine feedback from the system as a way of profiling company spending usage, one examining it, could see patterns that say what Web services are supporting which business goals and by what subsystems they are being realised, there can be decisions that follow analysis of this report. Some of this decisions can be changing the usage of the systems, closing some and putting tasks realised from some systems to the other ones. This is additional cost effectiveness, as running some systems not only cuts costs in term of energy savings but also in terms of maintenance. One has to remember that maintenance of systems is not only a cost of personnel, but also a number of contracts that allow to support the systems when in some breakdown occurs. Knowing what systems are used the most one can cluster other functionalities there. Of course this can be troublesome or involve misleading data, but gives an inkling to what is happening with the organization when it runs on the cloud and the SOA principles.

IT professionals can observe which services are used where. More, additional cost of documenting Web services with extra features is low compared with traditional semantic annotation technologies [Zhou, 2007, Mokarizadeh et al., 2010]. Once there is an agreement to use a postulated model is reached, an addition of new terms to LCVs is a manner of internal need.

This breaks with a developer oriented view of Web services and invites business users to examine available solutions themselves. This solution emphasizes that there should be a common way to obtain data on vital functionalities that could be easily used while making decisions. Of paramount importance is the paradigm of doing everything solution can to come up with an answer for the user. Additional way of obtaining an answer is matching free text queries on available markup terms. This technique has become a standard mechanism in information retrieval systems and has become some point a natural thing a system does in eyes of its users [Jansen et al., 2007]. Those who are familiarized through experience or job requirements to use other types of Informational Retrieval systems shall find no advanced options available in terms of issuing queries due to the inherent simplicity of the model.

4.6.2 Aggregation

Additional measure, that should help boost user experience is a ranking functionality. This ranking functionality centres on a number of requests of given Web service, level

of match of users query and number of times when other users chose a Web service in question and user's affiliation in terms of organizational structure. This is another method of simplifying access to potentially valuable data with minimum effort on specifying additional criteria.

A link between the demand stream (projects, organizational units, business goals) and Web services has to be underlined as it must be a first class citizen. Many a time this channel of information is neglected and although it should not be. Knowing exactly what is being used to achieve some business goals is empowering for business users. This might be viewed as a bridge among technology centric Information Technology personnel and task oriented business users. When one is to concentrate on a business user perspective, an examination of some project triggers an action that stores data on all resources used for its development. Traditionally, this resources are people and their skills and necessary additional tangible and intangible assets. This model postulates that a mandatory extension is explicit markup of Web services used for a project in question. This is a benefit for all parties involved as a business user does not have to identify technical details. He only needs to point, which project or projects he is interested in (in terms of similarity or further maintenance) and all involved Web services are presented to a development team. This shortcuts the overall process of gathering information resources necessary for any given project to launch. If no organization operates on a standard such as BPEL [Jordan and Alves, 2007], the postulated extension is to be perceived as an index to diagrams describing underlying processes with enhanced notation by inclusion of terms from the Local Controlled Vocabulary.

A similar purpose is served by including data on Web service usage in terms of various projects and sub-organizational units. The idea is straightforward, thus one can hope that is is readily available to any type of user. These data available can easily be visualised in a dashboard manner [Cleverley, 2001]. An insight resulting from association of projects and business entities with Web service operations used there is to be achieved at various levels by different types of interested users.

Thanks to gathering the above described data, it can provide a mean to categorise Web services in some coarse grained groups usable while presenting query results. This should enable users to map general usage of the categorised Web services and their operations in organisation's structure context to their own ideas on this structure and should facilitate general interaction with the Web service repository.

The concept of namespaces is somewhat overlapping with the categorisation possible by inclusion of the mentioned data. Configurable labels attached to the obtained

groups might span across various organisational units, projects and business processes. Hence they might give a broader view than a simple listing of Web service operations in a namespace defined for some actual Sub-organizational Unit.

In order to make the labels meaningful, it must be agreed upon what should induce membership in any given group. This decision can only be made by knowledgeable and experienced users whose propositions should be included as a initial setup that can be altered later.

The whole process is driven by an association of terms and phrases used in organisation, so that it is comprehensible for its users. Official aliases that map unofficial terms on the ones stored in corporate documentations are also available. While reviewing any given groups, one should focus on clear identification of the most frequent use cases of the elements of the group under scrutiny.

The obtained mapping is valuable as will be demonstrated later. It is by no means the only and the most important, but it is helpful in a number of scenarios. It is also important due to the fact that it presents Web services with no additional technology oriented details usually found daunting to the business users.

Group denomination can greatly boost the functionality for a user, yet this action similar in essence to building a set of tags (that many identify with a tag cloud which is essentially a visualisation [Aouiche et al., 2004]), allows only for a relatively small narrowing of resources possibly useful to a user. What is more, such a set of tags demands a wise balance between a number of concepts in the tag set and the number of Web services matching this concepts. Too few concepts shall result in a very long listings demonstrating the overall category. Too many keywords shall make it difficult to navigate for a user due to reversal of situation, instead of plethora of items under category label, one is confronted with multitude of categories.

An interesting aspect in categorization efforts is the challenge of combining the common organizational knowledge with Web services descriptions. Lets consider this topic in a reference to an example of information system populated with various entities.

4.6.3 Open issues of LCV building and application

As mentioned the cost of the solution must be at all times controlled as it is one of the differentiating traits with already available solutions. Therefore, as observed in data coming from the experiment on Web services harvested from the open Internet the changes made to the descriptions itself had to be limited. First of all, descriptions

take into account the names and the available documentation. In the first step an automated indexing is in place presenting a user with a set of terms related to the Web service operations. In addition, terms available for markup are supplied by domain documents. Second phase of the process is tailoring an available name to the proposed structure. This tailoring is heavily dependent on the raw material. Some available descriptions are really well prepared and tailoring them to the envisioned schema requires only repositioning already available elements. Dealing with a more complicated situation, a supplementation of data to the description must occur with heavy use of domain marker terms. In edge cases a complete description must be prepared with no or very few data available. It can by only accomplished by its creators or previous business owners as only those people have sufficient knowledge on its purpose.

An interesting extension worth pondering is attribution of marker terms to arguments and outputs of any given Web service operation. This could map available data types and messages with entities from business user reality. Unfortunately, this has many possible caveats as business objects can be expressed in a variety of data types. Lets explore a simple output of an operation such as a report. A report can be realised as a array of strings, a single string or a more compound object wit a set of data elements that store various metadata. This compound data structure can differ greatly in terms of a number of used elements and their concrete usage. This results in a mapping that is partly useful as it allows for markup of actual output yet hinders any actions in terms finding similar Web service operations basing on realisation of a markup term by one of many possible data structures.

As demonstrated above a schema of solution capabilities is to be drawn along with information on the scope of depicted mechanisms and their influence on the solution including discussion of their automation.

4.7 The functional Web service description structure

Realising the previously defined postulates and respecting the enlisted constraints, the outline of the model emerges. First of all, any operation serves some purpose, thus it is being expressed by a verb term that denotes this action. Later a context for this action is given with additional details that can be indirectly related to the operation. By a context one is to understand a set of targets and additional clarifications.

A general pattern of description is based on the form: do something - to what/whom - in what manner - with some level of some crucial indicators. It can be presented in a more structured form as:

A Web service operation $\langle(\alpha, \beta, \gamma), \mathbf{nfp}\rangle$

- α – action
- β – object
- γ – action-object supplement
- **nfp** – vector of NFP and its values

As the first two parts are fairly obvious stating the target and the action, the third clause is the most interesting one as it can vary a lot across different domains. To differentiate the third clause from the first two one is to make an assertion that both action and its target must be a single term (with exception of being able to use a compound term). On the other hand, third clause can contain a number of terms from the Locally Controlled Vocabularies in given organisation. Each term is automatically branded with information on what namespace it originates. For Web service operations that span across different systems and Sub-organisational Units it might contain terms annotated with foreign namespaces.

When one is to examine already existing Web service operations that are invocable on the open Internet he is quick to observe that the underlying pattern proposed earlier is readily available to some degree. This was an exact source of inspiration, a description method that does not completely break the available efforts, but tries to organize them using a common structure and method for description.

The decision to design it in this particular way is derived from the prevalent lack of purpose statement in the majority of the surveyed initiatives. The purpose statement should be understood as a method of answering the question of what a given Web service operation does in a given context. One can argue that a Web service operation name shall convey this information, or even that a definition of Input and Output values in an ontology used throughout the organisation is sufficient.

Unfortunately, one cannot agree with this due to the fact that names are often poorly defined and that the connection between the purpose of a Web service operation and its input and output parameters is somewhat remote. Even if when the Web service purpose is stated as a goal encrypted as a series of references to an ontology one can easily check that such definition is unfathomable to an average business user [Klusch and Kapahnke, 2008].

Instead, the phrase-based description builds on the federated effort of SUs that should possess sufficient knowledge to catalogue their Web service assets with terms

and phrases deemed most suitable in their context. The process of cataloguing is semi-automated as the model is able to foresee tools that should accept a number of documents, which are to be treated as a reference material to obtain an initial list of important terms that might be included at a later phase.

When accomplished, Local Controlled Vocabulary (LCV) serves as a master list of terms and phrases in a given SU. Web service operations are described with terms originating from LCV. It is very important as there is no guarantee that, when any given SU is preparing its LCV, a number of terms or compound terms used in descriptions will not be repeated. In addition, namespaces allow for customizing the results of Web service retrieval and mapping of terms across an organization.

A syntax of the phrase-query language is given in Table 4.2 in the form of Antlr [Parr and Fisher, 2011] grammar, where actual terms used to denominate phrases and namespaces were substituted by exemplary ones in order to make the syntax brief.

Table 4.2: A syntax for the phrase-query language

```
grammar phrase_query_grammar;
phrase_query: (a b g) nfp*;
a : 'a:' namespace compound_term;
b : 'b:' namespace compound_term;
g : ('g:' namespace compound_term)+;
namespace : '#' ('aaa'|'bbb'|'ccc');
compound_term :  term+;
term  : 'aaa' | 'bbb' | 'ccc' | 'ddd';
nfp : nfp_el ':' val;
nfp_el : 'nfp1' | 'nfp2' | 'nfp3';
val : sign number;
sign  : '+' | '-' |;
number  : digits '.' digits;
digits  : ('0'..'9')+;
```

The visualisation of the application of proposed functional description structure is given in figure 4.2. There, 6 exemplary Sub-organizational Units were given along with their respective namespaces. As one can notice, all SU are enfolded by the Non-functional properties (NFPs) which are common for the whole organisation. This

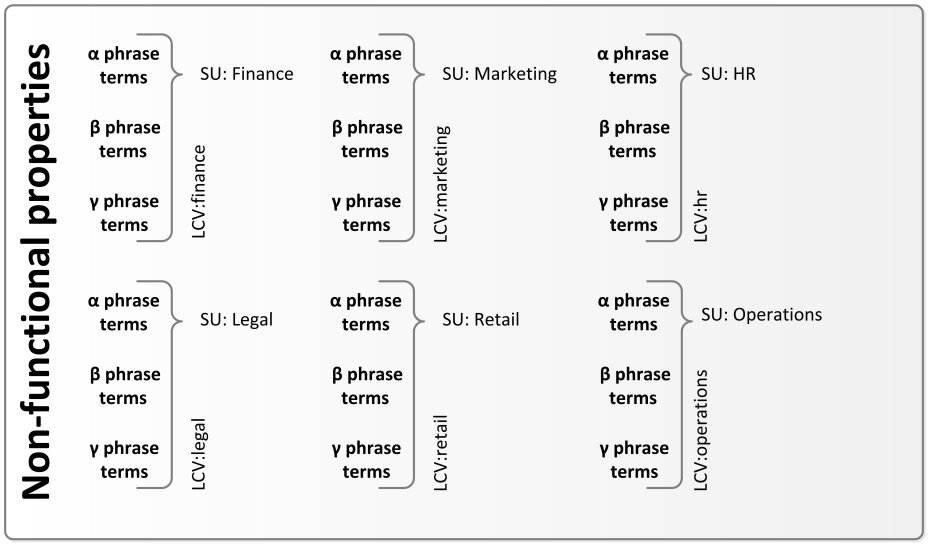

Figure 4.2: Functional Web service description structure

allows for easier comparison of performance of various properties. As previously explained, namespace warranties that terms can be disambiguated.

The introduction of the functional Web service description structure allows for an improved retrieval of Web service operations. Thanks to the highly efficient data structures and result caching it is possible to implement the search for Web service operations described with specific terms so that it is capable of handling tens of millions of Web service operations descriptions.

The high-level description of retrieval with phrase queries is given in Figure 4.3. As one can see, the final output of the phrase query retrieval is a list of Web service operations where user provided terms were found. It is ranked, as when no namespace is provided, or not every of phrases were used, the model includes additional data on usage, Sub-organizational Units and general popularity.

4.8 Overcoming Sub-organizational Units boundaries

Vocabulary changes everyday, either by a process of forging new terms and forgetting the old ones or by transmuting already existing ones with new meanings dependable on a usage context introducing therefore new levels of ambiguity [Kirby, 1998, Smith, 2004]. Therefore, a set of requirements must be met in order to make this task feasible

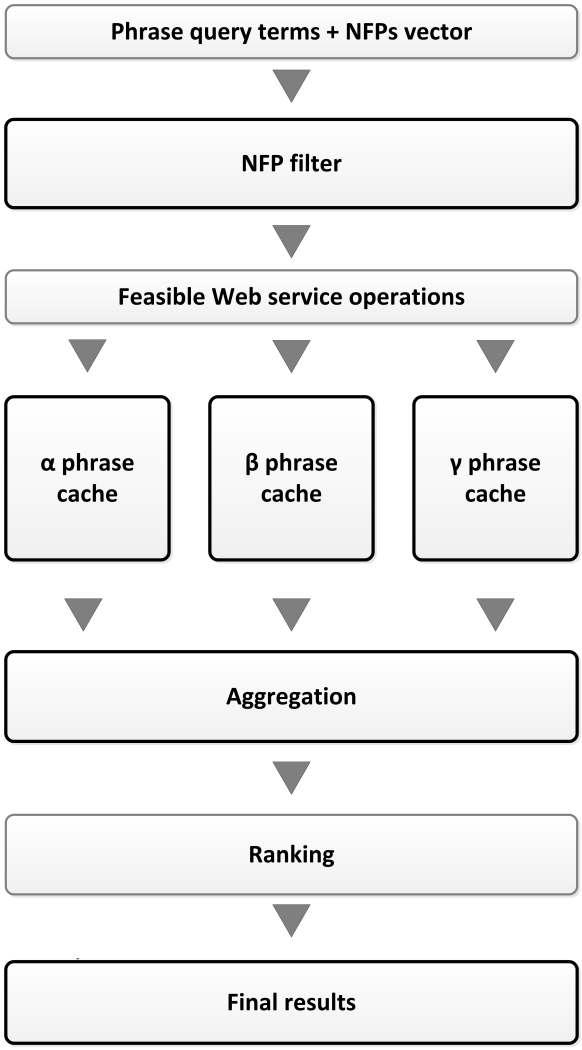

Figure 4.3: Phrase query retrieval overview

at all and additional requirements must be introduced in order to make the model capable of effective implementation.

As presented in section 2.4.1 there is a large number of available resources that can aid in this quest as reference material. Nevertheless, experience obtained by the domain while delving into semantic annotations and augmenting the traditional term retrieval approach, underline the need of special measures that help a user when searching in previously undiscovered repositories.

Properly described Web services apart from previously described dashboard advantages allow for informed choice when one or more Web service operations has the same functional description. A set of solutions was proposed in a variety of publications [Conti et al., 2002, Looker et al., 2004, Lee, 2008], due to the specifics of organization standards the values provided in all non functional categories shall be

treated as trustworthy. If Web services are provided by third parties this should be also holding as one cannot imagine allowing for Web services for production without quality control.

This solution must not strive for being a help for automated composition enabler yet it might be an aid in some situations where semantic technologies. Especially where the semantic description allows for stating a Web service goal. This shall be handled by extending ontology by assorted vocabulary items, phrases and actual descriptions. Varying on a chosen solution it shall be implemented as classes and actual implementations of these.

Thanks to the devised simple human-processable structure the cost of description should be held in reasonable bounds both in terms of learning curve and cost being a derivative of time spent on this process. On the plus side users are being handed a solution that uses a vocabulary corresponding to their work objectives organized by another user with the same background thus, providing a two plane support structure for their queries. Equipped in such a tool a user should gain better insight in a repository of Web services and their operations that allow for completing his objectives without the risk of suboptimal decisions resulting from lack of information.

The provision of linkage of user queries with actually chosen Web services and already available documentation parts and functional descriptions should allow for building a mappings between federated vocabularies that might additionally point users to other possibly interesting Web services.

The above is important addition to the mechanism of providing an answer when a user issues a query that contains one or more terms that are not present in any of the available Web service operation descriptions. The envisioned mechanism operates on the broadly understood notion of similarity. The similarity here, as it is impossible to decide whether one entity is similar to other without human intervention, is based on data available in the knowledge resources and algorithms examining the close neighbourhood of a given term.

Such functionality is delivered by the Local Context Anchoring that integrates available knowledge resources and processes them in order to deliver a list of suggestions that might cater for user's need. The detailed description of its functioning is given in separate section of the proceeding chapter.

Without such a mechanism, the model would be not complete as it would only enable its users to navigate the resources using dashboards and semi-automated search mechanisms restricted to the set of terms used in the total of LCV for a given organisation. While it would yield a perfect coverage of the assets, it would also make a

user become acquainted with all the Local Controlled Vocabularies what is counter-productive and would decrease the usability of the solution implementing the postulate model.

4.9 The model's application scenario

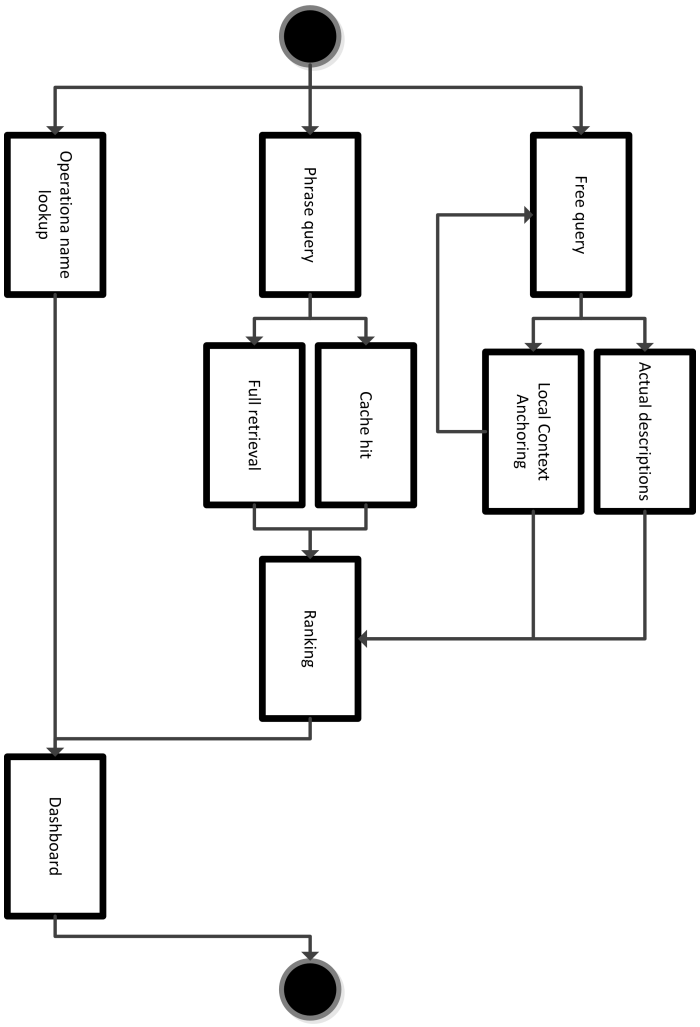

Figure 4.4: An overview of the flow through the main mechanisms supporting the proposed model

The described elements of the model can be organised as is proposed in Figure 4.4. There are basic three usage scenarios that can be interwoven in the final solution implementing the proposed model.

First of all, a user can apply the solution as a platform providing a set of overviews based on the data connected to Sub-organisational Units and their namespaces along

with additional data on usage of there represented Web service operations in various projects across the organisation. This is by far, the simplest manner implementing the classic notion of a directory known from the times of UDDI. However, thanks to connection to data on usage in projects, additional labels, data on deployment and associated SLA and QoS it is far more informative.

Second way of usage is supported by the mechanisms allowing for fast filtering of Web service operation based on the use of the proposed functionality description structure. It is relatively easy to build data structures that can provide extreme effects when it comes to dealing with tens of millions of Web service operation descriptions. An overview of proposed strategy for retrieval was given in Section 4.7.

Third way is addressed by so-called free queries, the idea and general description was given in Section 4.8. It is a key method, when searching for Web service operations providing some foreign to the usually used namespaces functionality. It is the universally important, as without it one would have to manually review the available catalogue of Web services, or try input a number of terms that could match a necessary functionality in his opinion.

4.10 Model's summary

The postulated solution positively influences organization adaptability to new business opportunities. Any venture that is to be launched as a response to the business environment changes is reinforced by a structure that holds organizational knowledge on Web services that can be readily available in new or re-engineered business processes. In addition, using the solutions' capabilities to grow with organization and build new mapping between already existing markup terms and phrases allows for integrating newly prepared Sub-organizational Units with the rest of the organization. This is of importance as it prevents the search space of functionality to become fragmented and annihilating its primary objective which is answering questions on existence of some particular Web service operation that serves a purpose defined in solutions description format.

The model does not neglect to include regular business users and executive users. It empowers them by provision of tools that should make their decisions easier and reaching their goals faster without unnecessary effort. Such decision is reflected in lower cost of the model's implementation when compared with semantic annotation oriented solutions. The Local Context Anchoring[3] can outdo the semantic solutions in

[3]Described in detail in following chapter

some situations, as it employs strategies aiming at replicating the associating abilities of human (within achievable boundaries). The general effectiveness of the solution implementing the model should be much higher than that of classic IR based solution as a Web service description has special structure that can be used to benefit users in a number of described ways.

The model addresses KRA of scalability, scope and purpose statement as well. The scalability is handled by early inclusion of data structures handling caching functionality. What is more, the functional Web service structure is designed in a manner that allows for quick decision whether a Web service operation description satisfies a user's query. The scope is also implemented in the functional Web service description along with a number of additional data available from other perspectives. Its inclusion in the core description representation structure, greatly influences the performance of query resolution. Finally, the purpose statement is finally a first class citizen thanks to the presented design choices. A Web service is described in human-like language that is to be ready to follow and recall. The details of applied strategies are given in the following chapter.

Chapter 5

The designed mechanisms

All of the mechanisms described here are detailed extensions of the goals of the model as defined in the previous chapter. Every mechanism might have a positive effect on boosting the level of compliance of one or more Key Requirement Aspects.

Subsequent groups of the mechanisms are unfolded to give an insight into their functioning and their role in achieving the desired coverage of the five Key Requirement Aspects by the postulated model. In order to present all the mechanisms to the fullest, where necessary, they are accompanied by the experiment results and data presenting their robustness and the area of application.

To begin with, one should focus on the most important aspect of the postulated model that is its new way of the description based on the three phrases positioning a Web service operation functionality in terms understood by SU and its members, along with the possible mappings onto other SUs and their idea of expressing functionality.

This is addressed in the first section that covers a number of mechanisms devoted to:

- preparing terms for description,
- choosing resources that enable semi-automated extraction of relevant terms,
- auxiliary actions aimed at higher quality of description.

Following that, a core set of mechanisms designed to efficiently retrieve functionality even when a user provided ambiguous requirements (stated in form of a free query) is presented. As given in the previous chapter, it is handled by the Local Context Anchoring.

This mode of presentation converges with the three model usage scenarios introduced in Section 4.9.

The research on the mechanisms introduced in this chapter, was organised based on:

- results of experiments concerning specific aspects of the designed model,
- designing and verifying of the algorithms responsible for the model's functioning,
- reviewing a number of case studies crucial to the goals of the model.

All of the above listed, led to a presentation of any given mechanism in the context most crucial to its application.

5.1 The designed mechanisms supporting description of Web services

This section focuses on the steps of preparation of the fully functional Local Controlled Vocabulary of a single Sub-organizational Unit.

5.1.1 Amassing the relevant terms for the LCV - shortlisting phase

The process of preparing a three phrase description is started with a phase of term selection. It is a semi-automated process where a designated person or committee compiles a set of terms that are the most significant for the given Sub-organizational Unit. For the brevity sake, lets denominate as the Supervising Entity (abbreviated as SE). These terms are the frame of reference for any given Web service operation's functionality and an agreement must be made among SUs members to recognize selected terms as the right manner of referencing to any desired functionality.

This phase is achieved by accumulating a number of documents concerning Sub-organizational Unit where a technique based on term frequency inverse document frequency (TF-IDF [Ramos et al., 2003,Robertson, 2004]) is to be applied to present a shortlist of the most significant terms. In addition, the Supervising Entity should surely extend the shortlist by a number of terms that might be not present in submitted documents yet hold value for the whole SU. Removal of terms that do not hold importance for this SU also occurs at this phase.

Important deviation from the ontology and database modelling is the federational and parallel nature of this phase. Ontology designers and database architects stand

before a task of modelling a total image of some aspect of an organization or world. Thus, a great deal of expertise and skill is needed to propose a workable result that satisfies the most interested parties with the least amount of compromises degrading models robustness as a foundation for desired applications.

On the other hand, the Supervising Entity is devoted to a single well known fragment of the given organization. It strives for the most complete list of terms that should be used to describe functionality and while pursuing this goal it does not focus on the external entities. This shortlisting takes place in other SUs and it is perfectly feasible to enable them to work simultaneously.

The obtained shortlist enriched by SU, is a product of the first phase. It is a cornerstone of the functionality description space that is used by its users to firstly describe Web service operations and then relate them to other ones.

5.1.2 Shortlist partitioning

The second phase of the process consists of making decisions on which terms should become one of the three phrases available for the Web service operation description. As mentioned in the chapter devoted to the model itself, the three phrases model a situation, when information is given by pointing to three aspects (The NFP vector is discussed elsewhere):

- action,
- object,
- action–object supplement.

This triple is to be referenced respectively as α, β and γ elements or a phrase description as a whole. The first two elements are compulsory and the third one is optional and it can be constituted of a number of terms eligible for an action–object supplement. Whereas, α and β elements must be expressed with a single term from the available shortlist of eligible ones.

As remarked in earlier parts of this work, this structure references some of the most common practices of enhancing actual data with metadata such as in RDF triples. It differs in its core objective to remain human readable and human processable in terms of fast evaluation of Web service operation's functionality.

In addition, readability and human processable form is in a lockstep with certain rigour on the description. This rigour realised in other solutions is referenced as a

description schema. Ontology designers use a number of schemas related to particular namespaces to achieve homogeneity and correct structure of entities built with concepts originating from their efforts [Jarrar et al., 2003].

Here, rigour should lead to reorganization of the previously used methods into one, common for the whole organisation. The same structure is the cornerstone of later described mechanisms allowing for mapping terms used in Web service operations' descriptions (implemented as LCA).

The α phrase is the first phrase that also answers to the question of what is being done. As noted, there are no situations where only the α phrase is used for description. It always act in conjunction with the second phrase which denotes the object of the action.

The β clause contains the most important actor of the description. Combined with the α phrase it answers fully what is being done to which object. Thus, a minimal description can be built. As an example one could provide a description in a form *send notification* (namespaces are omitted on purpose). Unfortunately, such description is not very helpful.

Therefore, the γ phrase was introduced to store supplementary data on the interaction of the main object with other entities. Its goal is to add all the necessary supplementary data that should make the descriptions fully comprehensible to its users. The example used above can be extended to such form: *send notification user maillimits*.

It is essential that, each term can be used more than once in the whole description. In addition, terms can be combined into new ones when the relation to building blocks is proportional to the length of combination. Combining two base terms results in a Web service operation to be related equally to two base terms.

5.1.3 Description sources

As reported earlier the source for description cannot be perceived as a uniform entity. This is due to the fact that, the traditional Web service description originate in the code comments of routines that are being shared with Web service technology. This code comments are a result of some requests issued by some team or project so that some business objective can be achieved.

To the author's best knowledge, there is a lack of studies measuring the percentage of actual business vision that is stored in comments for the routines that carry the

vision out, yet thanks to enormous amounts of available source code accessible for revision one might suggest that the actual percentage is very low.

Even when one is to consider enterprise based code, a broader business vision is not stored as a source code comment but as separate documents that archive the architecture of a system and all the vital interactions on various levels of its design.

Therefore, when one is to consider the actual Web service he can divide the sources of description in roughly two discrete channels:

- direct sources:
  - Web service operations' names,
  - operations' descriptions,
  - additional commentaries.
- indirect sources:
  - companion documentation provided by application architects,
  - requirement analysis provided by appropriate analysts,
  - actual usage in achieving business objectives.

As remarked, the value of direct sources is of mediocre vale at best. A study performed throughout research activities that resulted in this work demonstrates that presence of commentaries and documentation is not to be taken for granted. The quality of available documentation when available is very varied. Ranging from informative to cryptical and completely unhelpful. A person or team of people using data gathered by Supervising Entity and facing a task of description with the help of direct sources has a very difficult task. Especially that, the actual names of Web service operations can be difficult to interpret.

To emphasize one should review an exemplary data on Web service operations gathered in course of research in Table 5.1. It is beyond any question that in order to come up with a robust and effective system any team or individual performing a description task should refrain only to direct sources. The referenced data demonstrates that even when the operation's name is accurate one can only guess its true functionality, which cannot be accepted in terms of the goals of this work.

5.1.4 Description utility based on the real world Web service corpus

To gauge what is the level of utility of the Web service description in WSDL documents an experiment was carried out on the amassed body of Web services from the

Table 5.1: Exemplary data on Web service operations' names.

Some names of operations from Web services that constitute the research corpus
DNAYP_Freelist_Report_update, DNAYP_Freelist_Report_updatecompany, DNAYP_GetEmail, DNAYP_NOTFOUND, DNAYP_ReportSearchWord, DNAYP_Website_Feedback, DNAYP_validateuser, ExecuteTask, FetchOneCompany, GenerateVCodeImage, GetCompany, GetCompany1, GetCompletionList, GetLocation, GetPatternCoEfficient, GetProdList, GetProduct, GetProteinInfo, GetProteinPairPatterns, GetProteinPatterns, GetVersion, SMSEMAILToClient, SMSEMAILToVisitor, SMSVFS, SearchPatternAtOnce, SearchPatternSeperate , SendEmail, SendEmailWithBcc, VerifyEmail, getdataset, getmaxid, premiumMemberDelete, premiumMemberInsert, premiumMemberUpdate, premiumMemberUpdateclick, premiumMemberUpdaterank, ContactUsSendEmail, DNAYP_DisplayReport, DNAYP_Freelist_Report,

Internet. All Web services were categorised in order to come up with those that are in English and can be further processed.

Originating in the open Internet, a total number of 22456 Web services deployed as the asmx applications [1] were retrieved. Nearly 2000 addresses were identified as not viable for further checkup. Those eligible were tested whether there is a possibility to retrieve the WSDL documents. From the research undertaken it was found that a total of 9000 Web services was further viable for examination. This number does not include a number of Web services whose description contains non English passages.

As the general solution is prepared for English resources, in terms of ontologies, semantic nets and thesauri one cannot properly resolve the domain without access to the additional portion of information available in the documentation strings. In addition a total body of Web services described as WSDL and retrieved throughout previous research activities is larger. The number of obtained WSDL files exceeds 56000 Web services with over 800000 operations. Unfortunately, Web services manifested as WSDL documents are rarely a good research material due to the lack of actual binding points. One can say that these WSDL documents serve mainly as a reference whereas the asmx deployments are operational Web services.

Another problem with countable Web services is that due to the fact that when they are deployed with other technology there is no possibility to find them when deployed did not prepared resources with direct links. This observations make one realize that numbers given are only approximation that for sure must be multiplied

[1] Numbers available after analysis of data gathered throughout author's research activities

by unknown factor. One should also remember that the number of Web services to be found in the open Internet does not correspond to the actual number of Web services being in use in enterprise world. Trying to reckon on the actual number of Web services used can only lead to daring approximations that may be fuelled by observations and law of large numbers (applied as in [Barbour and Luczak, 2008]). From the corpus of eligible Web services additional services were filtered in order to come up with a body of Web services whose operations are well documented.

The total number of filtered Web services was 9000 with a total number of operations of 120000 and the average of operations per Web service was 13.13. 60% of the Web services had 1 to 5 Web service operations. Over 80% of Web service operations had defined only one input parameter of complex type (with at least 2 elements).

66% of Web services did not have any additional description characteristics (comments or documentation strings). The average length of available comment sections was 600 characters, the average length of documentation strings was under 300 characters. The quality was checked by random sample of 100 Web services that had some comments and documentation strings. The quality checked focused on discovering whether available documentation/comments provided informative description on Web service functionality and purpose. The results were very poor. It was estimated that only 20% of the total body of checked Web services with commentaries and documentation in a WSDL document could help in description efforts.

5.1.5 Post-partitioning mapping of SU local Business Objects

Due to the unhelpful nature of direct sources as described in the previous section there is now an obvious need of inclusion of the indirect sources into process of description. This is to be achieved by mapping of terms obtained thorough the partitioning phase to business processes, business objects and vital IT systems supporting them. In comparison to other approaches, one has once more highlight the locality of this solution in terms of the whole organization.

Selected terms were consciously taken from the business environment of a SU that is being in process of description of its Web service assets. Therefore, a mapping between knowledge available in organisation and terms that should be the building blocks of the descriptions allows for introduction of the core entities that should be referred as either objects (in β phrase) or supplements (in γ phrase) and connected with action terms (represented as members of α phrase).

As another difference from other solutions, one must take a close note on the fact that the resulting graph is not the final structure in terms of which all descriptions should be organised but a tool for mere help for those tasked with describing the SU assets. Its primary objective is to present a framework of reference.

Please consider that, in order to provide interested parties with stable framework the earlier mentioned mapping of knowledge must reach to indirect sources. A challenge of knowledge transfer from employees to structures that make it possible to store it beyond the career span of any single employee is well known and there are many possible scenarios to address it [Nonaka, 1994, Nonaka and von Krogh, 2009]. It is important to highlight that an organization, which is willing to apply a solution at all, should comply with the requirements stated in the previous chapter, as given in Section 4.3.3. The most important of those, in this context, is the organization culture that may and will enforce knowledge retention in order to sustain vital business processes. Conforming with these requirement equips one with a rich repository of indirect sources that should make the Web service operations description more accurate.

This method is used well beyond the currently discussed effort of building a reference framework. A situation when a person or a group of people is in need of quick realisation of the key elements of infrastructure arises many a time. The best example is the external support of Enterprise Resource Planning systems [Leknes and Munkvold, 2006]. Any maintenance or extensions are dependent upon smooth knowledge transfer. This transfer is realised in many ways but the most usual form is the documentation describing an IT infrastructure landscape. All beyond documentation is stored either in separate documents that might be withheld on purpose or by omission, or in individual specialists. Therefore, in case of unanswered questions there is a need for further communication in order to retrieve necessary information.

The value of official documentation is very varied, it ranges from good reference to overall status quo to unwieldy artifacts unusable in any context. The overall trend in managing IT infrastructure is emphasizing the obligatory nature of the documentation without underlying the necessity of introduction on an overall picture of collaboration of various IT systems.

Therefore, a mapping between key business objects and terms retrieved is of utmost importance in description.

The actual graph representing all relations among various Web services is a result of accomplished work of the SE augmented with mappings across sub-organizational boundaries along with dynamically retrieved mappings being a result of solution use.

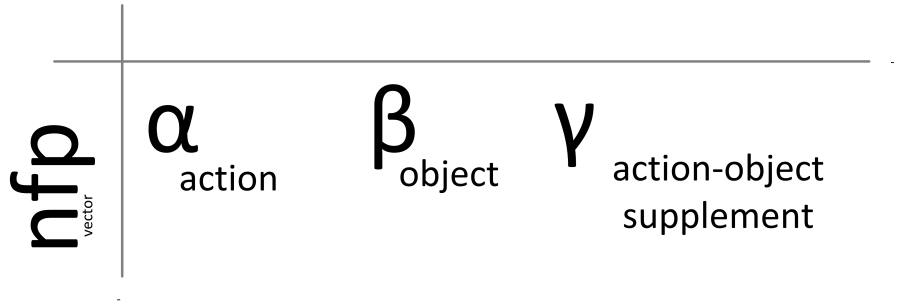

Figure 5.1: Functional Web service operation description structure

It provides a supporting structure for the overview/dashboard functionality, and is usable while ranking Web service operations in the rest of the envisioned usage scenarios.

5.1.6 Web service operation description

Being equipped with a set of partitioned terms with references to actual business objects of interest to SU in question an actual description phase follows.

For the purpose of discussion an exemplary set of partitioned terms along with references to business objects relevant to SU are given. The summary of business objects is available in further referenced Table.

One has to emphasize that in this example all of the entities are common for the whole organisation. As a description phase is viewed from the perspective of single SU, a set of partitioned relevant terms is given in table 5.2. Listed terms are related to a SU concerned with human resources management. As described earlier, there is no requirement that actual SU employees have to prepare descriptions, yet communication with them and actually description entity must exist (fulfilled by the SE, even when the actual SE members can originate from other units of the organization).

Let's envision a popular organisation structure where the whole IT infrastructure exists in a separate unit and serves other units based on demand. Therefore, SE is almost certainly a part of the IT unit. There is no guarantee that original developers of Web services are the ones describing them. Nevertheless, the SE has access to actual code, documentation, gathered partitioned set of terms, reference framework and original requests and all data relevant to it.

Thanks to this data, a description being prepared does refer to actual Web services being used by some particular SU (human resources in this context) with terms

Table 5.2: Set of partitioned terms relevant to exemplary SU

Actions
create, post-to, select, restart, stop, start, publish-to, add-to, get-from, set, report, validate, reject, delete-from, insert
Objects
user, database, server, login, report, usage, Project, statistics, Expense, timesheet, application
Action-Object supplements
Amanda, Flora, Accounting, Mercury, user, database, Business-Owner, application

Table 5.3: Exemplary Web service operations from HR Web service

Web service operations
ExpenseSelect, TimesheetApprove, TimesheetReject, ActivityTypeDelete, ExpenseReject, EmployeeSelect, Test1, ProjectDelete, EmployeeInsert, ExpenseSubmit, ProjectSelect, TimesheetSubmit, ExpenseApprove, ActivityTypeInsert, TimesheetInsert, LoginValidationSimple, ExpenseTypeSelect, ProjectInsert, LoginValidation, ExpensePay, TimesheetDelete, ExpenseTypeInsert, TimesheetSelect, EmployeeDelete, ExpenseDelete, ExpenseTypeDelete, ExpenseInsert, TimesheetPay, ActivityTypeSelect

Web service operations' names from a Human Resources domain serving and processing data on work record

gathered and rectified by SE that warrants its relevance to actual business objects, needs and users is prepared.

Accomplishing this stage allows for the first review of Web service operations in terms of their interrelations based on one of the three phrases and terms used to represent business objects. This is actual realisation of one of the postulates that are to help a business user to locate Web services of interest in natural manner (the mention overview/dashboard usage scenario).

As a note, one has to write that terms used for phrases can and should be transformed in a manner that makes them more appropriate for the end user. The examples used in Section 5.1.2 should benefit greatly by simple customization of the terms used such as bonding the informative preposition to the action term.

Table 5.3 gives data obtained from one of the gathered Web services, it is from the discussed exemplary domain of Human Resources. Data in the table consist of all the available Web service operations. There was no additional documentation whatsoever.

Combining data in tables 5.1 and 5.3 one can describe Web service operations with discussed three phrases in a manner given in the following examples:

- $Ws_{f_1} < \alpha, \beta, \gamma >=< Get, Expense, (report, Accounting) >$
- $Ws_{f_2} < \alpha, \beta, \gamma >=< Insert, Expense, (Type, Accounting) >$
- $Ws_{f_3} < \alpha, \beta, \gamma >=< Get, Activity, (Type, Mercury) >$
- $Ws_{f_4} < \alpha, \beta, \gamma >=< Delete, Employee, Amanda >$
- $Ws_{f_5} < \alpha, \beta, \gamma >=< Select, Timesheet, (Empoyee, Amanda) >$

The examples assume that Amanda is a name used to denote application responsible for tracking of HR data, Mercury is the communication system and Accounting is a system responsible for management of financial data.

5.1.7 Summary

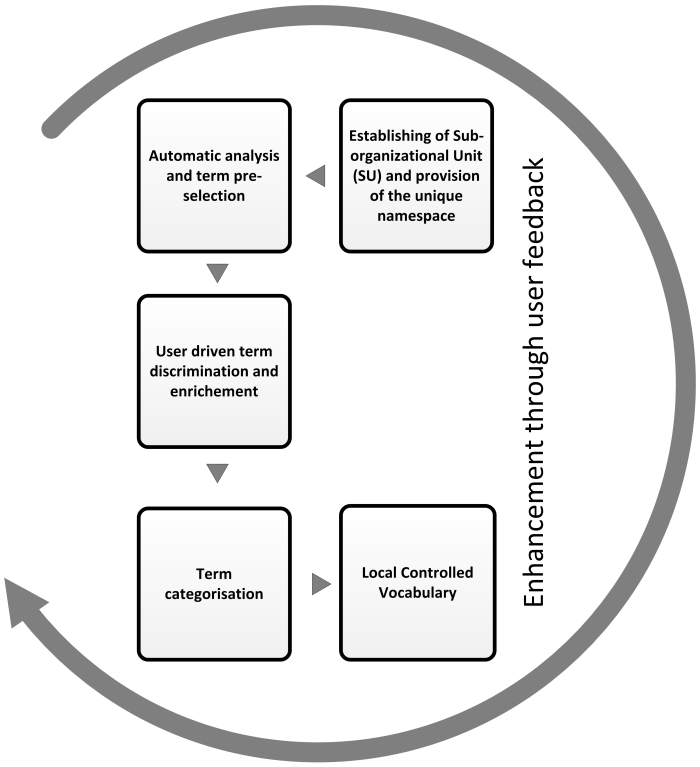

Figure 5.2: Steps necessary in preparation of term repository for functional description of Web service operations

As assumed in the text, LCV is assembled in a semiautomatic manner with final decisions on term inclusion left to delegated user or users.

The whole process can be repeated due to the fact that an organization while growing acquires new systems, employees and vocabulary. Therefore, various phases can be revisited by the SE and new terms can become part of the LCV. The overall depiction of the process is given in Figure 5.2.

5.2 The designed mechanism supporting functionality retrieval

5.2.1 Query matching

The search for Web service operations is made fast in comparison with a fully-fledged ontology description thanks to a solution acting as an automaton, which allows for rapid narrowing of the set of the feasible Web services. It has 4 ways of narrowing the search space, as there are three key phrases that can be of interest to a user and a forth one, which is devoted to NFP vector.

The simplest manner in which one can narrow the search space is with starting with the NFP vector. The operation of narrowing is just a simple filter on available Web service operations that can be augmented to the search of Web services where every operation has some desired trait and a value of this frame matches the filtering query.

This is especially important when a high level overview has to be delivered so that decisions can be made, which are in scope of key business objectives - boosting delivery of some type of services, revoking this type of services, changing scope and terms of agreements concerning quality and warranties. To developers this information can be of small importance, yet for the business users and executives this is a real benefit as its direct mapping of business concepts to Web service operations.

Addressing the three key phrases of Web service operation description (α, β and γ) is achieved thanks to inputting one or many possibly desired terms into the automaton. As all possible terms are handled by the envisioned system for some given namespace, and additional terms from other namespace are also reachable, query operates on the cache in form of hashes that contain terms as keys and Web service operations as values [Li et al., 2004]. A situation traditionally viewed as a key collision is perfectly acceptable one, as a list of Web service operations is a valid value and an algorithm can continue so that a query yields at some time a possible solution. The results yielded by this stage are to be stored in a data structure with traits of

a set. With this assumption, a set arithmetic operations should lead directly to a production of outputs satisfying requirements in a efficient manner.

Stage results are combined by an aggregation procedure, which reduces outputs from the previously obtained phase in order to single out Web service operations satisfying all of the requirements obtained from the user defined query. If assumption on set traits of the outputs holds, this part is satisfied with an operation of a set intersection, therefore efficiently computed even for large data sets (with the same assumptions as presented in [Baeza-Yates, 2004]). The pseudo-code that accomplishes the task in described manner is given in section 5.2.1.

Pseudo-code of the matching algorithm

Algorithm 1 The algorithm implementing query matching on the available Web service operation description repository.

C_α - α phrase cache
C_β - β phrase cache
C_γ - γ phrase cache
Q_u - generated user query
Q - all generated user queries
for all $Q_u \in Q$ **do**
  αQ_u, βQ_u, γQ_u, $nfpQ_u = decompose(Q_u)$
  $C_\alpha = filter(nfpQ_u, C_\alpha)$
  $C_\beta = filter(nfpQ_u, C_\beta)$
  $C_\gamma = filter(nfpQ_u, C_\gamma)$
  $wso_1 = match(\alpha Q_u, C_\alpha)$
  $wso_2 = match(\beta Q_u, C_\beta)$
  $wso_3 = match(\gamma Q_u, C_\gamma)$
  wso $= wso_1 \cap wso_2 \cap wso_3$
end for

5.2.2 User feedback role

The choice that users make while reviewing the available Web service operations or searching for them using either phrase-query mechanism or the LCA is used to enrich the pool of data available for further processing. As was discussed earlier this is especially relevant in the ranking activities.

A similar mechanism is used in search engines, where every click is monitored and information on relation of query terms and one of the results is captured and used for

further optimization [Baeza-Yates, 2004]. The real effects of this method are visible in conjunction with growing number of users and their queries.

There is also a possibility for a user to mark a Web service operation as similar to some other operation. This is valuable data as it is added by an actual prospective Web service operation invoker. Data gathered in this manner serve to present recommendations in the search list. Whenever a Web service operation mapped with other Web service is available, a suitable information should be presented to one issuing query.

The real potential of this data source can be leveraged, when an organisation decides to prepare a global map of systems, applications and other resources. This map can be used as a common place for reference of terms defined in various SUs. Such global map becomes then a global framework.

When an organisation imposes that terms gathered and bundled in LCVs by their respective SE are mapped a new tool emerges. It can aid in translation of various terms across the organisation. Of course, the perfect translation is not possible due to the reasons stated when discussing ontology modelling. Yet, a global framework providing a few hundreds crucial resources is cross-referenced with LCVs of all the Sub-organisational Units a new plane of Web service operation retrieval is available.

In addition, described mappings allow for deeper exploration of the concept of similarity of Web service operations' descriptions. As with prior mechanisms, a restriction for phrases must be bestowed. Addressing the α and β phrases should yield immediate results, whereas addressing γ imposes employment of more elaborate heuristics. The similarity heuristics rates higher descriptions being related to the same object or an object that can be translated as the same one, then an action is evaluated and finally action-object supplement is weighted into final result. Such order is favoured as it derives from the overarching structure of Web service operation description.

5.2.3 Synthesis of a Web service description

A Web service description is made up from a number of individual descriptions of its operations. Therefore, one has no single description, but a compound one aggregated upon certain criteria. This greatly emphasizes the role of Web service operation. Other initiatives such as OWL-S were oriented also on single Web service operations by imposing requirements on the number of operations in described Web services and limiting them to one [Martin et al., 2007].

Stating the above, a question of a Web service similarity is more complicated than before as this artificial description can be processed in many ways depending on what data is to be exposed at the forefront. With addition of mechanisms presented as a part of the postulated solution a similarity can be extended beyond number of operations, their structure, types of data for their inputs and outputs and method of access.

Having also access to those mentioned, standard features may provide an augmentation for automated composition of Web services as a first tier support for separating possibly matching Web service operations from the global repository.

As one might be interested in implementing such a scenario to cut cost of execution of automated composition, he has to remember that it is only possible when all data is described with an ontology common for all entities being described. Taking into account the requirements for the postulated solution, this might be perceived counterproductive.

Therefore, a question on purpose of such combined synthetic description must be issued. In general, an answer provided may be the implicit close relation of Web service operations being a part of a Web service. This is very weak inkling as deployment of Web services tend to include operations that were included in seemingly random fashion[2], in order to realise some business objective. Thanks to introduced description model, single operations can be grouped basing on their action phrase or their object phrase. In addition, whether operation relates to common terms in action-object supplement, it is also of value to a user.

To summarise, Web services as bundles of Web service operations have greatest merit as an indicator of potential relation of the former. Unfortunately, as remarked, there is no guarantee that this is the case. Yet, such data should be presented in details of Web service operation, with measures allowing for review of other operations.

5.2.4 Emerging structure of Web services and IT infrastructure

Accomplishing the description and mapping the core infrastructure with the available terms used in Web service operations, a user interested in the domain is presented with a map of organization systems and functionalities offered by them. This is an additional benefit of the postulated solution.

[2]This was a case time and time again when reviewing the corpus amassed and briefly described in Section 5.1.4

It fulfills one of the promises of SOA as it empowers organisation to manage its assets and use them more effectively in terms of not wasting effort on already implemented entities.

One can easily image a situation where some designated group should audit the assorted repository of descriptions along with its relations to particular IT infrastructure elements and monitor their business owners. This audit could be used to answer whether a functionality merger should be carried out, with some other potentially equivalent or very similar Web service operation.

5.2.5 Additional description traits

Apart from the core description expressed in the phrase description Web service operation is described with additional elements. They store data on relationships with clients, business owners and various other dependencies that may seem irrelevant from the technological point of view yet they build a body of information that can be a core of a competitive advantage of an organisation.

The canonical example of this type of data is the Service Level Agreement that constitutes boundaries in which cooperation with clients is being perceived as desired. Knowing that some Web services have a certain level of some trait is important in localising potential domains of improvement.

This is an important feature that enables additional filtering mechanisms to take action against an available repository of Web service operations. A decision on introduction of this secondary traits is a step toward bringing Web service operations closer to business users. There are other initiatives such as USDL [Cardoso et al., 2010] that strive for changing focus from technical details to a Web service as a part of business environment in a variety of aspects. Preceding this trend was inclusion of Non-functional properties which enabled to structure additional traits of any given Web service in terms of some knowledge representation entity.

The solution does not postulate a brand new method of description of Service Level Agreement or Non-functional properties. It allows for introducing any of the available solutions as long as it can be used as a filtering mechanism. The implementation of the idea was shown in Algorithm 1 and in the preceding section.

Web services and their operations cannot be left unattended in terms of performance and fault monitoring in organizations that use them as building blocks of their incarnation of the SOA paradigm. Therefore, there is a plethora of data that should be used as a decision making material. While functionality is given by the described

triple of phrases, the business centric traits should be stored as a vector of pairs where first element denotes some particular trait and second element value for this trait normalized across whole organization.

A business user can issue queries that target not only functional aspects, such as the object of a Web service operation but also any of the available traits and their values. Not every Web service operation can have all the available traits as not all are applicable to every one. Nevertheless, in general it should greatly broaden retrieval possibilities offered to a business user.

Including discussed traits into Web service operations' descriptions affects also a compound Web service description as the common traits can be averaged in order to present a form of summary to a business user. This summary serves as a context for given Web service operation, directly informing whether level of given trait differs a lot when other operations are considered.

There is a wide variety of available NFPs such as general performance, throughput, mean time to failure, responsiveness, invocation time. Others are directly linked to its developers, publishers, names, versions and dates. In addition to those above, there are traits directly relating to terms covered by SLAs or other documents of this type (Operation Level Agreements or Underpinning contracts). Some of them concerning for example Web service operations related to storage can be any of those:

- Availability,
- Maximum down-time,
- Failure frequency,
- Response time,
- Periods of operation,
- Service times,
- Accessibility in the case of problem,
- Backup,
- Bytes per second,
- Memory size.

5.3 Local Context Anchoring for unmatched query terms

All the previously described mechanisms up to this point have made the description of Web service operations clearer as they refer to a set of terms that describe a well

known domain with reference to actual business objects. Combining this with ability to combine terms is a flexible method to express functionality of any Web service operation.

More, Web service operations described so, are enriched with a namespace qualifier referring to the domain of described assets. This gives a leverage in situations when mappings among different domains are to be prepared. Terms, their combinations and Web services operations can be mapped across domain boundaries.

5.3.1 Overall objective of LCA

The LCA is designed to find all the possibly relevant terms that are present in available resources. These possibly relevant terms include elements forming:

- fixed phrases and idiomatic expressions,
- collocations,
- acronyms,
- compound phrases.

In addition, all the co-occurring terms are also retrieved from the resources. This allows LCA to act as a prosthetic to natural human capability of linking associate terms into phrases or operate with synonyms. When one used for a free query a phrase built with terms that are a part of an idiomatic expression or used a set of synonyms regular IR tools might fail due to the lack of a mechanism that should try match to it. The strategy was first introduced in previously cited works dealing with Query Expansion. Nevertheless, in a context of the functionality description this is a novel approach, that leverages important observations on human nature.

LCA incurs some non-negligible cost in terms of time needed to accomplish the retrieval of viable terms and initial processing of the resources. Throughout the lifetime of a system instantiating the proposed model, these costs will be diminished thanks to the caching mechanisms that store previously obtained results.

5.3.2 Steps necessary in LCA functioning

When querying for a term a list of matching resources ordered by one of the phrases is given. When there is no match a set of hints is presented based on term references with employed knowledge representation structure.

The mechanism for Local Context Anchoring depends on open data repositories and assorted corpora. It can be viewed as a specialised Query Expansion algo-

rithm [Voorhees, 1994] that takes into account specific data. The choice to make is for the interested entity, yet there are arguments for using open data repositories. The most important is that the coverage and size of an open data repository is considerably large and covers a lot of varied topics. More, the access to an open repository is free and not restricted by hostile licenses. Available data in the both mentioned resources amount to millions of defined terms along with examples, details and additional references. This mass of data is incomparable with any other resources aside closed repositories containing scanned and processed libraries of the biggest and best equipped universities. In comparison to the open repositories their greatest value is the depth of coverage on some topic. As they are not by and large accessible to general public there will be no further discussion on their merits and other disadvantages.

Whether one would like to replace or augment the available repository with his own resources it is a matter of his decision. As mentioned, an organization can posses vast resources on some topics that are key to their objectives and cannot be matched by those available in the open repositories.

The key concept of the Local Context Anchoring is probing the available repositories for terms that coincide with those unmatched with terms used in Web service operations' descriptions. This is not to be understood as traditional measures used by Information Retrieval based on statistics of direct neighbour co-occurrence as n-grams [Brown et al., 1992]. The unmatched term is queried across available resources. When this term is matched, its context is probed for terms present in the available LCVs and actual Web service operations' descriptions. Context of a search term is understood as a text frame that spans for n terms around the matched terms. The actual number of terms is dependent upon experiments, yet research performed demonstrates that frame which matches the length of a average paragraph in English texts is good choice.

The terms that fall into frame are normalized and stripped out of stop words in order to get rid of the noise. The mechanism yields best effects when multiple matches are found and an occurrence ranking of coinciding terms can be prepared.

Apart from the open repositories, a Web query is performed to broaden the list of available data. One must remember that every query issued to the Web resources consume some time, latency of network connection can diminish positive results of this mechanism. Nevertheless, there is possibility to store open repositories or their parts locally what can be a source of processing times savings. The best strategy is to cache the intermediate outputs for future use.

While querying for the unknown terms a frequency list is prepared, where most frequently coinciding terms are at the top. In addition to simple checkup whether a given coinciding term is available additional steps are undertaken to boost the performance of the mechanism. As semantic network such as Wordnet[3], ontologies such as CyC and various thesauri are available, an additional query can be issued to find whether there are synonyms, hypernyms or hyponyms to the unknown query that are used in organisation as description terms.

Apart from the fact that a user is not left with unhelpful information that no results are found as long as there are unsurveyed possibilities of matching terms a mechanism serves one additional purpose.

It presents an opportunity to perform mapping among functionalities in other SU described with other terms. As discussed SU has perfect freedom to choose a set of terms related to their needs and business environment. Yet, many a time, a situation can occur in which some aspect of functionality was described with a term that has many used synonyms. With Local Context Anchoring one can find potentially similar Web service operations.

As there is no guarantee that entities described with similar terms have similar functionality a decision has to be made by a query issuing user. This mechanism delivers a tool that in essence allows for asking a question to find descriptions phrased similarly.

The whole mechanism can be outlined in the following steps:

- Issuing a query where one or more terms is not present in available descriptions.
- Local Context Anchoring surveys the available resources:
  - A ranking list of most frequent coinciding terms is prepared.
  - A check on the list is performed with used terms.
  - A lateral check for auxiliary language semantic relations is performed.
- Quorum is used to prune the results.
- Results with frequency greater than threshold are selected.
- Results are modified to reflect reputation of resources they originate from.
- Results are matched with Web service descriptions.
- List is terms matching the descriptions is forwarded to mechanisms responsible for further ranking.

[3]As referenced in Section 2.4.1, it applies to all the following references

5.3.3 Quorum among resources used in Local Context Anchoring

Every resource yields a number of terms. These terms are given with occurrence counters that cover the importance of each terms in later procedures. There is an auxiliary mechanism that allows for removal of terms that are not found in some number of resources.

It was designed to catch the so-called outliers, which here should be understood as terms that were classified as co-occurring yet their occurrence was limited to only single resource. The quorum for a term is a situation, when each qualified term meets quorum's target.

The quorum targets are set at a level of a resource. The rule is, that the less one trusts given resource, the higher quorum level should be. In experiments results given in evaluation chapter, the highest quorum level was assigned to the open Internet. It is motivated as a precautionary measure to filter out unnecessary noise in data used for ranking.

To exemplify, quorum level of 1 denotes a situation when terms originating from given source do not need to occur in any other sources. This level should be used for well defined corpora such as specific documentation or trusted semantic networks and thesauri. Quorum's level over 1 denotes that a term originating in given resource must be found in at least the actually inputed quorum's level.

5.3.4 Reputation of sources for Local Context Anchoring

As described earlier the LCA uses a number of sources that enable to map unknown terms on those used in the system. It is believed that a customization of the method must be possible as different sources can be trusted in varying level. What is more, this level of trust to each source can change. A person or team introducing new sources can setup a predefined level of trust to each source.

The level of trust should be mapped onto an impact factor of results from given source on the overall ranking of results. To exemplify, lets assume that one is dealing with four sources:

- Internet - σTI
- Wikipedia - σWW
- Wordnet - σWO
- Internal document corpus - σID

As the most fitting the organization needs is the internal document corpus, the trust with which it starts is set to value 1. Following that is Wikipedia, which reputation value is set to 0.8. Then, the Wordnet was evaluated as valuable due to a considerable number of available terms stored so as it is possible to efficiently obtain synonyms of a term along with hypernyms and hyponyms (trust value set to 0.65). Finally the Internet sources are trusted the least, the trust value is set to 0.5. The values given in example are arbitrary and the overall procedure reflects the notion of strong preference. This can be transcribed as: $\sigma ID \succ \sigma WW \succ \sigma WO \succ \sigma TI$

The above given values and preference order is to be treated as an initial state of the ranking system which processes data outputted by LCA. Throughout its life cycle, thanks to the users input, it can change into one that is significantly different one. This is a important feature as it enables system to learn from the signals.

As an initial state is viable of change, the system adapts itself to user requirements. LCA presents results for all the unknown terms. Every unknown term is an input to LCA, and an output is a ordered list of terms found in the available sources along with the stored occurrence frequency. Whether a term is found in multiple sources its position is an outcome of computation of values associated with it in eligible sources.

The output from various sources can be presented as a tuple of two elements, the term found in a source and its frequency. The output list is prepared as follows:

1. The list of terms from the most trusted source is used to check whether they occur in other sources as well. The final value of given term from the most trusted source is computed as a sum of products of term frequency in every source that contains it and its trust level.

2. Terms from the most trusted source are used to establish the average frequency. Next, the terms that were not present in the most trusted source have their frequencies multiplied by their trust level. If the result is less than the established average, they are discarded.

3. Those terms that repeat in various sources and weren't discarded in the previous step, are positioned by their accumulated scores established in the previous step. It is possible, that a term can appear in 3 sources, and was discarded in 2, still it is represented by the result obtained in the one that provided result above the average.

The resulting list is used by other mechanisms.

5.4 Additional designed mechanisms used model-wide

5.4.1 Compound term decomposition

In order to balance flexibility and expressiveness of the solution, it was proposed to allow for combining the terms used in description phrases whether such a need should arise. This combination might easily lead to unwanted decrease of the solutions performance as terms build with other terms should not be easily traced by Local Context Anchoring. To remedy, all compound terms must be built with a tool that stores data on the constituent atomic terms.

Having accomplished this, a reconstruction of terms is a simple procedure that lookups a compound term in designated register. Users benefit from this feature as they can easily forge new description phrases terms that suit the character of their Sub-organizational Unit best. On the other side, the solution does not loose effectiveness in situations where a direct match is not present in the repository.

5.4.2 Result caching

Query caching in Information retrieval has a very important role as a feature that is boosting effectiveness of any IR system [Garrod et al., 2008,Ozcan et al., 2008,Martin et al., 2010].

The proposed solution also includes mechanisms that allow for results of query caching. This is mainly dictated by the need of further efficiency gains in terms of execution time. The novelty of the caching mechanisms proposed here is based by the retention of cache data through out the systems lifetime. It can be achieved due to relatively small amount of data concerning Web services in contrast with documents indexed by common purpose Web search engines.

The essence of this mechanism lays in the fact that every description of Web service operation is recorded by a solution implementing the model and the terms used in the phrases are easily traceable to previously issued queries.

Therefore, once a query was issued and a set of operations was retrieved it should be constantly updated so that subsequent queries refer to the result set bypassing the initial mechanism resolving the query and issuing necessary actions at specific subsystems.

In order to prevent memory exhaustion a set of supporting mechanisms is to be supported. Such auxiliary mechanism allow for coordination caching mechanism with

frequency of a particular query. When some previously defined threshold is met, a query result set is ripe for caching. The threshold is to be understand as a frequency over some period. Depending on the size of the repository and organization needs it may be expressed in days or in minutes.

The introduced modification of the caching mechanisms satisfies not only the effectiveness aspect of desired by users solution, but also a cost and scalability aspects. Being sure that the data in cache always cover the complete matching set of Web service operations, one does not risk a lapse originating in the fact that some important operation was omitted.

The scalability aspect is reinforced by decreasing the size of the search space by introducing a mapping between often issued queries and their constantly updated result sets.

What is more, there is no penalty from using cache as there is no possibility of having it outdated thanks to previously presented mechanism.

5.5 Summary

The utmost care was applied to designing the model in such a manner that there is a shift of focus from technical details of Web services to their functionality as might be perceived by regular business users and application developers.

As was remarked, this is of paramount importance as a service repository is crucial element of any SOA oriented organisation. It allows for searching of desired functionality independently of exact terms used in technical Web service operation description and additional functionality.

The introduced functional description structure as a triple of functional phrases completed by a vector of trait tuples conveying Non-functional properties is a realisation of this postulate. As detailed, it also improves the execution in terms of time in comparison with fully-fledged ontology frameworks as no operations incurring exponential time are necessary.

Functional triples and trait tuples are also an implementation of user orientation of the solution. As mentioned many a time in this work, there is no absolute way to present an all encompassing ontology that satisfies all the needs. There are initiatives that introduce controlled vocabulary that serves as an organising topology where simple relations of subsumption are the only available ones (as in [Muddamalle, 1998]). More bold attempts try to implement more advanced structures, yet a successful deployment is scarce due to the efforts and complexity of a task.

Instead, the solution encourages a coverage of local domain of any given organization substructure. This makes members of this Sub-organisational Unit comfortable with the terms gathered with their knowledge, that concern their daily tasks. Mapping across SU is facilitated by Local Context Anchoring which is not a definite solution but does not leave a user without assistance as long as possible by expanding all possible vectors of relation between unknown terms and terms used in Web service operations' descriptions.

Part III

Verification of the proposed model

Chapter 6

Evaluation of the proposed model in the electronic economy settings

This chapter focuses on the evaluation of the previously introduced artifacts. The Key Requirement Aspects defined in section 2.6 induce that one has to thoroughly focus on the assessment of compliance of the obtained artifacts to one or many requirements it was designed to satisfy.

The assessment was carried out either as an actual data-driven experiment where a test bed was prepared with addition of test data. Where a synthetic experiment was not feasible, an experiments with users was prepared in order to gather a feedback on the artifact's effectiveness.

Having obtained a number of research artifacts, one is bound to verify their applicability in terms of dissertation's thesis. This is a direct application of guideline 3 originating in [Hevner et al., 2004]. The dissertation's thesis is a complex and compound statement that ensures every of the design evaluation methods from the following list is applied:

- observational,
- analytical,
- experimental,
- testing,
- descriptive.

One has to note that, application of the above-listed methods requires a rigourous approach to the research as given in [Lakatos, 1978].

6.1 Validation setup

6.1.1 The invited professionals

In order to validate whether the proposed model caters for the requirements define in Section 2.6 available pool of professionals willing to participate in experiments had to be assessed and classified.

This classification and assessment were necessary so that any observations from the experiments could be matched with the assumptions as to the objectives of various user groups and their skills. The following part of this subsection describes background of assembled group. What is more, these professionals were the ones that enabled to refine the initial set of requirements into recalled Key Requirement Aspects.

The invited professionals represented a variety of organizations, some of them thanks to the nature of their line of work could share insight of more than one organisation. Data gathered in this panel is relevant to period between 2006 and 2011. The questionnaire, which is covered below, aims at refinement of initial assignment of the professionals to their groups. In addition it introduces a variety of data that is used to verify whether the scope and features of the model presented in this dissertation along with its prototype instantiation are well received by the potential users. Among the questions asked, there were none concerning specific types of solutions available in any organisation, therefore no trade secrets were infringed.

This was a panel research where the population was purposefully sampled from organisations operating on territory of Poland. The home organisations of the surveyed professionals were in majority large ones (over 1000 employees). Over a half of the home organisations operated on other than Polish territory.

The professionals answered questions based on their professional experience and expertise gathered in organisations from the following sectors:

- Logistics,
- Wholesale of food,
- Meat processing,
- Automotive,
- Oil processing,
- Banking,
- Insurance,
- Public services,

- Electronic Payments,
- Breweries,
- Consulting,
- Higher education,
- Defence industry.

The questionnaire consisted of the following questions:

1. Are you familiar with an idea of Web services and Service Oriented Architecture?
2. Are you familiar with one of methods of service implementation such as: Web services, REST services, AJAX based content requests, xml-rpc? If there are other methods please elaborate.
3. Are you familiar with Semantic Web services implemented in technologies such as OWL-S, SAWSDL, WSMO, SWRL, WSDL-S or one of the subtypes or derivations?
4. Are you familiar with a concept of an ontology as a knowledge representation structure?
5. Are you familiar with a concept of controlled vocabulary as a knowledge representation structure?
6. Are you familiar with a concept of taxonomy/hierarchy as a knowledge representation structure?
7. Are you familiar with a concept of content tagging as a knowledge representation structure?
8. Are you familiar with a concept of controlled vocabulary as a knowledge representation structure?
9. Is any of the above inquired knowledge representation structures used in your organisation? If the answer is affirmative please share which one? If its negative please share if there are any other methods available in your organisation as knowledge representation structure?
10. When in need of finding out whether a certain functionality is offered by your organisation, is it possible to use some previously used knowledge representation structure?

6.1.2 The qualitative results analysis

Due to the relatively small group of respondents and impossibility to produce statistically relevant results, the questionnaire was treated as an axis during the interviews

Table 6.1: Summary of questionnaire results

Job title	Business	Web services	SOA	Semantics	K. Cat.	Search
Consultant	Oil processing	P	P	T	P	S,D
Manager	Food wholesale	N	N	N	T	P
Administrator	Logistics	T	T	T	P	P, D
Programmer	Higher education	P	T	T	P	S,D
Administrator	Higher education	T	T	N	P	S,D
Consultant	Breweries	P	T	T	P	S,D
Consultant	Breweries	P	T	N	P	S,D,P
Project Manager	Insurance	N	T	N	P	D,P
Consultant	Telecom.	N	T	N	P	P
Manager	Banking	T	T	N	T	P
Programmer	Defence	P	T	N	P	S,D
Analyst	Defence	T	T	N	P	D, P
Manager	Defence	T	T	N	P	D, P
Consultant	Automotive	P	T	T	T	P, D
Programmer	Wholesale	P	P	N	P	S,D
Manager	Consulting	T	T	N	P	P
Analyst	Banking	N	N	N	P	P
Consultant	Meat processing	P	T	T	P	S,D
Analyst	Electronic payments	P	P	P	P	S,D,P
Researcher	Higher education	P	P	P	P	S,D,P

Results given in this table present overall level of familiarity with various technologies and concepts. Due to low number of respondents results are not statistically significant. Upon request, a clarification of question with additional examples was provided. Results were aggregated according to main interest areas. Values possible: P - practical application, T - theoretic knowledge, N - not familiar with. Options for last column: S - search engine, D - documentation, P - query to peers. One or more options possible.

with the participants. The results were aggregated in table 6.1 so that the key interest points for this work could be reviewed.

Web services

The majority of the interviewees was aware of a Web service technology, both thanks to the education and actual usage observed in one or many organisations they were working for. Those that could not connect idea of a Web service with its most popular implementation realised in a WSDL document were business users dealing with project management, process planning or daily usage of a set of systems as an analyst.

Service Oriented Architecture

Once more, technically inclined interviewees were aware of the Service Oriented Architecture paradigm. In addition there was strong correlation of the type of an organisation and its activity on global scale and the presence of the SOA implementation elements such as widely available services and publicly available repositories listing them along with some kind of description.

Semantic background of the interviewees

The depth of knowledge of semantic description technologies such as OWL-S or WSMO was dependent mainly on two factors. The first one was received education, the interviewees which graduated the specialities oriented on Computer Science, Information Technology or Computer related engineering studies during the period of the last 7 to 9 years were aware of the existence of such a trend as a Semantic Web, yet only those graduating within last 5 years were familiar with one of the inquired technologies.

The second factor was experience gathered in pursued occupation. There was a strong correlation with developer skills or consulting expertise (especially in domain of interface preparation). What is interesting, none of the interviewees recalled actual deployment of a system that used a semantic based technology (using OWL-S, WSMO, SAWSDL, SWRL or WSDL-S) in production environment outside the research performed by the academia.

Categorisation of resources and means of purpose statement

In most of the organisations that were covered by the interviewees there were some means of repositories and directories listing available IT resources. Majority of implementations was based on the idea of the Intranet or Corporate portal [Winklbauer and Seidenberg, 2001]. Interviewees of developer oriented background added a number of source control systems [Rochkind, 1975] and knowledge bases [Amsler, 1984] along with documentation.

Both IT and business oriented users were not capable of naming solutions in their organisations that would allow for capturing a purpose statement. The most frequent techniques they reported to use was search engines (on internal and external data) and peer inquiries. As recounted by few participants, the documentation was usually the last resort.

6.1.3 Overall assessment of the situation in terms of desired features

From the carried out research among the various professionals it is clear that the most sophisticated technologies aimed at better overall management of Web service assets are poorly recognised. The exception are those professionals that are either technologically inclined thanks to their carrier choices or have some CS or IT oriented curriculum.

What is more, majority of the organizations where the invited professionals worked, doesn't have a complete solutions supporting the SOA paradigm. It is unclear whether the reason for lack of thereof is a cost of implementation, satisfaction with currently used solutions or lack of knowledge on the possibility of change and the benefits of SOA as given in [Papazoglou and Heuvel, 2007].

Above all, nearly all of the queried professionals said that there are some measures to find necessary functionality in their organisation. However, it was emphasized that the most commonly used tools are search solutions that follow the standard set of IR capabilities. As additional methods asking of peers was given, and reviewing documentation.

Business users emphasized that search solutions that are too cumbersome to operate are promptly discarded due to internal choice of effort versus result and other means are preferred to answer the most important at the time questions.

Professionals expressed that many a time there is some kind of documentation intended to broadcast messages to future developers or to self. It can be even structured

by so called code of conduct [Bia and Kalika, 2007]. Therefore, solutions that try to leverage already established methods were highly regarded thanks to increased cohesion of daily used toolset. Solutions that offered a lot in exchange for new model of thinking and altered usage scenario were regarded as not appealing to the questioned professionals.

As an explanation, many professionals gave a distaste for cluttering their already crowded set of systems and applications used to achieve their business objectives.

6.2 Coverage on key solution aspect

The presented model and the mechanisms enabling it address all of the enlisted aspects. There is no direct mapping on mechanisms and aspects as some of the mechanisms allow for addressing more than one aspect. It was generally perceived, that the most important change in a Web service description was introduction of functionality description that is easy to grasp for an untrained user. The discussion will be organized by reviewing impact of the crucial of the model's capabilities and mechanisms implementing one of the above enlisted aspects.

The new Web service description model influences, to some degree, all of the aspects. Nevertheless, the obvious primarily targeted aspect is the purpose statement. Thanks to introduction of the new description model the rest of the aspects is positively influenced as well. The model improves scalability a great deal as it is possible to match a millions of Web service operation descriptions in less than one second and is capable to sustain this level of robustness for 10 times more by caching of results and parallel execution.

The cost of the whole description and retrieval is greatly diminished thanks to the simplicity of description, its distribution among Sub-organisational Units and no learning curve when querying the resources. The effectiveness is, in general, higher than in standard IR based solutions. This is of course, a result of clear definition of functionality in the new model that is similar to the semantic based solutions.

The inclusion of the NFP vector naturally enhances scope of the model by linking individual Web service operation descriptions with metrics important for business users.

The Local Context Anchoring, is the key mechanism that boosts the overall effectiveness of the model in situations where a user would be left without any support. It surveys a number of possible paths that may lead to mapping of user supplied terms on those available in Web service operation descriptions and its aggregates. This is a

very valuable feature in opinion of all the surveyed professionals. Its value, in interviewees opinions, stems from the obvious increase of usability and general friendliness of the system and increase of user comfort realized by further flattening of learning curve. This is viable even in situations where some evidently misplaced Web service operations are included in results.

Mechanisms responsible for Local Controlled Vocabulary are specially designed to cut the overall cost of the process. As discussed earlier, they provide a set of tools that automate the process of term selection when a batch of relevant documents is submitted. Thus, the mechanism presented creates a beachhead for further actions administered by a user. Interviewed professionals emphasize that starting point provided in an automated manner is a great advantage over situation when user has to develop everything on his own.

It is crucial to repeat, that this mode of work is not supported by any semantic based solution as a key feature, and might be seen only as a part of deployment methodology. The obvious improvement is incorporation of such a strategy as a core feature.

6.3 Comparison of the presented mechanisms with the alternatives

Taking only the raw numbers presented in section 3.6 without further discussion cannot be justified as the tasks that were prepared demanded higher level of processing capabilities than those necessary in the proposed model operating on the extended phrase triple.

Nevertheless, from the business user perspective, the size of the ontology and the number entities annotated with it can be perceived as not satisfactory (to the point where special measures are devised to assess the quality of an ontology [Yao et al., 2011]).

Even taking into account the complicated nature of the reasoning, waiting time for a number of queries is unacceptable under the paradigm of the robust resource retrieval (the longest accepted time limit for information retrieval is set at 4 seconds [Tomasic and Garcia-Molina, 1993], while at a time of writing, a reasonable one, it was further narrowed down to a range of 1 to 2 seconds for simple tasks [Hoxmeier and DiCesare, 2000]).

What is more, great variance of time needed to retrieve results is undesirable. It does not build a consistent user experience what can repulse potential users for this type of solution. Techniques used by semantic enabled solutions (computing of all possible statements in advance) directly affect the initial load time of any of the solutions. As remarked, when an error occurs, reboot of any service introduces a considerable disturbance for all of the users. It is another unacceptable characteristic of a robust solution.

6.4 Experiments with the semi-automated shortlist building

To demonstrate the idea of semi-automated shortlist building please consider the following example. Its main aim is to present how the automation process can output relevant terms based on some initial setup. The measure of length was chosen arbitrarily to decrease the influence of common verbs and nouns that were not qualified as stop words.

A set of 57 documents was taken into consideration. All the documents were submitted to standard Information Retrieval procedures to remove unnecessary terms[1].

The 57 documents were tagged as those regarding the bioinformatics[2]. The total number of terms was 20401. Table 6.2 demonstrates relation between a length of terms eligible for shortlist and their frequency in documents. The best effects in terms of manageability of resulting shortlist when the length of term exceeds the average length of English word and frequency the term is 6 or higher.

Exemplary data is given in table 6.3. As one can see, a selection of terms is quite broad, yet with the highest values for frequency and length, a number of terms was lost so that a nature of shortlist cannot be easily guessed without prior knowledge of its source. Therefore a safe choice for automated procedure would be to compute a list of terms resulting from parametres close to the efficiency level yielding a shortlist of circa 100 terms. The experiment was performed with documents originating from other domains such as Agriculture, Biochemistry, Bioengineering, Cryobiology, Health and Medicine, Psychobiology, Botany and Genetics.

[1] As unnecessary one takes terms being in the available lists of stop words [Fox, 1989]. Additional actions such as term case homogenization or removal of stray characters occur as a part of this.

[2] Obtained from the `http://www.biology-online.org`

Table 6.2: Shortlist's length in terms of frequency of a term and its length.

	1	2	3	4	5	6	7	8
1	1479	724	429	277	194	136	101	72
2	1452	713	422	273	191	134	100	71
3	1373	676	396	255	179	125	92	66
4	1198	586	346	222	155	106	79	57
5	1018	489	285	179	121	80	65	49
6	814	384	223	141	99	64	51	40
7	621	281	162	103	74	46	37	30
8	423	190	107	67	45	30	25	20

Frequency increases with every row, length of a term increases with every column. Used range for both parameters is 1 to 8.

Table 6.3: Exemplary data from an experiment on automated shortlisting

Term length, term frequency	Example shortlist
5 , 6	africans, albicans, arabidopsis, baumannii, botulinum, bushmen, cancer, cancers, candida, change, changes, chilean, climate, clostridia, compounds, comprehensive, computer, course, database, design, development, diabetes, disease, duplications, europe, expression, fluorescens, fumigatus, genetic, genome, genomes, genomic, genotypes, giardia, graham, haplotypes, immune, important, individuals, information, institute, malignant, melanoma, method, microbial, micrornas, murphy, mutation, mutations, nervous, neuropeptides, nonlinear, organism, patient, patterns, peptides, perkinelmer, potato, potatoes, primers, probes, project, proteins, reference, region, research, rosetta, sanger, science, sequences, southern, species, squirrel, system, tetrahymena, variation, wellcome, werren, wolbachia

6 , 7	baumannii, botulinum, cancers, candida, chilean, climate, compounds, computer, database, development, disease, duplications, expression, fluorescens, fumigatus, genetic, genomes, genomic, genotypes, giardia, haplotypes, individuals, information, institute, malignant, melanoma, microbial, micrornas, mutation, mutations, neuropeptides, nonlinear, patterns, peptides, perkinelmer, potatoes, primers, project, proteins, reference, research, rosetta, science, sequences, southern, species, squirrel, tetrahymena, wellcome, wolbachia
7 , 8	baumannii, botulinum, compounds, database, development, duplications, expression, fumigatus, haplotypes, information, institute, malignant, melanoma, microbial, micrornas, mutation, mutations, nonlinear, patterns, perkinelmer, potatoes, proteins, reference, research, sequences, southern, squirrel, tetrahymena, wellcome, wolbachia

Equipped with a shortlist provided by a computer routine, it must be enriched by terms missing or changed by the IR techniques. The changes performed were sure to flatten terms such microRNAs to its lowercase form and remove punctuation marks in order to more easily process input text. This simply could be unacceptable for specialists from some domain, as all those filtered peculiarities are meaningful in their specific context. Therefore, the provided list shall be examined for such changes and remedied by SE.

What is more, without additional investigation of the outputted list a number of common terms can be included such as term variations and terms that are over-represented in used corpus. The given length of 100 terms is used as a artificial boundary. A better indicator can be a compound boundary could be a composite one that would steer a routine to output at most of 100 terms from the 10% of the top ranked in the document corpus where some additional corpus could be used as an enhanced list of stop words.

As mentioned, missing terms should be introduced with special focus on terms that are to act as verbs (α terms). This is important action as verbs are usually short terms and many from the most often used auxiliary verbs are treated as stop words therefore removed in the preparatory phase. Without terms expressing some action

Table 6.4: Common action terms in WSDL documents

add, change, config, contact, delete, display, execute, export, fetch, generate, get, insert, list, load, log, report, report, save, search, send, set, update, validate, view

proposed description should amount to nothing more than a list of tags (techniques such as Semantic Compression can be also used [Ceglarek et al., 2010]).

Some of the most common action terms found in Web service operations' names (obtained during experiments on available Web services in the open Internet) are given in Table 6.4:

By no means this is a complete list of eligible terms, yet it allows for forming an opinion on the nature of the action terms used. It may be necessary to extend it by a number of specialised action terms in order to fit the domain of SU. As mentioned, this is a task of SE.

In essence, this phase is not very different from any another approach available either in the ontology building or designing the relational database structure as it is an activity focused on enumerating a shortlist of the most important entities in the modelled world.

6.5 Experimental verification and improvement of the match algorithm

The experiment was prepared so that two of the Key Requirement Aspects - the effectiveness and the scalability - could be validated. The effectiveness is verified by the capability of the prepared algorithm of fetching the results complying with user queries build with the elements described in the Web service description model. Scalability is measured by submitting an increasing number of Web service descriptions to the proposed algorithm and measuring whether consequent batches comply with the imposed criteria of time efficiency.

Due to considerable number of Web service operation descriptions needed to verify the above key requirement aspects, a special test bed was prepared. As the proposed algorithm was built in phases this process is reviewed beginning with the initial implementation execution results and experiment setup and finishing with the final one that includes various observations gathered in the course of the research activities.

The initial version of the experiment proved that the whole proposed idea is a feasible one and that it can be applied to a substantial repository of descriptions. First of all, experiment's results are given along with presentation of experiment's setup. Following that a discussion on the initial version of algorithm is given along with important observations that led to further improvements.

The initial setup follows the general procedure given in section 5.2.1. Web service descriptions are filtered first by values of the NFP vector present in user's query, then phrase elements are matched with phrase caches in order to narrow down the search space. The actually matching Web service operation descriptions are produced by intersecting the sets containing the results from the query-phrase to phrase-caches match up.

The initial experiment was prepared with the below given constraints. The number of Web service descriptions in the initial experiment was increased with every iteration from the initial of 100 descriptions to a final value of 1 million descriptions. Every Web service description was prepared according to a previously discussed model. There was a constraint on α and β phrases to use a single term. The γ phrase could be built from 1 to 8 terms. The nfp vector contained a variable number of elements (also from 1 to 8) that could be different for every description. Generated queries were short, in order to reproduce the nature of Web queries (please refer to [Belkin et al., 2003] and [Arampatzis and Kamps, 2008]). There were mandatory 2 elements, one for α and β phrase each, and up to 3 γ phrase elements. The generated query nfp vector could contain from 1 to 8 elements with values within lower and upper bounds of the NFP elements observed in the nfp vector generated for Web service operation descriptions.

One can easily see, that the obtained results deliver a good level of effectiveness. Yet when one is to consider the mounting time efficiency between subsequent operations, he is ready to observe that it is not as good as one could imagine.

As the matching process is driven by the intersection of sets, one can discover with an auxiliary experiment the baseline efficiency of theoretically similar operation. To begin with, one has to be aware of the complexity of the set intersection operation. As can be verified in both [Baeza-Yates, 2004] and [Dachman-Soled et al., 2009] the set intersection boundaries in terms of computational complexity can be defined as follows:

- the worst case – $O(n * m)$,
- the best case – $O(n * logm)$,

where m and n are the sizes of sets to be intersected.

Apart from inefficiencies in the initial implementation, one could blame the overall unrealistic nature of the initial set of data that is not correlated with the actual usage patterns that can be observed in WSDL documents observed in the open Internet. Other important observation is that when intersecting more than two sets one has to be aware of the fact that complexity is further increases by a factor of $O(n_1 * m_1)$ in the worst case, where n_1 is the number of elements in the product of the intersection of the two first sets and m_1 is the number of elements of the consequent set. If one is to follow the procedure described in section 6.5, there are at least three sets. One for each of the phrases.

Therefore a baseline had to be computed in order to verify the robustness of the initial version. It was accomplished as a result of an auxiliary experiment that provided data on time efficiency of set intersections with incrementally increasing number of elements. In order to have the worst possible scenario, all three sets were equinumerous. The setup was similar to the initial experiment and consisted of six repeated runs for the mentioned three sets, starting with 3 sets of 100 elements each, finishing at 3 sets of 10 million elements each. Results are summarised in Table 6.5. If one is to plot the results the obtained chart demonstrates similar shape, yet the initial experiment's results proved to lack greatly in comparison with the baseline obtained from the auxiliary experiment.

Table 6.5: Results from the auxiliary experiment on set intersection time overhead

Number of elements	100	1000	10000	100000	1000000	10000000
Time in milliseconds	<1	1	7	29	552	7638

As the obvious inefficiencies were removed in the subsequent implementation, several enhancements were introduced along with new constraints as to the Web service description structure used in the experiments. It resulted in efficiency gains that surpassed author's expectations. Within a fraction of a time needed to filter 1 million of Web service descriptions in the initial experiment, 10 millions could be filtered.

The above results are presented in 6.6. As mentioned efficiency gains were obtained by re-engineering of the code and different strategies as to the structure of a Web service operation description. The structure of descriptions and queries was set as follows:

- One and only one element for an α phrase from the corpus of 50 elements (in initial version 90 possible terms),

Table 6.6: Results of the improved matching algorithm

Number of descriptions	Average time for algorithm	Nominal set intersection time
100	0.000038175809	0.0000259876257
1000	0.000092013316	0.0001471042636
10000	0.000257968902	0.0016930103302
100000	0.001834869384	0.0287311077118
1000000	0.021018028259	0.5525181293493
10000000	0.441106071472	7.6216189861323

Results from effectiveness experiment measuring execution time of improved query matching algorithm against test data reflecting the new structure of Web service operation description

- One and only one element for a β phrase from the corpus of 90 elements (down from 90),
- One or two γ phrase elements from the corpus of 110 (down from 190),
- Five NFP elements the same for every description varying in actual value.

The above structure is derived from the actual WSDL documents gathered in the course of the research activities. It is obvious that none of them adhered to the proposed model of Web service description, yet many of the available Web service operations had a structure resembling one proposed in the model. The exemplary data is provided in table 5.1. The closer examination carried out on the available corpus that using more than 2 gamma phrases would be groundless in majority of situations. What is more, γ phrases with more elements incur additional cost at the description stage.

The lower number of γ phrase elements impacts the overall performance, yet it is not the only source of robustness. A number of test was run to measure the impact of lower number of elements. The results showed that up to a maximum of 12 γ phrase elements the performance is still better than the baseline. Results are given in table 6.7.

In addition to the smaller number of γ phrase elements used in the tests, a number of extensions was introduced in comparison with the original algorithm as given in 1. The most important one is implementation of full match and partial match features. Full match describes a situation when every term from every phrase along with proper

Table 6.7: Impact of number of γ phrase elements on the execution time

Number of elements:	4	6	8	10	12
Time efficiency in seconds:	0.98	1.37	1.73	1.905	2.75

Results from additional tests measuring impact of the number of γ phrases on overall performance. All tests values are gathered for matching 10 million Web service operation descriptions against a single query.

namespace is found when tested against user given query. This is obviously the must have scenario.

In addition, a partial match was implemented to avoid situations where there are no results due to the a mismatch in namespaces. A rationale for this action is the fact that a user might not be interested in namespaces at all. He might be willing to review all the Web service operations that were described by the terms he provided, and later on decide which he prefers based on the origin of Web service operation given by a namespace and additional data linked to it.

To implement that mechanism, an additional cache was implemented which stores namespace-phrase pairs. Due to a great many combinations of these two elements the size of the cache is considerable, yet it is beneficial factor, as thanks to the cache characteristics it easily allows for filtering of a great number of Web service operations.

Both matches, the full and the partial one were decided to be implemented side by side so that the success of a query match is made more probable than in the initial implementation tested in the initial test. It is important especially in environments that have a lot of independent divisions with their own namespaces. This is supporting the model's premise of handling independent Sub-organizational Units.

Final improvement to the implementation of the mechanism was the support for concurrency. Thanks to the plain structure of Web service operation repository and easy partition of its content distribution of its parts across several subprocesses is readily available. Thanks to the above, the execution time is diminished nearly (allowing for synchronisation overhead) proportional to the number of available processes running on separate cores.

The data gathered and presented were obtained by running tests on moderately new workstation equipped with 3 GB of RAM and running Pentium Core 2 Duo processor 2.4 GHz. All the mechanisms were implemented in Python and run with the PyPy 1.7 implementation.

6.6 Local Context Anchoring evaluation

The Local Context Anchoring is important for the whole model due to the fact that it decreases the overall cost of searches as it bridges unknown terms with those indexed as parts of the available Web service descriptions in organization's repository. Apart from the cost effectiveness it boosts the effectiveness of retrieval as it does not leave model instantiation users without possibly relevant Web service operation descriptions.

To evaluate robustness of the LCA a set of tasks was prepared. The tasks were designed to capture user's satisfaction level with the proposed Web services operations descriptions delivered by the algorithms implementing the mechanism.

In order to manage the whole scenario, a test repository was crafted based on available data gathered in the course of research. Due to the fact that Web service operation names are very diversified, only those that adhered to the structure proposed in the model were used. In order to prepare the experiment, a number of transformations had to be applied to the available corpus of Web service operations' names.

The use of already available Web service operation names was dictated by the fact that it is very difficult to prepare sufficiently smart algorithm generating Web service operations names in such a manner that should allow users to relate with them in terms of functionality description. Even when, the naming structure of Web services operations is far from the one postulated by the model introduced in this work, it was found that the non random experiments are far more plausible due to the possibility of the presentation of the source WSDL documents.

In order to provide meaningful evaluation, that gives some insight into the reception of the LCA functioning, users were given a number of queries and results proposed by the mechanisms. Those were evaluated by them in terms of similarity between perceived intent of a query and the presented results.

It has to be noted that, no additional information was made available. An experiment was conducted under principle of reliance of the Web service operation description. Every batch of queries and suggested results was processed by users with business background and those specialising in Information Technologies.

This is important, as one could easily observe whether the given datasets are perceived as helpful by different groups of interested users.

Important aspect of the experiment was the evaluation of the capability of the algorithms to bridge the gap between user's perception of business environment and the actual terms used in Web service operation description.

6.6.1 Experiment organisation

The test corpus consisted of three separate domains pertaining to individual namespaces (characteristic is given in Table 6.8:

- communication subcorpus - **Cc**,
- operations management subcorpus - **Oc**,
- financial subcorpus - **Fc**.

Table 6.8: Description of subcorpora used in experiment

Subcorpus	Number of operations	Eligible operations	Diversity	Expertise
Cc	933	850	Low	High
Oc	2428	2408	High	Moderate
Fc	2297	2267	Moderate	Moderate

Each of sub-corpora was built on data available from the previous research activities of the author. From this corpus, over 20000 are Web services deployed in ASP.NET technology characteristic due to asmx file extension. These are well suited for testing as they come with some description more often than pure WSDL documents. What is more, a large number of the retrieved WSDL documents from the ASP.NET Web services [3] were functional at the time of experiments. **Eligible** operations informs on the Web service operations that were at least two words long. All Web service operations were chosen to be in English. **Diversity** column implies how many related subdomains were covered by any of the subcorpora. Level of **expertise** informs on the level of specialistic knowledge needed to understand the purpose of Web service operations.

The three sub-corpora characterised in table 6.8 are amassed from the actual Web services operations obtained throughout the research activities. It was decided to use the ready available Web services due to two the fact that random building of Web service operations adhering to the postulated model would produce a lot of meaningless results. What is more, it is difficult to reasonably manage the domains to produce Web service operations. Using ready Web services allowed for preliminary human driven classification and tests on real world originating data.

The resources used for the term resolution were:

- domain publications - available articles, books and manuals were applicable and available,

- Wikipedia [4],
- Freebase [5],
- Wordnet [6],
- The free dictionary - dictionary and thesaurus [7],
- Wiktionary [8],
- Roget's II: The New Thesaurus [9],
- Merrriam-Webster dictionary and thesaurus [10]
- results from the Web search[11].

The running time of the LCA depends on the caching mechanism. If the query terms are present in the caching mechanism, the average time of response is below 100 milliseconds. Whether, a term or a number of terms is not present in the cache, the response time is much longer. The average for two unknown terms is 35 seconds. This is due to the fact that implementation of the LCA must contact a number of Web resources that differ in response time and level of complexity. Some resources fork into a number of other resources that have to be processed.

The used implementation was prepared in such a manner that 500 terms present in Web service operation names were cached before the experiment so that probability of extensive wait for users was low.

In order to make the whole process manageable for the participating users, a number of tasks was limited to a total of 6. All 6 were questions concerning the level of match between a given query and its results and the rest was devoted to description of the objectives, formulation of queries and evaluation of results. Each set was processed by a total of 8 users. Four with business background and four technology oriented. Tasks concerned all of the listed earlier domains. Table 6.9 provides a content of a form used to gather user responses for tasks 1 to 6.

As the LCA is not a traditional mechanism for retrieval of information, a classic approach of query evaluation cannot be applied [Vaughan, 2004]. There is no means to

[4]English version of the Wikipedia - The free Encyclopedia `http://en.wikipedia.org/`

[5] Community driven collaborative knowledge base - `http://www.freebase.com/`

[6] Implementation of [Miller and Fellbaum, 2007] delivered by the Natural Language Toolkit `http://nltk.org/`

[7] Multilingual dictionary accompanied with various thesauri `http://www.thefreedictionary.com/`

[8]An open, collaboratively built multilingual dictionary `http://en.wiktionary.org/wiki/Wiktionary:Main_Page`

[9]Hosted at `http://education.yahoo.com/reference/thesaurus/`

[10]Hosted at `http://www.merriam-webster.com/`

[11]The DuckDuckGo search engine was used as it does not demote queries issued from command line along with high quality of search results `http://duckduckgo.com/`

definitely state whether what is the level of precision and recall [Davis and Goadrich, 2006] without producing a sufficiently controllable group of Web service operations names and deciding that they should be present in results concerning a number of queries.

Without examination of a Web service operation capabilities one has to evaluate user preferences as to the results of query. To exemplify, results of the same query can be valued differently by users with different backgrounds. Their valuation is affected with their education and experience. Thus relevance is very subjective matter.

In addition, LCA is a supplementary mechanism which produces results later processed by ranking procedures to include popularity of selected Web service operations.

To summarise, every task required a valuation on the users side. The valuation given by a user considered:

- a level of overall satisfaction with the proposed results - scale 1 to 10, where 1 is recommendation are extremely unhelpful and 10 is recommendations are extremely helpful - this can be perceived as traditional measure of precision, yet it is not,
- a level of noise - scale 1 to 10, where 1 is definitely too many irrelevant results, 10 is only relevant results - this can be perceived as traditional measure of recall, yet it is not.

The satisfaction and the level of noise cannot be precision and recall due to the previously remarked fact that one cannot state the total number of relevant Web service operation names due to its subjectivity.

Query number 3 is a special test query that provides a sanity check as to the values of both measured elements. The suggestions are scare due to the limited number of Web service operations, and the they are only a slight extension of the terms available in query itself. The values provided by the participants are expected to be close to 10.

Rest of the tested queries addresses a combination of functionalities important both to IT centric personnel and business users.

Table 6.9: Evaluation of LCA

Task	Query	Results	Subcorpus
1.	project schedule	GetHistoricalTimeSeries, ExportWorkOrdersByActionSystemTimeWindowPortfolio, GetHistoricalTimeSeriesWithStartDate, ProjectGetTimeEntriesByDate, ProjectGroupGetProjects, ProjectGroupFindByNumberList, ProjectFindByNumberList, ExportWorkOrdersByActionSystemTimeWindow, ExportWorkOrderActionsByTimeWindowPriority, ProjectGetTimeEntries, TimeEntryGetProject, ExportWorkOrderActionsBySystemTimeWindow, StochasticProjection, SendVertWorksheet, ExportWorkOrdersByActionTimeWindow, ExportWorkOrderActionsByTimeWindow, ProjectGetOpenSubProjects, SendWorksheet	Fc
2.	user contact	ContactsAffiliateFindNotAssociatedUsers, ContactsManufacturerFindNotAssociatedUsers, ContactsVendorFindNotAssociatedUsers, ContactsManufacturerFindAssociatedUsers, ContactsAffiliateFindAssociatedUsers, ContactsVendorFindAssociatedUsers	Oc
3.	user list	ListUsers, ListAdminUsers	Cc
4.	banking offer	ContentModuleControllerFindOffers, BusinessRulesAvailableProcessorsGetOfferProcessorPriority, PaymentBankDeposit, MarketingOfferGetAllOffers, GetQuickBooksSalesJournalEntriesByCompanyIDAndOrderSource, GetSalesOrderByAccountCode, BusinessRulesAvailableProcessorsUpdateOfferPriorities, GetSalesOrderSummaryByAccountCode	Fc

5.	budget information	BudgetFigureSetAmountDefaultCurrency, BudgetFigureGetData, AccountingPeriodGetData, TimeEntryGetData, AccountingYearCreateFromData, CostTypeGetDataArray, BudgetFigureCreateFromDataArray, CompanyGetData, CostTypeGroupGetDataArray, AccountingYearCreateFromDataArray, CompanyUpdateFromDataArray, BudgetFigureGetAmountDefaultCurrency, TemplateCollectionGetData, CostTypeGetData, TimeEntryUpdateFromDataArray, AccountingPeriodGetAccountingYear, TimeEntryGetDataArray, CompanyGetDataArray, BudgetFigureCreateFromData, AccountingYearGetPeriods, GetSearchMetadata, TimeEntryCreateFromData, CompanyUpdateFromData, CostTypeGroupGetData, AccountingPeriodGetDataArray, TemplateCollectionGetDataArray, BudgetFigureGetDataArray, AccountingYearGetDataArray, AccountingYearGetData, TimeEntryCreateFromDataArray, BudgetFigureUpdateFromData, TimeEntryUpdateFromData, BudgetFigureUpdateFromDataArray	Fc
6.	change filter	RemoveIntegratedFilter, ListSocketFilters, ClearFilters, ConfigFilters, RemoveActiveFilter, DeleteFilteredEmails, ListIntegratedFilters, ConfigAccountDefaultFilters, DeliverFilteredEmails, ListScriptFilters, SendNdrFilteredEmails, RetryFilteredEmails, ListFilteredEmails, RemoveSocketFilter, RemoveScriptFilter, ListActiveFilters	Cc

6.6.2 Experiment results

The results of the validating experiment are available in Table 6.10. As one can observe, the results of evaluation are especially good when one is to consider the satisfaction of the experiment participants with the provided suggestions.

The average satisfaction level is **above 7**. This proves that the implemented LCA is capable of presenting reasonable suggestions based on a query across a body of descriptions not matched directly by the query terms.

The nature of the experiment sub-corpora was explained in the previous section. One has to add, that there was no guarantee that there are Web service operations that can provide specific functionality.

It is especially visible in satisfaction level results of the forth query and its suggestions. As the level of satisfaction is rather low, one had to examine the corpus more closely. If as majority of the experiment participants, one would expect a Web service operations delivering a consolidated offer for bank services, there is no Web service operation satisfying this need. Nevertheless, the LCA is built to suggest some Web service operations that could fit any possible query sense. As a result, Web services containing references to accounts and the act of finding are present.

The average noise level is **above 6**. It is worse than the satisfaction level, yet one has to remember that the noise should be further decreased by the subsequent mechanisms.

The most important one is the ranking functionality that is bound to remove all the unpopular and of low usage Web service operations from the list presented to the user. What is more, division of the results according to the position of query issuing user will further allow for levelling up the overall satisfaction with the level of noise.

In general, participants were impressed with the presented queries and resulting suggestions. They especially liked situation when suggestions reached beyond traditional query expansion an presented results that used not only synonyms and related terms but also those are somehow relevant.

The presented results due to low number of participants cannot be statistically significant. Nevertheless, all of the participants originated from different organizations, did not share opinions on mutual responses to survey and had different work background.

6.7 The overall evaluation of the designed model

In order to evaluate the developed model as well as the assumptions followed, the experiment with practitioners was designed and performed. The scenario of the experiment was carefully planned according to the followed research methodology. The scenario encompasses the following phases:

- Design of the experiment group.
- LCV construction based on provided data and additional changes performed from the perspective of the practitioners.

Table 6.10: Results of the evaluation of the LCA. Tasks 1 to 6.

Task	KPI	P1	P2	P3	P4	P5	P6	P7	P8	Avg
1	S	8	8	7	8	8	4	5	7	6.88
	N	2	6	4	9	9	8	3	5	5.75
2	S	10	9	1	2	4	2	10	10	6.00
	N	10	8	4	10	3	7	8	6	7.00
3	S	10	10	10	10	10	7	10	10	9.63
	N	10	9	10	10	10	1	10	9	8.63
4	S	1	6	3	7	2	10	8	6	5.38
	N	1	3	4	8	1	10	7	6	5.00
5	S	8	7	8	9	7	7	9	8	7.88
	N	3	4	4	10	5	3	10	6	5.63
6	S	9	8	8	7	4	7	8	7	7.25
	N	5	3	6	9	2	6	9	7	5.88
Participant avg	S	7.7	8	6.2	7.2	5.8	6.2	8.3	8.0	**7.17**
Participant avg	N	5.2	5.5	5.3	9.3	5.0	5.8	7.8	6.5	**6.31**

- Use of LCV to annotate a set of Web service operations.
- Retrieval of a desired set of Web service operations using LCA as the main mechanism.
- Comparative study of ontology design and use.

First of all, there were four participants, two with technical background and two strictly business oriented. The main emphasis was laid onto a good mix of skills and expertise, the overall evaluation of the designed model aimed to receive response from both the technical and non-technical users. The size of the group was limited yet the skills and expertise of the participants were very wide. The participants consisted of middle-executive, business analyst, consultant and IT specialist.

They had to work with the preprocessed data on terms relevant to communication Web services. The terms handled by the responsible mechanisms originated from manuals, forums and available documentation.

The shortlist of terms amounted to 231 terms. The initial task before the participants was to assign terms to each of the phrase categories. They were instructed as to the nature and goal of each of the phrases. The instruction consisted of the syntax of the used model, and a short description of the objectives of each of the phrases. It was encouraged to mark terms that were to be left unused as a part of any of the three sets representing description phrases. In the next step, the participants were asked to add any terms that in their opinion are crucial for the description of the

domain and were missing from the initial list. Next task, was to decide whether there was any compound terms that could be produced from the already partitioned sets.

Having accomplished this, the participants were asked to describe a number of Web service operations delivered by the author. The description were to contain elements of the phrase sets that were prepared by them.

The prepared descriptions were used as a test repository were LCA was a key element that was responsible for the Web service operations' retrieval. In order to simulate a situation where one party prepares descriptions and other party retrieves them with specific queries, the repositories were interchanged among the participants in following manner:

- every business user was to perform a search on a repository assembled by technically inclined users and the other business user's repository,
- every technically inclined user was to perform search on a repository assembled by business users and the other technically inclined's repository.

In total, every participant performed three searches. He was to rate every term in terms of a level of overall satisfaction and a level of noise (scales 1 to 10, 1 extremely bad, 10 excellent).

Due to the common base of the separately prepared LCVs and well working LCA the results of this part of experiment demonstrated that the average of the overall satisfaction level was 9.01 and the average for the level of noise was 8.75. These are considerably better results from those achieved in the evaluation of LCA based on the corpora from the open Internet.

After the evaluation of effectiveness, the participants were introduced to the Ontology Development 101[12]. In order to fully check whether they comprehend the document, they were asked to prepare an ontology on communication based on their ideas, experience and data provided in the terms list used in the experiment. They were asked to use Protege[13] as the tool chosen by the authors of the Ontology Development 101. When ready with the ontology, they were asked to describe Web service operations in terms of ontology concepts they defined.

The total time allocated to the task was 8 hours. After this time, the participants were asked which method of Web service operation description took less time to accomplish. Which method they preferred. What were the advantages of the chosen method.

[12] http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html

[13] http://protege.stanford.edu/

The results were as follows:

- technically inclined users preferred the first method mainly due to a fact that it was much faster to allocate terms to one or more phrase sets and then it was straightforward to assign phrase elements to actual Web service operations,
- technically inclined users noted that ontology building is much more comprehensive than the first method and that it might not bring much better results than the first method, thus it is not worth using it with the task at hand
- business users found the first method to be very approachable, whereas the ontology building contained too much concepts that they were unaware of, thus making it incomprehensible,
- members of both groups emphasised that the time necessary for building of the phrase sets and constructing the descriptions with the first method is considerably shorter than ontology comprehension, building and description,
- both parties acknowledged that the proposed model is of high quality and streamlines the tasks they were subjected to.

Due to the small sample of participants, it is not possible to draw statistically valid results. Nevertheless, the obtained data demonstrate that the ontology building is deemed as complex and not justifying the effort in comparison to the model presented in this dissertation.

The experiment performed was the most comprehensive one in terms of the presented model as it required the participants to step through the major steps of Web service description and discovery. Thanks to it, it was further reassured that the proposed model is effective at lower cost than the semantic based approach. The cost is as previously, a function of time spent of familiarization with the description methodology, time necessary to prepare the description resources and time required to annotate individual Web services. The actual value is dependant of wages offered to actual implementers.

6.8 Summary

The verification was performed through experiments and surveys. The elements of model and mechanisms that could be isolated were thoroughly tested using a considerable number of data, both generated and actual in order to measure a number of indicators. These indicators were used to answer whether the model is scalable and whether its performance in terms of execution time is acceptable.

The elements that could not be isolated were researched thanks the cooperation with a number of professionals lending their time and expertise in order to assess them.

The experiments where the designed methods supporting the proposed modern Web service description and retrieval model were used allow for a validation of the dissertation's thesis. Both the measurable and unmeasurable results captured with help of panel study and specially designed experiments, demonstrate a considerable improvement regarding the Key Requirement Aspects which are in direct relationship of the quality of the description and retrieval process as perceived by the interested parties.

Chapter 7

Conclusions

The main goal of this work was to validate the following thesis: **The modern approach for Web service description and retrieval derived and rectified from the state of the art solutions shall increase quality of the retrieval process in comparison to the available means in concordance with the identified requirements of organizations implementing the Service Oriented Architecture paradigm.**

The thesis was validated thanks to a rigourously driven research activities that were initiated by the thorough analysis of available solutions and opinions in realm of Web service description and discovery. This analysis led to formulation of the postulates that are to be met when discussing a viable solution. The proposed model is driven by this postulates and so are the designed mechanisms that fulfill the model's features.

7.1 Main results of the research activities

The whole effort summarised here was organised by the research methodology discussed in the beginning of this work. The most influential part of the methodology was the Concept-Knowledge Theory followed by the Design Science. The C-K Theory allowed the author to organize the research process and make it comprehensible, especially that one had to address not only technical aspects, but more challenging ones such as grasping the functionality description task in terms of economics and its complexity for users and their organisations. The design science paradigm allowed for neat organisation of various results into artifacts that adhere to the structure proposed by [Hevner et al., 2004]. Finally, the most classic methodology toolset of a

researcher as given by [Lakatos, 1978] proved to be invaluable at each an every step of the research process.

The presented work satisfies the main research goals stated in 1.2. The postulated model is a cost effective alternative to solutions based on semantic annotation technologies. In addition, thanks to its straightforwardness it is a viable competitor for the Information Retrieval based models as it allows for the purpose statement with low overhead in terms of additional effort of organizations' members.

What is more, it successfully addresses all of the Key Aspect Requirements defined in 2.6. Therefore, the presented model is a valuable proposition for Service Oriented Enterprises that need to manage their internal repositories without investments in expensive, in terms of time that hast to be invested into learning and implementing it, semantic technologies.

The model was built taking into account detailed analysis of available approaches. From a body of over one hundred relevant scientific publications, over forty were chosen for analysis. The analysis was driven by the already introduced Key Requirement Aspects. The used publications were classified into three groups that cover the diversity of the Web service description and retrieval domain. Every classified approach was evaluated, and the results served as a base for building the overview of the whole domain. The obtained overview corresponds with the observations submitted by users originating from various business environments. None of the general approaches satisfies all of the key requirement aspects in satisfactory manner.

The proposed model was evaluated through a number of experiments oriented on capturing pure performance and opinions on its robustness expressed by experiments' participants. The setup of the experiments along with their results were given and discussed in detail.

It is believed that the unique effort to capture requirements from broader range of potential Web service users should make the results presented here potentially successful when they are instantiated. As throughout the research activities, the author gathered feedback at various research events, it is necessary to once more underline that of paramount importance, is the description of pure functionality. It differs from all the mainstream solutions as they are oriented on the description of technical elements of a Web service.

The description of inputs and outputs does not determine the functionality perceived by the end user. While this can be useful to developers and programmers skilled with programming abstractions and coding techniques, a business user is lost when he has to express his needs in terms of concepts from an ontology. One has

also remember that programmers usually infer on procedure function using its name. When this is insufficient they refrain to examination of the source code.

This is not possible in the SOA paradigm where Web services are treated as black boxes. Therefore, it is felt that the proposed model brings a much desired change of balance between a technical description and actually desired traits by non-technical personnel. These traits are a projection of a simple need to be able to retrieve efficiently a desired functionality. By efficiently one has to understand a situation, where a user is presented with the available functionalities that match his requirements encoded in a query in a timely fashion with the least necessary effort considering that the search space can be of considerable size reaching tens of millions of items.

7.2 The contribution of the proposed model to the economics of information

As stated in the first part of the work, the issue discussed here is of great importance to economics of information. Its main goal is to ameliorate the state of affairs in the Web service description and retrieval areas. As Web services are splendid examples of information goods, the traditional ways of retrieval are not sufficient when applied to them, and the semantic-based ones impose too much effort on the organisations and their users. Thus, it is believed that a user has two manners of finding a desired Web service that both fail his expectations due to specific reasons. Therefore, he is unable to cope with the uncertainty regarding the fact whether he was presented with all of the viable propositions.

In essence this work proposes a model that under a set of constraints offers a solution that improves the situation regarding the level of uncertainty when making a choice. What is more, it does so with lesser expenses in terms of time spend both on executing the model based solution and implementing it. The proposed model is a tool that allows to robustly evaluate the utility function of any set of Web services described in proposed manner. It excels in providing a straightforward answer to a question of whether there are some Web service operations that satisfy a user functionality description. What is more, it does yield a positive results even when the functional description is not given in the form defined in the model thanks to a set of mechanisms.

The fact that a Web service is considered here as a information good rather than a specific kind of computer-processable document, the effort to analyse available alternatives in terms of various economic categories, and the global effort to provide a model that allows for maximisation of user's perceived utility function are the most important economic aspects of this work.

7.3 Future plans and open issues

The proposed model along with other artifacts is a tremendous opportunity for a fully-fledged implementation in an organisation willing to leverage the whole potential of the SOA paradigm along with empowering its users with a set of solutions making their work easier and less costly. Such implementation would enable one to observe all of the artifacts in full usage and would add a considerable trove of data that could further refine the model presented here.

An implementation in an organization that operates according to the SOA principle could provide an invaluable trove of data in which direction optimization of the designed mechanisms should be driven. The sheer number of Sub-organisational Units and the number of processes in an organisation could provide a considerable amount of data on user preferences and the way they make choices regarding Web services. One would like to know what strategy is the most worthwhile in user's opinion, at a scale of tens of thousands employed in some organisation. Many sources suggest that the killer application in majority of electronic markets is a simple search tool that provides answers as fast as possible and tries to handle the implicit uncertainty present in the user's query. Yet, despite the designed openness of the model to various access methods, the nature of an organization willing to implement the model might induce a novel approach or choose some other predefined scenario.

In addition, it is believed that the further research on the user motivation while performing choice of any given entity should be invested in. As there are comprehensive studies on users interacting with traditional retrieval engines in form of Web searches, one would like to be able to present a comparable resources on the motivation behind functionality choice. This could be applied in a number of domains, as it is very common to search for entities in order to find ones that perform some service, rather than for those that contain a certain set of words.

Bibliography

[met, 2004] (2004). *Understanding Metadata*. NISO Press.

[Akkiraju et al., 2005] Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M.-T., Sheth, A., and Verma, K. (2005). Web service semantics - wsdl-s. *International Business*, 2008(Version 1.0):1–42.

[Al-Masri and Mahmoud, 2008] Al-Masri, E. and Mahmoud, Q. H. (2008). Investigating web services on the world wide web. *Proceeding of the 17th international conference on World Wide Web WWW 08*, 32(3):795.

[Alam et al., 2010] Alam, M., Zhang, X., Nauman, M., Khan, S., and Alam, Q. (2010). Mauth: A fine-grained and user-centric permission delegation framework for multi-mashup web services. *2010 6th World Congress on Services*, pages 56–63.

[Allen, 2000] Allen, B. (2000). The future of microeconomic theory. *Journal of Economic Perspectives*, 14(1):143–150.

[Allen et al., 1990] Allen, B., of Pennsylvania. Center for Analytic Research in Economics, U., and the Social Sciences (1990). *Information as an Economic Commodity*. University of Pennsylvania, Center for Analytic Research in Economics and the Social Sciences.

[Alonso, 2003] Alonso, G. (2003). Web services. *The Service-Oriented Media Enterprise*, 1(3):309–320.

[Amsler, 1984] Amsler, R. A. (1984). Lexical knowledge bases. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*, ACL '84, pages 458–459, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Anadiotis et al., 2009] Anadiotis, G., Kotoulas, S., Lausen, H., and Siebes, R. (2009). Massively scalable web service discovery. In *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications*, AINA '09, pages 394–402, Washington, DC, USA. IEEE Computer Society.

[Andrikopoulos and Plebani, 2011] Andrikopoulos, V. and Plebani, P. (2011). Retrieving compatible web services. *Web Services, IEEE International Conference on*, 0:179–186.

[Anh and Moffat, 2005] Anh, V. N. and Moffat, A. (2005). Inverted index compression using word-aligned binary codes. *Inf. Retr.*, 8(1):151–166.

[Antoniou and Van Harmelen, 2004] Antoniou, G. and Van Harmelen, F. (2004). Owl web ontology language. *Ubiquity*, 2007(September):1–1.

[Aouiche et al., 2004] Aouiche, K., Lemire, D., and Godin, R. (2004). Collaborative olap with tag clouds. *Most*, pages 1–8.

[Arampatzis and Kamps, 2008] Arampatzis, A. and Kamps, J. (2008). A study of query length. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 08*, pages 811–812.

[Arrow, 1984] Arrow, K. J. (1984). *Collected Papers of Kenneth J. Arrow, Volume 4: The Economics of Information: The Economics of Information*. Collected Papers of Kenneth J. Arrow, Volume Four. Belknap Press.

[Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. *The Semantic Web*, 4825(Springer):722–735.

[Averbakh et al., 2009] Averbakh, A., Krause, D., and Skoutas, D. (2009). Exploiting user feedback to improve semantic web service discovery. *The Semantic Web-ISWC 2009*, page 33–48.

[Baader and Sattler, 2001] Baader, F. and Sattler, U. (2001). An overview of tableau algorithms for description logics. *Studia Logica*, 69(1):5–40.

[Baeza-Yates, 2004] Baeza-Yates, R. (2004). A fast set intersection algorithm for sorted sequences. *Lecture Notes in Computer Science*, 3109:400–408.

[Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*, volume 463. Addison Wesley.

[Baghdadi, 2012] Baghdadi, Y. (2012). A framework to select an approach for web services and soa development. In *Innovations in Information Technology (IIT), 2012 International Conference on*, pages 277 –282.

[Bakos, 1998] Bakos, Y. (1998). The emerging role of electronic marketplaces on the internet. *Communications of the ACM*, 41(8):35–42.

[Bakos et al., 1999] Bakos, Y., Brynjolfsson, E., and Lichtman, D. (1999). *Shared Information Goods*. John M. Olin Program in Law & Economics working paper. University of Chicago Law School.

[Barbour and Luczak, 2008] Barbour, A. D. and Luczak, M. J. (2008). Laws of large numbers for epidemic models with countably many types. *The Annals of Applied Probability*, 18(6):2208–2238.

[Barros and Dumas, 2006] Barros, A. P. and Dumas, M. (2006). The rise of web service ecosystems. *It Professional*, 8(5):31–37.

[Bashir et al., 2010] Bashir, S., Khan, F. H., Javed, M. Y., Khan, A., and Khiyal, M. S. H. (2010). Indexer based dynamic web services discovery. *Journal of Computer Science*, 7(2):153–159.

[Basu et al., 2008] Basu, S., Casati, F., and Daniel, F. (2008). Toward web service dependency discovery for soa management. In *Services Computing, 2008. SCC '08. IEEE International Conference on*, volume 2, pages 422 –429.

[Battle et al., 2005] Battle, S., Bernstein, A., Boley, H., Grosof, B., Gruninger, M., Hull, R., Kifer, M., Martin, D., McIlraith, S., McGuinness, D., and et al. (2005). Semantic web services framework (swsf) overview. *W3C Member Submission*, 2009(09/20).

[Begg et al., 2008] Begg, D. K., Fischer, S., and Dornbusch, R. (2008). *Economics*. McGraw-Hill higher education. McGraw-Hill.

[Belkin et al., 2003] Belkin, N. J., Kelly, D., Kim, G., Kim, J. Y. Y., Lee, H. J., Muresan, G., Tang, M. C. M., Yuan, X., and Cool, C. (2003). Query length in interactive information retrieval. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval SIGIR 03*, page 205.

[Benson et al., 2006] Benson, E., Wasson, G., and Humphrey, M. (2006). Evaluation of uddi as a provider of resource discovery services for ogsa-based grids. In *Proceedings of the 20th international conference on Parallel and distributed processing*, IPDPS'06, pages 36–36, Washington, DC, USA. IEEE Computer Society.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

[Bhiri et al., 2009] Bhiri, S., Gaaloul, W., Rouached, M., and Hauswirth, M. (2009). Semantic web services for satisfying soa requirements. In Dillon, T., Chang, E., Meersman, R., and Sycara, K., editors, *Advances in Web Semantics I*, volume 4891 of *Lecture Notes in Computer Science*, pages 374–395. Springer Berlin / Heidelberg.

[Bia and Kalika, 2007] Bia, M. and Kalika, M. (2007). Adopting an ict code of conduct: An empirical study of organizational factors. *Journal of Enterprise Information Management*, 20(4):432–446.

[Bizer and Schultz, 2009] Bizer, C. and Schultz, A. (2009). The berlin sparql benchmark. *Group*, 5(2):1–24.

[Blondel et al., 2004] Blondel, V., Gajardo, A., Heymans, M., Senellart, P., and Van Dooren, P. (2004). A measure of similarity between graph vertices. *Technology*, page 647–666.

[Boley, 2006] Boley, H. (2006). The ruleml family of web rule languages. *Language*, (June):1–17.

[Bravo et al., 2008] Bravo, M., Montes, A., and Reyes, A. (2008). Natural language processing techniques for the extraction of semantic information in web services. *2008 Seventh Mexican International Conference on Artificial Intelligence*, pages 53–57.

[Brewster et al., 2009] Brewster, C., Jupp, S., Luciano, J., Shotton, D., Stevens, R. D., and Zhang, Z. (2009). Issues in learning an ontology from text. *BMC Bioinformatics*, 10(Suppl 5):S1.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a search engine.

[Brown et al., 1992] Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479.

[Bruijn, 2005] Bruijn, J. D. (2005). The web service modeling language wsml. *W3C Member Submission*, (October):590–604.

[Bruno et al., 2005] Bruno, M., Canfora, G., Penta, M. D., and Scognamiglio, R. (2005). An approach to support web service classification and annotation. *2005 IEEE International Conference on eTechnology eCommerce and eService*, pages 138–143.

[Brynjolfsson and Hitt, 2000] Brynjolfsson, E. and Hitt, L. M. (2000). Beyond computation: Information technology, organizational transformation and business performance. *Journal of Economic Perspectives*, 14(4):23–48.

[Business, 2001] Business, I. (2001). Uddi executive white paper. *International Business*.

[Buxmann, 2009] Buxmann, P. (2009). Internet of services. *Business Information Systems Engineering*, 1(5):341–342.

[Cabral et al., 2004] Cabral, L., Domingue, J., Motta, E., Payne, T. R., and Hakimpour, F. (2004). Approaches to semantic web services: an overview and comparisons. In *ESWS*, pages 225–239.

[Calvanese, 1996] Calvanese, D. (1996). *Finite model reasoning in description logics*, page 292–303. MORGAN KAUFMANN PUBLISHERS.

[Cardoso, 2007] Cardoso, J. (2007). The semantic web vision: Where are we? *IEEE Intelligent Systems*, 22(5):84–88.

[Cardoso et al., 2010] Cardoso, J., Barros, A., May, N., and Kylau, U. (2010). *Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments*, page 602–609. IEEE.

[Cattuto et al., 2007] Cattuto, C., Baldassarri, A., Servedio, V. D. P., and Loreto, V. (2007). Vocabulary growth in collaborative tagging systems. *arXiv*, 704(2006):6.

[Ceglarek et al., 2010] Ceglarek, D., Haniewicz, K., and Rutkowski, W. (2010). Quality of semantic compression in classification. In *ICCCI (1)*, pages 162–171.

[Celik and Elci, 2006] Celik, D. and Elci, A. (2006). Discovery and scoring of semantic web services based on client requirement(s) through a semantic search agent. *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, pages 273–278.

[Chappell, 2004] Chappell, D. A. (2004). *Enterprise Service Bus*, volume 4. O'Reilly Media, Inc.

[Cheng et al., 2007] Cheng, S., Chang, C. K., Zhang, L.-J., and Kim, T.-H. (2007). *Towards Competitive Web Service Market*, pages 213–219.

[Choi et al., 2010] Choi, J., Nazareth, D. L., and Jain, H. K. (2010). Implementing service-oriented architecture in organizations. *Journal of Management Information Systems*, 26(4):253–286.

[Chowdhury, 2004] Chowdhury, G. (2004). *Introduction to modern information retrieval*, volume 23. Facet Publishing.

[Christensen et al., 2001] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). Web service definition language (wsdl).

[Christiansen and Kirby, 2003] Christiansen, M. H. and Kirby, S., editors (2003). *Language evolution*. Oxford University Press, Oxford New York.

[Chu-Carroll and Prager, 2007] Chu-Carroll, J. and Prager, J. (2007). An experimental study of the impact of information extraction accuracy on semantic search performance. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management CIKM 07*, page 505.

[Chukmol et al., 2008] Chukmol, U., Benharkat, A.-N., and Amghar, Y. (2008). Enhancing web service discovery by using collaborative tagging system. *2008 4th International Conference on Next Generation Web Services Practices*, pages 54–59.

[Cleverley, 2001] Cleverley, W. O. (2001). Financial dashboard reporting for the hospital industry. *Journal of Health Care Finance*, 27(3):30–40.

[Colgrave et al., 2004] Colgrave, J., Akkiraju, R., and Goodwin, R. (2004). External matching in uddi. pages 226–.

[Colucci et al., 2003] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F. M., and Mongiello, M. (2003). *Description logics approach to semantic matching of Web services*, volume 11. Univ. Zagreb.

[Conte et al., 2010] Conte, T., Blau, B., and Xu, Y. (2010). *Competition of Service Marketplaces – Designing Growth in Service Networks.*

[Conti et al., 2002] Conti, M., Kumar, M., Das, S. K., and Shirazi, B. A. (2002). Quality of service issues in internet web services. *IEEE Trans. Comput.*, 51(6):593–594.

[Conway, 1968] Conway, M. E. (1968). How do committees invent? *Datamation*, 14(4):28–31.

[Costa et al., 2009] Costa, P., Zahn, T., Rowstron, A., O'Shea, G., and Schubert, S. (2009). Why should we integrate services, servers, and networking in a data center? *Proceedings of the 1st ACM workshop on Research on enterprise networking WREN 09*, page 111.

[Crasso et al., 2010] Crasso, M., Rodriguez, J. M., Zunino, A., and Campo, M. (2010). Revising wsdl documents: Why and how. *IEEE Internet Computing*, 14(5):48–56.

[Cuadrado et al., 2008] Cuadrado, F., García, B., Dueñas, J. C., and Parada, H. A. (2008). A case study on software evolution towards service-oriented architecture. In *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops*, AINAW '08, pages 1399–1404, Washington, DC, USA. IEEE Computer Society.

[Dachman-Soled et al., 2009] Dachman-Soled, D., Malkin, T., Raykova, M., and Yung, M. (2009). *Efficient Robust Private Set Intersection*, volume 5536, pages 125–142. Springer Berlin Heidelberg.

[Davis and Goadrich, 2006] Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. *Proceedings of the 23rd international conference on Machine learning ICML 06*, 10(2):233–240.

[Debreu, 1959] Debreu, G. (1959). *Theory of Value: An Axiomatic Analysis of Economic Equilibrium.* Number t. 17 in Monograph. John Wiley.

[Denning et al., 2005] Denning, P., Horning, J., Parnas, D., and Weinstein, L. (2005). Wikipedia risks. *Communications of the ACM*, 48(12):152.

[D'Mello and Ananthanarayana, 2009] D'Mello, D. A. and Ananthanarayana, V. S. (2009). A tree structure for efficient web service discovery. *2009 Second International Conference on Emerging Trends in Engineering & Technology*, pages 826–831.

[D'Mello and Ananthanarayana, 2010] D'Mello, D. A. and Ananthanarayana, V. S. (2010). A review of dynamic web service description and discovery techniques. *2010 First International Conference on Integrated Intelligent Computing*, pages 246–251.

[Eatwell et al., 2000] Eatwell, J., Milgate, M., and Newman, P. (2000). *The New Palgrave: A Dictionary of Economics: Four Volume Boxed Set*. The New Palgrave : a dictionary of economics. Palgrave Macmillan.

[Eppler and Mengis, 2003] Eppler, M. J. and Mengis, J. (2003). A framework for information overload research in organizations. *Organization*, (September):1–42.

[Erl, 2005] Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design By Thomas Erl*.

[Espinoza and Mena, 2007] Espinoza, M. and Mena, E. (2007). Discovering web services using semantic keywords. *2007 5th IEEE International Conference on Industrial Informatics*, 2:725–730.

[Faye et al., 2012] Faye, D. C., Cure, O., and Blin, G. (2012). A survey of rdf storage approaches. *Revue Africaine de la Recherche en Informatique et Mathematiques Appliquees*, 15(1):11–35.

[Fehr and Tyran, 2005] Fehr, E. and Tyran, J.-R. (2005). Individual irrationality and aggregate outcomes. IEW - Working Papers iewwp252, Institute for Empirical Research in Economics - University of Zurich.

[Fellbaum, 1999] Fellbaum, C. (1999). *Organization of Verbs in a Semantic Net*, pages 93–109. Kluwer.

[Feng and Fan, 2012] Feng, X. and Fan, Y. (2012). Research on application of rdfa in restful web services. In *Internet Computing for Science and Engineering (ICICSE), 2012 Sixth International Conference on*, pages 266 –269.

[Ferreira et al., 2011] Ferreira, J. J., Araujo, R. M., and Baiao, F. A. (2011). Identifying ruptures in business-it communication through business models. In Filipe, J., Cordeiro, J., Aalst, W., Mylopoulos, J., Rosemann, M., Shaw, M. J., and Szyperski, C., editors, *Enterprise Information Systems*, volume 73 of *Lecture Notes in Business Information Processing*, pages 311–325. Springer Berlin Heidelberg.

[Fielding, 2000] Fielding, R. T. (2000). Chapter 5 representational state transfer (rest). *Architectural Styles and the Design of Networkbased Software Architectures Doctoral dissertation University of California Irvine*, pages 76–106.

[Fox, 1989] Fox, C. (1989). A stop list for general text. *SIGIR Forum*, 24(1-2):19–21.

[Francez, 1982] Francez, N. (1982). *Extended naming conventions for communicating processes*, pages 40–45. ACM.

[Freiden et al., 1998] Freiden, J., Goldsmith, R., Takacs, S., and Hofacker, C. (1998). Information as a product: not goods, not services. *Marketing Intelligence & Planning*, 16(3):210–220.

[Fu et al., 2005] Fu, X., Bultan, T., and Su, J. (2005). Synchronizability of conversations among web services. *IEEE Transactions on Software Engineering*, 31(12):1042–1055.

[Gacitua et al., 2007] Gacitua, R., Sawyer, P., Piao, S., and Rayson, P. (2007). Ontology acquisition process: A framework for experimenting with different nlp techniques. *Artificial Intelligence*.

[Galle et al., 2008] Galle, D., Kop, C., and Mayr, H. C. (2008). A uniform web service description representation for different readers. *Second International Conference on the Digital Society*, pages 123–128.

[Gao et al., 2009] Gao, H., Stucky, W., and Liu, L. (2009). Web services classification based on intelligent clustering techniques. *2009 International Forum on Information Technology and Applications*, pages 242–245.

[Gardiner et al., 2006] Gardiner, T., Horrocks, I., and Tsarkov, D. (2006). Automated benchmarking of description logic reasoners. *Most*, 189:167–174.

[Garrod et al., 2008] Garrod, C., Manjhi, A., Ailamaki, A., Maggs, B., Mowry, T., Olston, C., and Tomasic, A. (2008). Scalable query result caching for web applications. *Management*, 1(1):550–561.

[Gelman and Butterworth, 2005] Gelman, R. and Butterworth, B. (2005). Number and language: how are they related? *Trends in Cognitive Sciences*, 9(1):6–10.

[Geoffrion and Maturana, 1995] Geoffrion, A. and Maturana, S. (1995). Generating optimization-based decision support systems. In *Proceedings of the 28th Hawaii International Conference on System Sciences*, HICSS '95, pages 439–, Washington, DC, USA. IEEE Computer Society.

[Geurts, 1997] Geurts, B. (1997). Good news about the description theory of names. *Journal of Semantics*, 14:319–348.

[Giles, 2005] Giles, J. (2005). Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901.

[Günther et al., 2007] Günther, O., Tamm, G., and Leymann, F. (2007). Pricing web services. *International Journal Business Process Integration and Management*, 2(2):132–140.

[Government, 2005] Government, A. (2005). Naming conventions for electronic documents. *Order A Journal On The Theory Of Ordered Sets And Its Applications*.

[Guizzardi, 2006] Guizzardi, G. (2006). On ontology, ontologies, conceptualizations, modeling languages, and (meta)models. volume 155 of *Frontiers in Artificial Intelligence and Applications*, pages 18–39. IOS Press.

[Guo et al., 2004] Guo, Y., Pan, Z., and Heflin, J. (2004). An evaluation of knowledge base systems for large owl datasets. *International Semantic Web Conference*, 3298(329):274–288.

[Guo et al., 2005] Guo, Y., Pan, Z., and Heflin, J. (2005). Lubm: A benchmark for owl knowledge base systems. *Web Semantics Science Services and Agents on the World Wide Web*, 3(2-3):158–182.

[Gupta et al., 2010] Gupta, M., Li, R., Yin, Z., and Han, J. (2010). Survey on social tagging techniques. *SIGKDD Explor. Newsl.*, 12(1):58–72.

[Haase and Nagl, 2008] Haase, T. and Nagl, M. (2008). Service-oriented architectures and application integration. pages 727–740.

[Hadar and Fox, 2009] Hadar, L. and Fox, C. R. (2009). Information asymmetry in decision from description versus decision from experience. *SciencesNew York*, 4(4):317–325.

[Hagemann et al., 2007] Hagemann, S., Letz, C., and Vossen, G. (2007). Web service discovery - reality check 2.0. In *Proceedings of the Third International Conference on Next Generation Web Services Practices*, NWESP '07, pages 113–118, Washington, DC, USA. IEEE Computer Society.

[Haidar and Abdallah, 2009] Haidar, A. N. and Abdallah, A. E. (2009). Abstractions of web services. *2009 14th IEEE International Conference on Engineering of Complex Computer Systems*, pages 182–191.

[Haller et al., 2005a] Haller, A., Cimpian, E., Mocan, A., Oren, E., and Bussler, C. (2005a). Wsmx - a semantic service-oriented architecture. In *Proceedings of the IEEE International Conference on Web Services*, ICWS '05, pages 321–328, Washington, DC, USA. IEEE Computer Society.

[Haller et al., 2005b] Haller, A., Gomez, J. M., and Bussler, C. (2005b). Exposing semantic web service principles in soa to solve eai scenarios.

[Hansen et al., 2003] Hansen, M., Madnick, S., and Siegel, M. (2003). Data integration using web services. *Efficiency and Effectiveness of XML Tools and Techniques and Data Integration over the Web VLDB 2002 Workshop EEXTT and CAiSE 2002Workshop DIWeb*, 2590:165–182.

[Harshavardhanan et al., 2012] Harshavardhanan, P., Akilandeswari, J., and Sarathkumar, R. (2012). Dynamic web services discovery and selection using qos-broker architecture. In *Computer Communication and Informatics (ICCCI), 2012 International Conference on*, pages 1 –5.

[Hatchuel and Weil, 2008] Hatchuel, A. and Weil, B. (2008). C-k design theory: an advanced formulation. *Research in Engineering Design*, 19(4):181–192.

[Hausmann et al., 2004] Hausmann, J. H., Heckel, R., and Lohmann, M. (2004). Model-based discovery of web services. *Proceedings. IEEE International Conference on Web Services, 2004.*, pages 324–331.

[Heal, 1999] Heal, G. (1999). Valuing ecosystem services. Papers 98-12, Columbia - Graduate School of Business.

[Hecht and Ullman, 1974] Hecht, M. S. and Ullman, J. D. (1974). Characterizations of reducible flow graphs. *Journal of the ACM*, 21(3):367–375.

[Herrmann et al., 2007] Herrmann, M., Dalferth, O., and Aslam, M. A. (2007). Applying semantics (wsdl, wsdl-s, owl) in service oriented architectures (soa). *10th Intl Protégé Conference*, pages 1–3.

[Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

[Heylighen, 2004] Heylighen, F. (2004). Complexity and information overload in society: why increasing efficiency leads to decreasing control. *BULLETIN OF THE MEDICAL LIBRARY ASSOCIATION*, 87:2.

[Hilbert and Lopez, 2011] Hilbert, M. and Lopez, P. (2011). The world's technological capacity to store, communicate, and compute information. *Science*, 60(6025):60–65.

[Hitzler et al., 2009] Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., and Rudolph, S. (2009). Owl 2 web ontology language primer. *W3C Recommendation*, 27(October):1–123.

[HOLMAN RECTOR, 2008] HOLMAN RECTOR, L. (2008). Comparison of wikipedia and other encyclopedias for accuracy, breadth, and depth in historical articles. *Reference Services Review*, 36(1):7–22.

[Hommes and Reijswoud, 1999] Hommes, B.-J. and Reijswoud, V. v. (1999). Assesing the quality of business process modelling techniques. *Conference on Information Systems Concepts*, pages 20–22.

[Hoonlor et al., 2012] Hoonlor, A., Szymanski, B. K., Zaki, M. J., Thompson, J., and Thompson, J. (2012). An evolution of computer science research.

[Horrocks et al., 2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., and Dean, M. (2004). Swrl: A semantic web rule language combining owl and ruleml. *Syntax*, 21(May):1–22.

[Horrocks and Tessaris, 2002] Horrocks, I. and Tessaris, S. (2002). Querying the semantic web: a formal approach. *Proc of the 13th Int Semantic Web Conf ISWC2002*, 2342(2342):177–191.

[Hoxmeier and DiCesare, 2000] Hoxmeier, J. A. and DiCesare, C. (2000). System response time and user satisfaction: An experimental study of browser-based applications. *Proceedings of the Association of Information*, (2):1–26.

[Hu et al., 2009] Hu, L., Ying, S., Zhao, K., and Chen, R. (2009). A semantic web service description language. *2009 WASE International Conference on Information Engineering*, pages 449–452.

[Huang et al., 2005] Huang, C.-l., Lo, C.-c., Li, Y., and Chao, K.-m. (2005). Service discovery through multi-agent consensus. *IEEE International Workshop on Service-Oriented System Engineering (SOSE'05)*, pages 37–44.

[Hutchison et al., 2001] Hutchison, D., Wolf, L., and Steinmetz, R. (2001). Quality of service. *Image Rochester NY*, 6(2):43–55.

[Ingason et al., 2008] Ingason, A. K., Helgadóttir, S., Loftsson, H., and Rögnvaldsson, E. (2008). A mixed method lemmatization algorithm using a hierarchy of linguistic identities (holi). In *Proceedings of the 6th international conference on Advances in Natural Language Processing*, GoTAL '08, pages 205–216, Berlin, Heidelberg. Springer-Verlag.

[Iqbal et al., 2008] Iqbal, K., Sbodio, M. L., Peristeras, V., and Giuliani, G. (2008). Semantic service discovery using sawsdl and sparql. *2008 Fourth International Conference on Semantics, Knowledge and Grid*, pages 205–212.

[James E. Shirt and Baru, 2011] James E. Shirt, R. E. B. and Baru, C. (2011). How much information? 2010 report on enterprise server information.

[Jansen et al., 2007] Jansen, B. J., Spink, A., and Koshman, S. (2007). Web searcher interaction with the dogpile.com metasearch engine. *J. Am. Soc. Inf. Sci. Technol.*, 58(5):744–755.

[Jarrar et al., 2003] Jarrar, M., Demey, J., Meersman, R., Spaccapietra, S., March, S. T., and Aberer, K. (2003). On using conceptual data modeling for ontology engineering. *October*, 2800(October):185–207.

[Jensen, 1998] Jensen, M. C. (1998). Organization Theory and Methodology. *Social Science Research Network Working Paper Series*.

[Jiang et al., 2012] Jiang, D., Xue, J., and Xie, W. (2012). A reputation model based on hierarchical bayesian estimation for web services. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*, pages 88 –93.

[Jin et al., 2007] Jin, B., Zhang, L., and Zang, Z. (2007). A unified service discovery framework. *Sixth International Conference on Grid and Cooperative Computing (GCC 2007)*, (2006):203–209.

[Jin and Liu, 2006] Jin, Z. and Liu, L. (2006). Web service retrieval: An approach based on context ontology. *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, pages 513–520.

[Jónsson et al., 2006] Jónsson, B., Arinbjarnar, M., Tórsson, B., Franklin, M. J., and Srivastava, D. (2006). Performance and overhead of semantic cache management. *ACM Trans. Internet Technol.*, 6(3):302–331.

[Jordan and Alves, 2007] Jordan, D. and Alves, A. (2007). Web services business process execution language version 2 . 0. *Language*, 11(April):1–264.

[Karimpour and Taghiyareh, 2009] Karimpour, R. and Taghiyareh, F. (2009). Conceptual discovery of web services using wordnet. *2009 IEEE Asia-Pacific Services Computing Conference (APSCC)*, pages 440–444.

[Khare and Çelik, 2006] Khare, R. and Çelik, T. (2006). *Microformats: a pragmatic path to the semantic web*, pages 865–866. Number January. ACM Press.

[Khdour and Fasli, 2010] Khdour, T. and Fasli, M. (2010). A semantic-based web service registry filtering mechanism. *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, pages 373–378.

[Kirby, 1998] Kirby, S. (1998). Language evolution without natural selection: From vocabulary to syntax in a population of learners. *Edinburgh Occasional Paper in Linguistics EOPL981*, (Edinburgh Occasional Papers in Linguistics).

[Klema and Laub, 1980] Klema, V. C. and Laub, A. J. (1980). Singular value decomposition. *European Journal Of Operational Research*, 154(3):164–176.

[Klusch et al., 2009] Klusch, M., Fries, B., and Sycara, K. (2009). Owls-mx: A hybrid semantic web service matchmaker for owl-s services. *Web Semantics Science Services and Agents on the World Wide Web*, 7(2):121–133.

[Klusch and Kapahnke, 2008] Klusch, M. and Kapahnke, P. (2008). Semantic web service selection with sawsdl-mx.

[Kollia et al., 2011] Kollia, I., Glimm, B., and Horrocks, I. (2011). Answering queries over owl ontologies with sparql. *Direct*.

[Kona et al., 2006] Kona, S., Bansal, A., Gupta, G., and Hite, T. D. (2006). Web service discovery and composition using usdl. pages 65–.

[Kopecký et al., 2007] Kopecký, J., Vitvar, T., Bournez, C., and Farrell, J. (2007). Sawsdl: Semantic annotations for wsdl and xml schema. *IEEE Internet Computing*, 11(6):60–67.

[Korenius et al., 2004] Korenius, T., Laurikkala, J., Järvelin, K., and Juhola, M. (2004). Stemming and lemmatization in the clustering of finnish text documents. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 625–633, New York, NY, USA. ACM.

[Korsgaard and Jensen, 2009] Korsgaard, T. R. and Jensen, C. D. (2009). Reengineering the wikipedia for reputation. *Electronic Notes in Theoretical Computer Science*, 244(244):81–94.

[Kritikos and Plexousakis, 2009] Kritikos, K. and Plexousakis, D. (2009). Requirements for qos-based web service description and discovery. *IEEE Transactions on Services Computing*, 2(4):320–337.

[Kungas and Dumas, 2009] Kungas, P. and Dumas, M. (2009). Cost-effective semantic annotation of xml schemas and web service interfaces. *2009 IEEE International Conference on Services Computing*, pages 372–379.

[Kuster and Konig-Ries, 2007a] Kuster, U. and Konig-Ries, B. (2007a). Semantic mediation between business partners - a sws-challenge solution using diane service descriptions. *2007 IEEEWICACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops*, pages 139–143.

[Kuster and Konig-Ries, 2007b] Kuster, U. and Konig-Ries, B. (2007b). Semantic service discovery with diane service descriptions. *2007 IEEEWICACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops*, 21(4):152–156.

[Lakatos, 1978] Lakatos, I. (1978). *The methodology of scientific research programmes*. Cambridge University Press, Cambridge.

[Lancaster, 1979] Lancaster, F. W. (1979). *Information retrieval systems : characteristics, testing, and evaluation / F. Wilfrid Lancaster*. John Wiley & Sons, New York :, 2nd ed. edition.

[Lederer and Mendelow, 1988] Lederer, A. L. and Mendelow, A. L. (1988). Convincing top management of the strategic potential of information systems. *MIS Quarterly*, 12(4):525–534.

[Lee et al., 2007] Lee, D., Kwon, J., Yang, S., and Lee, S. (2007). Improvement of the recall and the precision for semantic web services search. *6th IEEEACIS International Conference on Computer and Information Science ICIS 2007*, (Icis):763–768.

[Lee et al., 2004] Lee, J., Wu, C.-L., Lee, S.-J., Wang, Y.-C., Ma, S.-P., and Deng, W.-Y. (2004). A possibilistic petri-nets-based service discovery. 1:670 – 675 Vol.1.

[Lee, 2008] Lee, Y. (2008). Quality context taxonomy for web service quality classification. *Proceedings 3rd International Conference on Convergence and Hybrid Information Technology ICCIT 2008*, 1:230–235.

[Leknes and Munkvold, 2006] Leknes, J. and Munkvold, B. E. (2006). The role of knowledge management in erp implementation: a case study in aker kvaerner. In *ECIS*, pages 1767–1778.

[Lew et al., 2006] Lew, M. S., Sebe, N., Djeraba, C., and Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19.

[Li et al., 2004] Li, J., Stribling, J., Gil, T. M., Morris, R., and Kaashoek, M. F. (2004). Comparing the performance of distributed hash tables under churn. In *Proceedings of the Third international conference on Peer-to-Peer Systems*, IPTPS'04, pages 87–99, Berlin, Heidelberg. Springer-Verlag.

[Liang et al., 2009] Liang, Q., Li, P., Hung, P. C. K., and Wu, X. (2009). Clustering web services for automatic categorization. *2009 IEEE International Conference on Services Computing*, pages 380–387.

[Liu et al., 2009] Liu, L., Thanheiser, S., and Schmeck, H. (2009). Assessing the impact of inherent soa system properties on complexity. In *Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services*, ICIW '09, pages 429–434, Washington, DC, USA. IEEE Computer Society.

[Lo et al., 2012] Lo, W., Yin, J., Deng, S., Li, Y., and Wu, Z. (2012). Collaborative web service qos prediction with location-based regularization. In *Web Services (ICWS), 2012 IEEE 19th International Conference on*, pages 464 –471.

[Looker et al., 2004] Looker, N., Munro, M., and Xu, J. (2004). *Assessing Web Service Quality of Service with Fault Injection.*

[Lund et al., 2007] Lund, K., Eggen, A., Hadzic, D., Hafsoe, T., and Johnsen, F. T. (2007). Using web services to realize service oriented architecture in military communication networks. *Communications Magazine, IEEE*, 45(10):47 –53.

[Luo et al., 2006] Luo, J., Montrose, B., Kim, A., Khashnobish, A., and Kang, M. (2006). Adding owl-s support to the existing uddi infrastructure. *2006 IEEE International Conference on Web Services (ICWS'06)*, pages 153–162.

[Ma et al., 2008] Ma, J., Zhang, Y., and He, J. (2008). Efficiently finding web services using a clustering semantic approach. *Proceedings of the 2008 international workshop on Context enabled source and service selection integration and adaptation organized with the 17th International World Wide Web Conference WWW 2008 CSSSIA 08*, pages 1–8.

[Ma et al., 2006] Ma, L., Yang, Y., Qiu, Z., Xie, G., Pan, Y., and Liu, S. (2006). Towards a complete owl ontology benchmark. In Sure, Y. and Domingue, J., editors, *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 125–139. Springer Berlin / Heidelberg.

[Maclaren et al., 1995] Maclaren, J., Sakellariou, R., and Krishnakumar, K. T. (1995). Towards service level agreement based scheduling on the grid. *Service Management*, page 100–102.

[Mahmoud and Gomez, 2008] Mahmoud, T. and Gomez, J. M. (2008). Integration of semantic web services principles in soa to solve eai and erp scenarios; towards semantic service oriented architecture. In *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, pages 1 –6.

[Maigre, 2010] Maigre, R. (2010). Survey of the tools for automating service composition. *2010 IEEE International Conference on Web Services*, pages 628–629.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*, volume 25. Cambridge University Press.

[Mao and Le, 2009] Mao, Y. and Le, J. (2009). Research on qs-based web service description language. *Science And Technology*, pages 0–3.

[Marr et al., 2004] Marr, B., Schiuma, G., and Neely, A. (2004). Intellectual capital – defining key performance indicators for organizational knowledge assets. *Business Process Management Journal*, 10(5):551–569.

[Martin et al., 2007] Martin, D., Burstein, M., McDermott, D., McIlraith, S., Paolucci, M., Sycara, K., McGuinness, D. L., Sirin, E., and Srinivasan, N. (2007). Bringing semantics to web services with owl-s. *World Wide Web Internet And Web Information Systems*, 10(3):243–277.

[Martin et al., 2010] Martin, M., Unbehauen, J., and Auer, S. (2010). Improving the performance of semantic web applications with sparql query caching. *The Semantic Web Research and Applications*, 6089:304–318.

[Masanet et al., 2011] Masanet, E. R., Brown, R. E., Shehabi, A., Koomey, J. G., and Nordman, B. (2011). Estimating the energy use and efficiency potential of u.s. data centers. *Proceedings of the IEEE*, 99(8):1440 –1453.

[McIlraith et al., 2001] McIlraith, S. A., Son, T. C., and Zeng, H. (2001). Semantic web services. *IEEE Intelligent Systems*, 16(2):46–53.

[Mell and Grance, 2009] Mell, P. and Grance, T. (2009). The NIST Definition of Cloud Computing. Technical report.

[Merali, 2006] Merali, Y. (2006). Complexity and information systems: the emergent domain. *Journal of Information Technology*, 21(4):216–228.

[Messner and South, 2011] Messner, M. and South, J. (2011). Legitimizing wikipedia. *Journalism Practice*, 5(2):145–160.

[Middleton and Baeza-yates, 2007] Middleton, C. and Baeza-yates, R. (2007). A comparison of open source search engines. *Evaluation*, page 46.

[Miller and Fellbaum, 2007] Miller, G. A. and Fellbaum, C. (2007). Wordnet then and now. *Language Resources And Evaluation*, 41(2):209–214.

[Mokarizadeh et al., 2010] Mokarizadeh, S., Küngas, P., and Matskin, M. (2010). Ontology learning for cost-effective large-scale semantic annotation of web service interfaces. In *Proceedings of the 17th international conference on Knowledge engineering and management by the masses*, EKAW'10, pages 401–410, Berlin, Heidelberg. Springer-Verlag.

[Morsey et al., 2011] Morsey, M., Lehmann, J., Auer, S., and Ngomo, A.-c. N. (2011). Dbpedia sparql benchmark – performance assessment with real queries on real data. *System*, 2(257943):454–469.

[Muddamalle, 1998] Muddamalle, M. R. (1998). Natural language versus controlled vocabulary in information retrieval: a case study in soil mechanics. *J. Am. Soc. Inf. Sci.*, 49(10):881–887.

[Nah, 2004] Nah, F. F. (2004). A study on tolerable waiting time: how long are Web users willing to wait? *Behaviour & Information Technology*, 23(3):153–163.

[Nardi and Brachman, 2003] Nardi, D. and Brachman, R. J. (2003). An introduction to description logics. *The description logic handbook theory implementation and applications*, page 1–40.

[Nayak and Lee, 2007] Nayak, R. and Lee, B. (2007). Web service discovery with additional semantics and clustering. *Transformation*, pages 555–558.

[Needleman, 2001] Needleman, M. (2001). Rdf the resource description framework. *Serials Review*, 27(1):58–61.

[Nezval and Bartolo, 2011] Nezval, V. and Bartolo, F. (2011). A model for easy public searching of web services. In Yonazi, J. J., Sedoyeka, E., Ariwa, E., and El-Qawasmeh, E., editors, *e-Technologies and Networks for Development*, volume 171 of *Communications in Computer and Information Science*, pages 209–222. Springer Berlin Heidelberg.

[Niles and Pease,] Niles, I. and Pease, A. Towards a standard upper ontology. *Proceedings of the international conference on Formal Ontology in Information Systems FOIS 01*, 2001:2–9.

[Nonaka, 1994] Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science*, 5:14–37.

[Nonaka and von Krogh, 2009] Nonaka, I. and von Krogh, G. (2009). Perspective—tacit knowledge and knowledge conversion: Controversy and advancement in organizational knowledge creation theory. *Organization Science*, 20(3):635–652.

[Oasis, 2004] Oasis (2004). Introduction to uddi: Important features and functional concepts. *Focus*, (October).

[Ondrus and Pigneur, 2009] Ondrus, J. and Pigneur, Y. (2009). C-k design theory for information systems research. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, DESRIST '09, pages 28:1–28:2, New York, NY, USA. ACM.

[Ortiz et al., 2005] Ortiz, G., Hernández, J., and Clemente, P. (2005). How to deal with non-functional properties in web service development. *Web Engineering*, page 98–103.

[Ounis et al.,] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., and Lioma, C. *Terrier: A High Performance and Scalable Information Retrieval Platform*, volume 2006, page 18–25. Citeseer.

[Ozcan et al., 2008] Ozcan, R., Altingovde, I. S., and Ulusoy, z. (2008). Static query result caching revisited. *Performance Evaluation*, page 1169.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

[Paliwal et al., 2007] Paliwal, A. V., Adam, N. R., and Bornhovd, C. (2007). Web service discovery: Adding semantics through service request expansion and latent semantic indexing. *IEEE International Conference on Services Computing SCC 2007*, (Scc):106–113.

[Paolucci et al., 2002] Paolucci, M., Kawamura, T., Payne, T. R., and Sycara, K. P. (2002). Importing the semantic web in uddi. In *Revised Papers from the International Workshop on Web Services, E-Business, and the Semantic Web*, CAiSE '02/ WES '02, pages 225–236, London, UK, UK. Springer-Verlag.

[Papageorgiou et al., 2010] Papageorgiou, A., Krop, T., Ahlfeld, S., Schulte, S., Eckert, J., and Steinmetz, R. (2010). Enhancing availability with self-organization extensions in a soa platform. In *Proceedings of the 2010 Fifth International Conference on Internet and Web Applications and Services*, ICIW '10, pages 161–166, Washington, DC, USA. IEEE Computer Society.

[Papazoglou and Heuvel, 2007] Papazoglou, M. P. and Heuvel, W.-J. (2007). Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal*, 16(3):389–415.

[Papazoglou et al., 2007] Papazoglou, M. P., Traverso, P., Dustdar, S., and Leymann, F. (2007). Service-oriented computing: State of the art and research challenges.

[Paper, 2007] Paper, T. W. (2007). Service oriented architecture (soa) and specialized messaging patterns. *Architecture*, 345:1–15.

[Parr and Fisher, 2011] Parr, T. and Fisher, K. (2011). Ll(*): the foundation of the antlr parser generator. In Hall, M. W. and Padua, D. A., editors, *PLDI*, pages 425–436. ACM.

[Pasley, 2005] Pasley, J. (2005). How bpel and soa are changing web services development. *Internet Computing, IEEE*, 9(3):60 – 67.

[Patil et al., 2004] Patil, A. A., Oundhakar, S. A., Sheth, A. P., and Verma, K. (2004). Meteor-s web service annotation framework. *Proceedings of the 13th conference on World Wide Web WWW 04*, page 553.

[Perez et al., 2006] Perez, J., Arenas, M., and Gutierrez, C. (2006). Semantics of sparql. *Syntax*, 4(D):1–29.

[Peroni et al., 2008] Peroni, S., Motta, E., and d'Aquin, M. (2008). Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. *Third Asian Semantic Web Conference ASWC 2008*, pages 242–256.

[Pierre et al., 2009] Pierre, G., Schütt, T., Domaschka, J., and Coppola, M. (2009). Highly available and scalable grid services. In *Proceedings of the Third Workshop on Dependable Distributed Data Management*, WDDM '09, pages 18–20, New York, NY, USA. ACM.

[Plebani and Pernici, 2009] Plebani, P. and Pernici, B. (2009). Urbe: Web service retrieval based on similarity evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 21(11):1629–1642.

[Prazeres et al., 2009] Prazeres, C. V. S., Teixeira, C. A. C., and Pimentel, M. D. G. C. (2009). Semantic web services discovery and composition: Paths along workflows. *2009 Seventh IEEE European Conference on Web Services*, pages 58–65.

[Qin et al., 2010] Qin, Z., Li, P., Zhu, Q., and Tian, C. (2010). Swee: Approximately searching web service with keywords effectively and efficiently. *2010 2nd International Conference on Advanced Computer Control*, pages 569–574.

[Radetzki and Cremers, 2006] Radetzki, U. and Cremers, A. B. (2006). Automatic discovery and composition of services with iris. *22nd International Conference on Data Engineering Workshops ICDEW06*, pages 39–39.

[Ramos et al., 2003] Ramos, J., Eden, J., and Edu, R. (2003). Using tf-idf to determine word relevance in document queries. *Processing*.

[Reichwald et al., 2002] Reichwald, R., Piller, F. T., and Möslein, K. M. (2002). *Mass Customization Concepts for the E-Conomy: Four Strategies to Create Competitive Advantage With Customized Goods and Services on the Internet*. University of Wollongong.

[Ren and Xu, 2008] Ren, W. and Xu, Z. (2008). A New Web Service Discovery Method Based on Semantic. *2008 Workshop on Power Electronics and Intelligent Transportation System*, (2):223–226.

[Robertson, 2004] Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for idf. *Journal of Documentation*, 60(5):503–520.

[Rocco et al., 2005] Rocco, D., Caverlee, J., Critchlow, T., and Liu, L. (2005). Domain-specific web service discovery with service class descriptions. *IEEE International Conference on Web Services ICWS05*, pages 481–488.

[Rochkind, 1975] Rochkind, M. J. (1975). The source code control system. *IEEE Transactions on Software Engineering*, 1(4):364–370.

[Roman et al., 2005] Roman, D., Keller, U., Lausen, H., Bruijn, J. D., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., and Fensel, D. (2005). Web service modeling ontology. *Applied Ontology*, 1(1):77–106.

[Roth et al., 2008] Roth, R., Rambow, O., Habash, N., Diab, M., and Rudin, C. (2008). Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pages 117–120, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Sanderson and Croft, 2012] Sanderson, M. and Croft, W. B. (2012). The history of information retrieval research. *Proceedings of the IEEE*, 100(13):1444 –1451.

[Schewe, 2001] Schewe, K.-D. (2001). *UML: A Modern Dinosaur? A Critical Analysis of the Unified Modelling Language*, volume 67, pages 185–202. IOS Press.

[Schmidt et al., 2009] Schmidt, M., Hornung, T., Lausen, G., and Pinkel, C. (2009). sp^2bench: A sparql performance benchmark. *2009 IEEE 25th International Conference on Data Engineering*, pages 222–233.

[Services and Architecture, 2001] Services, W. and Architecture, C. (2001). Web services conceptual architecture (wsca 1.0). *Architecture*, 5(May):6–7.

[Shafiq et al., 2010] Shafiq, O., Alhajj, R., and Rokne, J. (2010). Light-weight semantics and bayesian classification: A hybrid technique for dynamic web service discovery. *Information Reuse and*, pages 121–125.

[Shin, 1999] Shin, N. (1999). Strategies for competitive advantage in electronic commerce. *Business*, 2(4):164–171.

[Sirin and Parsia, 2004] Sirin, E. and Parsia, B. (2004). Planning for semantic web services. *Mind*, 103(3):609–624.

[Sirin and Parsia, 2007] Sirin, E. and Parsia, B. (2007). Sparql-dl : Sparql query for owl-dl. *Computer*, 4:1–10.

[Smith and Welty, 2001] Smith, B. and Welty, C. A. (2001). Fois introduction: Ontology - towards a new synthesis. In *FOIS*, pages iii–ix.

[Smith, 2004] Smith, K. (2004). The evolution of vocabulary. *Journal of Theoretical Biology*, 228(1):127–142.

[Song et al., 2007] Song, H., Cheng, D., Messer, A., and Kalasapur, S. (2007). Web service discovery using general-purpose search engines. In *Web Services, 2007. ICWS 2007. IEEE International Conference on*, pages 265 –271.

[Sowa, 2000] Sowa, J. (2000). *Ontology, Metadata, and Semiotics*, volume 1867, pages 55–81. Springer.

[Sriharee, 2006] Sriharee, N. (2006). Semantic web services discovery using ontology-based rating model. *2006 IEEEWICACM International Conference on Web Intelligence WI 2006 Main Conference ProceedingsWI06*, pages 608–616.

[Srivastava et al., 2007] Srivastava, B., Ponnalagu, K., Narendra, N. C., and Kannan, K. (2007). Enhancing asset search and retrieval in a services repository using consumption contexts. *IEEE International Conference on Services Computing SCC 2007*, (Scc):316–323.

[Staab et al., 2003] Staab, S., Van Der Aalst, W., Benjamins, V. R., Sheth, A., Miller, J. A., Bussler, C., Maedche, A., Fensel, D., and Gannon, D. (2003). Web services: been there, done that? *IEEE Intelligent Systems*, 18(1):72–85.

[Stegmaier et al., 2009] Stegmaier, F., Gr, U., D, M., and Kosch, H. (2009). Evaluation of current rdf database solutions. *Information Systems Journal*.

[Steinmetz et al., 2009] Steinmetz, N., Lausen, H., and Brunner, M. (2009). Web service search on large scale. In *Proceedings of the 7th International Joint Conference on Service-Oriented Computing*, ICSOC-ServiceWave '09, pages 437–444, Berlin, Heidelberg. Springer-Verlag.

[Stephens et al., 2011] Stephens, B., Cox, A. L., Rixner, S., and Ng, T. S. E. (2011). A scalability study of enterprise network architectures. In *Proceedings of the 2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems*, ANCS '11, pages 111–121, Washington, DC, USA. IEEE Computer Society.

[Stigler, 1961] Stigler, G. J. (1961). The economics of information. *The Journal of Political Economy*, 69(3):213–225.

[Stollberg et al., 2007] Stollberg, M., Hepp, M., and Hoffmann, J. (2007). *A caching mechanism for semantic web service discovery*, page 480–493. Springer-Verlag.

[Sun et al., 2007] Sun, W., Zhang, K., Chen, S.-K., Zhang, X., and Liang, H. (2007). Software as a service : An integration perspective. *Integration The Vlsi Journal*, 4749:558–569.

[Sun et al., 2011] Sun, Y., Zhao, Y., Song, Y., Yang, Y., Fang, H., Zang, H., Li, Y., and Gao, Y. (2011). Green challenges to system software in data centers. *Frontiers of Computer Science in China*, 5:353–368. 10.1007/s11704-011-0369-3.

[Tamilarasi and Ramakrishnan, 2012] Tamilarasi, K. and Ramakrishnan, M. (2012). Design of an intelligent search engine-based uddi for web service discovery. In *Recent Trends In Information Technology (ICRTIT), 2012 International Conference on*, pages 520 –525.

[Tamm and Wünsche, 2003] Tamm, G. and Wünsche, M. (2003). *Strategies to reduce information asymmetry in web service market*, pages 1898–1912.

[Thakker et al., 2010] Thakker, D., Osman, T., Gohil, S., Lakin, P., Lane, C., and House, P. (2010). A pragmatic approach to semantic repositories benchmarking. *Knowledge Creation Diffusion Utilization*, pages 379–393.

[Toma et al., 2006] Toma, I., Burger, T., Shafiq, O., Doegl, D., Behrendt, W., and Fensel, D. (2006). Grisino: Combining semantic web services, intelligent content objects and grid computing. *2006 Second IEEE International Conference on eScience and Grid Computing eScience06*, pages 39–39.

[Tomasic and Garcia-Molina, 1993] Tomasic, A. S. and Garcia-Molina, H. (1993). Caching and database scaling in distributed shared-nothing information retrieval systems. *ACM SIGMOD Record*, 22(2):129–138.

[Tosic and Pagurek, 2005] Tosic, V. and Pagurek, B. (2005). On comprehensive contractual descriptions of web services. *2005 IEEE International Conference on eTechnology eCommerce and eService*, pages 444–449.

[Traverso and Pistore, 2004] Traverso, P. and Pistore, M. (2004). Automated composition of semantic web services into executable processes. *The Semantic Web–ISWC 2004*, 3298(1):380–394.

[Treiber and Dustdar, 2007] Treiber, M. and Dustdar, S. (2007). Active web service registries. *IEEE Internet Computing*, 11(5):66–71.

[Tzagarakis et al., 2000] Tzagarakis, M., Karousos, N., Christodoulakis, D., and Reich, S. (2000). *Naming as a Fundamental Concept of Open Hypermedia Systems*, pages 103–112. ACM Press.

[Van Harmelen and Horrocks, 2000] Van Harmelen, F. and Horrocks, I. (2000). Faqs on oil: The ontology inference layer. *IEEE Intelligence Systems*.

[Van Rijsbergen, 1979] Van Rijsbergen, C. J. (1979). *Information Retrieval*, volume 30. Butterworths.

[Vaughan, 2004] Vaughan, L. (2004). New measurements for search engine evaluation proposed and tested. *Information Processing & Management*, 40(4):677–691.

[Verma and Sheth, 2007] Verma, K. and Sheth, A. (2007). Semantically annotating a web service. *IEEE Internet Computing*, 11(2):83–85.

[Vinoski, 2002] Vinoski, S. (2002). Putting the "web" into web services. *IEEE Internet Computing*, 6(August):90–92.

[Vitvar et al., 2008] Vitvar, T., Kopecky, J., Viskova, J., and Fensel, D. (2008). Wsmo-lite annotations for web services. *Knowledge Creation Diffusion Utilization*, 5021:674–689.

[Voorhees, 1994] Voorhees, E. M. (1994). *Query expansion using lexical-semantic relations*, pages 61–69. Springer-Verlag New York, Inc.

[Wang et al., 2010] Wang, L., Liu, F., Zhang, L., Li, G., and Xie, B. (2010). Enriching descriptions for public web services using information captured from related web pages on the internet. *2010 Fifth IEEE International Symposium on Service Oriented System Engineering*, pages 141–150.

[Weithöner et al., 2006] Weithöner, T., Liebig, T., Luther, M., and Böhm, S. (2006). *What's Wrong with OWL Benchmarks?*, pages 101–114. Athens, GA, USA.

[Weitzman, 1979] Weitzman, M. L. (1979). Optimal search for the best alternative. *Econometrica*, 47(3):pp. 641–654.

[Werth et al., 2006] Werth, D., Leyking, K., Dreifus, F., Ziemann, J., and Martin, A. (2006). Managing soa through business services - a business-oriented approach to service-oriented architectures. In Georgakopoulos, D., Ritter, N., Benatallah, B., Zirpins, C., Feuerlicht, G., Schönherr, M., and Nezhad, H. R. M., editors, *ICSOC Workshops*, volume 4652 of *Lecture Notes in Computer Science*, pages 3–13. Springer.

[Wilson, 2006] Wilson, T. D. (2006). On user studies and information needs. *Journal of Documentation*, 62(6):658–670.

[Winklbauer and Seidenberg, 2001] Winklbauer, H. and Seidenberg, M. (2001). Corporate intranet. *Integration The Vlsi Journal*.

[Wu, 2009] Wu, S. (2009). A new description model of web service. *2009 International Conference on Industrial and Information Systems*, pages 77–79.

[Yao et al., 2011] Yao, L., Divoli, A., Mayzus, I., Evans, J. A., and Rzhetsky, A. (2011). Benchmarking ontologies: Bigger or better? *PLoS Comput Biol*, 7(1):e1001055.

[Ye and Zhang, 2006] Ye, L. and Zhang, B. (2006). Web service discovery based on functional semantics. *2006 IEEE AsiaPacific Conference on Services Computing APSCC06*, pages 57–57.

[Yu et al., 2008] Yu, Q., Liu, X., Bouguettaya, A., and Medjahed, B. (2008). Deploying and managing web services: issues, solutions, and directions. *The VLDB Journal*, 17:537–572. 10.1007/s00778-006-0020-3.

[Zhang et al., 2009] Zhang, J., Yu, X., Liu, P., and Wang, Z. (2009). Research on improving performance of semantic search in uddi. *2009 WRI Global Congress on Intelligent Systems*, pages 572–576.

[Zhang and Li, 2005] Zhang, P. and Li, J. (2005). Ontology assisted web services discovery. *IEEE International Workshop on ServiceOriented System Engineering SOSE05*, pages 45–50.

[Zhou et al., 2005] Zhou, C., Chia, L.-T., and Lee, B.-S. (2005). Semantics in service discovery and qos measurement. *It Professional*, 7(2):29–34.

[Zhou et al., 2008] Zhou, J., Zhang, T., Meng, H., Xiao, L., Chen, G., and Li, D. (2008). Web service discovery based on keyword clustering and ontology. *2008 IEEE International Conference on Granular Computing*, 1:844–848.

[Zhou, 2007] Zhou, L. (2007). Ontology learning: state of the art and open issues. *Information Technology and Management*, 8(3):241–252.