Visual Semantic Analysis to Support Semi Automatic Modeling of Service Descriptions

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des Grades
Doktor-Ingenieur (Dr.-Ing.)

vorgelegt von
Dipl.-Ing. Nadeem Bhatti
geboren in Faisalabad (Pakistan)

Referenten
Prof. Dr. techn. Dieter W. Fellner
Prof. Dr. Tobias Schreck

Tag der Einreichung: 29.04.2011
Tag der Disputation: 10.06.2011

Darmstadt 2011
D17

Abstract

A new trend *Web service ecosystems* for Service-Oriented Architectures (SOAs) and Web services is emerging. Services can be offered and traded like products in these ecosystems. The explicit formalization of services' non-functional parameters, e.g. price plans and legal aspects, as Service Descriptions (SD) is one of the main challenges to establish such *Web service ecosystems*. The manual modeling of Service Descriptions (SDs) is a tedious and cumbersome task.

In this thesis, we introduce the innovative approach Visual Semantic Analysis (VSA) to support semi-automatic modeling of service descriptions in *Web service ecosystems*. This approach combines the semantic analysis and interactive visualization techniques to support the analysis, modeling, and reanalysis of services in an iterative loop. For example, service providers can analyze first the price plans of the already existing services and extract semantic information from them (e.g. cheapest offers and functionalities). Then they can reuse the extracted semantics to model the price plans of their new services. Afterwards, they can reanalyze the new modeled price plans with the already existing services to check their market competitiveness in *Web service ecosystems*.

The experts from different domains, e.g. service engineers, SD modeling experts, and price plan experts, were interviewed in a study to identify the requirements for the VSA approach. These requirements cover aspects related to the analysis of already exiting services and reuse of the analysis results to model new services.

Based on the user requirements, we establish a *generic process model for the Visual Semantic Analysis*. It defines sub processes and transitions between them. Additionally, the technologies used and the data processed in these sub processes are also described. We present also the formal specification of this generic process model that serves as a basis for the *conceptual framework of the VSA*.

A *conceptual framework of the VSA* elucidates structure and behavior of the *Visual Semantic Analysis System*. It specifies also system components of the *VSA system* and interaction between them. Additionally, we present the external interface of the *VSA system* for the communication with *Web service ecosystems*.

Finally, we present the results of a user study conducted by means of the *VSA system* that is developed on the base of the *VSA conceptual framework*. The results of this user study show that the *VSA system* leads to strongly significant improvement of the time efficiency and offers better support for the analysis, modeling and reanalysis of service descriptions.

Zusammenfassung

Es entsteht ein neuer Trend *Web-Service-Ökosysteme* auf denen Web-Services (Dienste) wie Güter angeboten, gehandelt und zu Mehrwertdiensten kombiniert werden können. Um dies zu verwirklichen müssen die formalen Beschreibungen der nicht-funktionalen Parameter von Diensten, z. B. Preispläne und juristische Aspekte als Dienstbeschreibungen definiert werden. Die manuelle Modellierung von Dienstbeschreibungen ist eine mühsame und langwierige Aufgabe.

In dieser Arbeit stellen wir den innovativen Ansatz Visual Semantic Analysis (VSA) vor, um semi-automatische Modellierung der Dienstbeschreibungen in *Web-Service-Ökosystemen* zu unterstützen. Dieser Ansatz kombiniert die semantische Analyse- und interaktive Visualisierungstechniken für die Analyse, Modellierung und Reanalyse der Dienstbeschreibungen. Zum Beispiel, Dienstanbieter können die Preispläne der schon existierenden Dienste analysieren und die semantischen Informationen (z. B. günsitigste Angebote und Funktionalitäten) daraus extrahieren. Danach können sie diese extrahierten semantischen Informationen wiederverwenden und damit die Preispläne eigener Dienste modellieren. Abschließend können sie neue modellierte Preispläne mit den existierenden Diensten zusammen analysieren, um ihre Konkurenzfähigkeit in *Web-Service-Ökosystemen* zu überprüfen.

Die Experten aus verschiedenen Domainen, z. B. Service-Engineers, Dienst-Modellierungs-Experten sowie Preisplan-Experten wurden in einer Studie interviewt, um die Anforderungen für die VSA herauszufinden. Diese Anforderungen umfassen Aspekte der Analyse existierender Dienste und Wiederverwendung der Analyseergebnisse für die Modellierung der neuen Dienstbeschreibungen.

Wir stellen ein *generisches Prozessmodell für die Visual Semantic Analysis* auf der Basis von Anforderungen, die aus der Studie gewonnen wurden, vor. Dabei werden Teilprozesse und ihre Abläufe festgelegt. Zusäzliche werden die verwendeten Technologien und verarbeiteten Daten in diesen Teilprozessen zusammengefasst. Wir präsentieren auch formale Spezifikationen dieses generischen Prozessmodells, das als Grundlage für den konzeptionellen Framework der VSA dient.

Weiterhin wird ein *konzeptionelles Framework für die VSA* vorgeschlagen, dieses stellt den strukturellen Aufbau und das Verhalten von *VSA-System* dar. Zusätzlich werden die Systemkomponenten des *VSA-Systems* und deren Interaktionen spezifiziert. Darüberhinaus präsentieren wir die externe Schnittstelle des *VSA-Sytems* für die Kommunikation mit den *Web-Service-Ökosystemen.*

Abschliessend präsentieren wir die Ergebnisse einer Benutzerstudie, die mittels *VSA-System* durchgeführt wurde. Das *VSA-System* wurde auf der Basis

von dem oben beschriebenen *konzeptionellen Framework der VSA* entwickelt. Die Ergebisse der Benutzerstudie zeigen, dass das *VSA-System* zur signifikanten Verbesserung der Effizienz führt und bessere Unterstützung für die Analyse, Modellierung und Reanalyse von Diensten bietet.

Contents

Chapter 1

Introduction

Many people are familiar with web-based marketplaces like eBay and Amazon, on which they can buy or sell goods. However, web-based marketplaces for business software are not very well known to the general public. Benioff [Eba10] has described web-based marketplaces for business software as *eBay for business software*. Software makers and customers can swap and sell their applications there. The first marketplaces for business software were Salesforce.com, StrikeIron.com, GrandCentral.com. These companies exploit XML and Web service technologies to offer business software as services. Their early successes have paved the way to Web services marketplaces (like Amazon and eBay).

Distinct type of marketplace and their characteristics are summarized in Table 1.1. The *Software-as-a-Service marketplaces* use WSDL, structured and unstructured text to describe technical interfaces, pricing and provider information, and benefits of services. Pay per use and hosting are their business models. *One-stop citizen and constituency services* are offered by the public sector and their emphasis is less on generating revenue. These are typically portals with links to information and services of governmental agencies. Best deals for consumers, commission fees, referral fee or sales of related hardware are the business models of *Business Service Marketplaces*. The Service Descriptions (SDs) in these marketplaces are provided as free-text (description of services) and structured text (pricing, provider details, reputation and categorization) [CBMK10].

SAP from 2000 to 2006. This can be traced to the fact that the service metadata specified in essentially technical standards is insufficient to facilitate independent discovery of services by consumers: consumers need prior, “offline” knowledge about services that are diversely supplied and not specific to a particular domain. In other words, service semantics was inadequate. Moreover, critical aspects related to service delivery (non-functional ones) were absent. Yet, consumers are reluctant to engage in business transactions without knowing about timeliness, reliability, privacy and settlement.

Service marketplaces have emerged as a later development following the successes of Internet marketplaces (like Amazon and eBay). Table 1 summarizes distinct types of marketplaces and their characteristics.

services of governmental agencies. Service descriptions are general and are geared to an audience that has high confidence in accessing exposed services and wishes to avoid inefficient government silos. Being public sector, the emphasis is less on generating revenue, although these platforms do facilitate certain service delivery functions like single-point payment and service tracking. For agency access, gateways underpinning the platforms are useful for technical integration, e.g. usa.gov exposes services like taxation details and street validation which are used by Software-as-a-Service initiatives like Salesforce.

A recent development are *Business Service Marketplaces*, e.g. American Express Intelligent Online Marketplace (AXIOM), Intel Business Exchange or IBM SmartMarket. Business service marketplaces are centrally governed by a dominant commercial player

Marketplace	URL	Service Description	Business Model
Software-as-a-Service marketplaces	www.salesforce.com, www.workday.com, www.webservicex.net	WSDL and structured description and free text	Pay per use, commission fees and rewards
One-stop citizen and constituency services	DirectGov.uk, usa.gov	Link directory and government agency documents.	Cost savings and political incentive for improved efficiency, transparency and community building for government services
Business service marketplaces	AXIOM, Intel Business Exchange, IBM SmartMarket	Structured text and free-text	Best deals, commission fees for referrals Commission fees, referral fee, or via related hardware sales.

Table 1. Classification of marketplaces and their characteristics

Software-as-a-Service marketplaces like Salesforce.com and Workday are similar to public software registries, with key differences: they are governed by a commercial player, they pertain to a specific domain (e.g. CRM in the case of Salesforce and HR in the case of Workday) and they strive for a business model that features pay-per-use pricing and hosting. While they use WSDL to provide technical interfaces, most of the remaining description is presented as structured and unstructured text. Structured text includes a categorization of the service, pricing or provider information. Unstructured text is used to point out the

focused on differentiation while deriving adjacent or outsourced services from wide and global partnerships. The dominant player benefits from increased revenue by exposing wide choice and best deals for consumers. Partners also enjoy a greater market visibility for their services. As the range of services on these marketplaces is quite diverse, descriptions are mostly free-text in addition to basic attributes like price, provider details, reputation and categorization.

From our discussion we can summarize that service descriptions in various domains are based on textual descriptions that are presented to consumers to be

Figure 1.1: Classification of marketplaces and their characteristics [CBMK10]

1.1 Motivation and Challenges

As Service-Oriented Architectures (SOAs) and Web services mature and move to main stream, a new trend *Web service ecosystems* is emerging. Service providers can offer and trade their services in these ecosystems. Service brokers offer services of different service providers to service consumers. They can also combine services form different service providers to create new value added services and offer them to services consumers. The combination of services from different service providers may need mapping between different input and output formats of services. *Service mediators* offer mapping between different input and output formats of services. Service providers, service brokers, and service mediators have to work together in order to offer services to service consumers in *Web service ecosystems*[BD06, BDB05]. Today, The Internet of Services (IoS) is considered as an infrastructure for the Web service ecosystem, where services are, published, discovered and consumed via different business channels [CBMK10, OBB+09, CWV09].

The *flexible service discovery* to find and combine services is one of the main challenges for Web service ecosystems (e.g. IoS) [BD06, CBMK10, CWV09, OBB+09]. To offer *flexible service discovery*, service providers must describe non-functional parameters of services, e.g. price plans, legal aspects, Service Level Agreements (SLAs), penalties, and delivery modes. The service descriptions can be very complex, e.g. the price plans of SDs may contain information like pay per use, monthly fee, flat rate, discounts, and special offers. The dependencies between non-functional parameters, e.g. price plans, functionality of service, and Quality of Service (QoS) parameters, make the service description even more complex. Therefore, the explicit and formalized description of services' non-functional parameters is the bottleneck for Web service ecosystems [CBMK10, OBB+09, CWV09]. The manual modeling of Service Descriptions (SDs) is a tedious and cumbersome task. The automation of SD modeling task is desirable as it is mentioned in the different research work [WWWC08, SP07, Sab06, SWGS05, WGG+04].

Although the product descriptions of the tradable products on marketplaces like eBay and Amazon are not so complicated, modeling tools such as ChannelAdvisor are available to support the semi-automatic modeling of product descriptions. In the same way, the modeling tools are required to support semi-automatic modeling of services in ecosystems [WWWC08, SP07, Sab06, SWGS05]. These tools should support service providers to deal with the complexity of SDs and accelerate the SD modeling process.

The modeling of service description required not only the cost intensive manual effort, but also market analysis techniques to offer competitive services in Web service ecosystems. A number of market research tools (e.g. terapeak) for the marketplaces eBay and Amazon are available today. Sellers use these tools to analyze the market and develop new business ideas. In the same way, the market analysis techniques are required for the Web service ecosystems. These techniques should support analysis of non-functional parameters of SDs, e.g. price plans, SLAs and legal aspects. The results of market analysis should facilitate modeling of market competitive services in Web service ecosystems.

Visual techniques take into account human perceptual capabilities to detect patterns and outliers in visual information. The application of visualization techniques facilitate understanding about extracted complex data, structures, hierarchy and relations in market analysis. Sabou [SP07, Sab05] has purposed application of visual techniques for the semi-automatic modeling of service descriptions.

The semi-automatic modeling of SDs is already introduced in different research work [WWWC08, SP07, HJK04, POSV04]. The semantic analysis techniques are applied in these approaches to extract semantics from the functional parameters and textual descriptions of services in order to annotate Web Services. The non-functional aspects, like price plans and SLAs, are not handled in these approaches. The simultaneous application of semantic analysis and interactive visualization techniques for the analysis and modeling is also not the focus of these approaches.

The semantic analysis techniques [PF09a, JNT09, BTN08, CHS05, Mae02] offer machine learning methods to analyze service description, but they don't address the context of service description modeling. The application of interactive visualization techniques for semantic analysis is also not discussed in these techniques. The Visual Analytics (VA) [KAF+08, MKJ08] offers a combination of data analysis and visualization techniques to analyze Semantic Descriptions, but they don't deal also with the modeling of service descriptions.

1.2 Aim of this thesis and Contribution

This thesis investigates new methods and techniques to support the semi-modeling of Service Descriptions in Web service ecosystems. Main focus is given to the good understanding of semantic analysis and service modeling processes. In particular, the combination of semantic analysis and visualization techniques are investigated to support semi-automatic modeling of Service Descriptions.

A significant effort is devoted for the development of new techniques, methods and tools to support service providers and service brokers in the SD modeling process. Attention has been devoted to the establishment of an innovative approach Visual Semantic Analysis (VSA) to support semi-automatic modeling of services in Web service ecosystems. It combines semantic analysis and interactive visualization techniques to support analysis, modeling, and reanalysis in an iterative loop. Additionally, it offers intuitive User Interfaces to deal with the complexity of Service Descriptions. The theses of this thesis are as follows:

The main theses of the work are that the VSA approach:

1. Improves time efficiency regarding time duration required to analyze, model and reanalyze services.
2. facilitates service analysis and semantic extraction.
3. supports semi-automatic modeling of new service description by reusing and adapting the extracted semantics.
4. enables reanalysis of new modeled services to check their market competitiveness.
5. facilitates users to deal with the complexity of SD modeling.

These theses lead to the following research questions:

1. What are the user requirements for the VSA?
2. How to support users to deal with the complexity of SD modeling?
3. What is the process model for the VSA?
4. What is the conceptual framework for the VSA?

This thesis introduces the Visual Semantic Analysis (VSA) approach that combines semantic analysis and visualization techniques in order to support semi-automatic modeling of SDs. My contributions are:

A study was conducted to identify the requirements for the VSA.

The knowledge experts from different domains such as service engineers, SD modeling experts, price plan experts, SLA experts, and legal experts were interviewed in this study. The requirements were considered from the perspectives of users, semantic analysis and SD modeling. The requirement from the perspective of semantic analysis covers the aspects related to analysis of the available service and usage of visualization techniques for this analysis. The modeling of new services and usage of visualization for the service modeling are addressed by the requirement from the SD modeling perspective. The focus of the users perspective is the interplay between semantic analysis and SD modeling and usage of visualization techniques in order to support the analysis of available services, modeling of new services and reanalysis of new modeled services with the available services in an iterative loop. These requirements serve as basis for the definition of a generic process model for the VSA.

A generic process model for the VSA is introduced.

This generic process model describes (i) sub processes and tasks, (ii) transitions between sub processes, (iii) technologies used in sub processes, and (iv) data processed and generated in the sub processes. The iterative loop illustrates the analysis, modeling, and reanalysis of services in an iterative manner. The manual validation and refinement of automatically extracted semantics and semi-automatic modeling of new services by reusing and adapting extracted semantics require synchronization of these manual changes in the VSA process. The merging of manually generated semantic information and automatically extracted semantics are also described in this generic process model of the Visual Semantic Analysis. Furthermore, the perspectives in the VSA approach allow users to deal with the complexity of SDs by considering different aspects of SDs, e.g. price plans and Service Level Agreements (SLAs), individually or simultaneously. This generic process model serves as prerequisite for the formal specification of the VSA process and conceptual framework of the VSA.

A formal specification of the VSA process is presented

For the specification of the VSA process model, a definition of a process is presented first, and then a specification of the VSA process on the base of the definition of the process is illustrated. The elements (i) sub processes, (ii) transitions between sub processes (iii) conditions for the transitions between sub processes (iv) techniques applied in sub processes (v) data processed and generated in sub processes, and (vi) inputs and out puts of sub process are presented in the VSA specification. Furthermore, the context and semantics processed and generated in this sub process are also part of this formal specification. The formal specification serves as guidelines for the conceptual framework of the VSA.

A Conceptual Framework for the VSA approach is described.

A conceptual framework of the VSA describes system components of the Visual Semantic Analysis System. These components manage sub processes and their execution order according to the generic process model of the VSA. The consideration of sub set of services and individual aspect of SDs in different perspectives is particularly considered within this conceptual framework. Additionally, different communication components are introduced that define external interfaces and communication with Web service ecosystems. The import of the available services and publishing of new modeled services on Web service ecosystems is also discussed. Furthermore, the management of context and semantics processed and generated in the sub processes or different sessions of the VSA process is also part of the conceptual framework.

A user study was conducted to verify the theses of this thesis.

A prototype of the Visual Semantic Analysis System Sophie is developed on the VSA conceptual framework. Different application benefits referring to theses of this thesis efficiency, service analysis, service modeling, reanalysis of modeled services, complexity of SDs and user satisfaction as described above are expected from this prototype. A user study was conducted to verify these aspects. The VSA prototype Sophie was compared with a service modeling tool USDL-Editor. The evaluation of the results of this user study demonstrates that the time efficiency and the support for the analysis, modeling and reanalysis of services are improved significantly. Furthermore, it shows that the perspectives for the VSA facilitate users to deal with the complexity of SDs. These improvements lead to increase the general user satisfaction.

1.3 Application Benefits

The evaluation of the results of the conducted user study verifies the theses of this work. The application benefits of the VSA approach are as follows:

The VSA approach improves time efficiency regarding time duration required to analyze, model and reanalyze services.

The VSA prototype allows users to extract semantic information from available service and model new services by reusing and adapting extracted semantics instead of modeling new services from scratch. The reanalysis of new-modeled services together with existing services facilitates the comparison of newly modeled services with existing ones. In this way, it improves the time efficiency to model new services.

We show that the VSA approach facilitates service analysis and semantic extraction.

The VSA approach offers analysis of available services by using semantic analysis techniques. The usage of either one semantic analysis technique or a combination of them facilitates the extraction of semantics from available services. The visualization of semantic analysis results allows certain patterns like clustering of similar services regarding to their prices plans, QoS parameters, or functionalities to be detected. It means that the analysis of services and visualization of analysis results improve the support for the service analysis.

The reuse and adaption of extracted semantics supports semi-automatic modeling of new service descriptiosn significantly with the VSA approach.

The interactive visualization of extracted semantics provides a better overview of extracted semantics, where the services with specific patterns, e.g. price plans, SLAs, and functionality, can easily be identified. The interactive visualization doesn't only allow similar or competitor services to be identified, but also supports the reuse or adaption of extracted semantics to model new services. In this way, the semantic service descriptions of similar or competitor services are available during the service modeling process that helps users to understand and compare the complex SDs of similar services. Additionally, It supports users to reuse or adapt extracted semantic to model new services under the consideration of the competitiveness of their new modeled services, therefore, this approach improves support for semi-automatic modeling of services.

The VSA approach enables reanalysis of new modeled services to check their market competitiveness.

The reanalysis of new modeled services with available services and visualization of reanalysis results allow users to compare new modeled services with available services easily. In this way, they enter into a loop, where semantic analysis and interactive visualization techniques allows them to analyze, model and reanalyze services in an interactive manner until they are satisfied with the competitiveness of their new modeled services. We show that support for the reanalysis of services is improved by applying this approach.

The perspectives in the VSA approach facilitate users to deal with the complexity of SD modeling.

The VSA offers multiple perspectives in order to deal with the complexity of SDs. It allows users to select a sub set of individual aspects of SDs to define different perspectives, before they apply visual semantic analysis approach in these perspectives. The results of the user study prove that perspectives in the VSA facilitate users to deal with complexity of service descriptions.

1.4 Outline of This thesis

The theoretical fundamentals related to services and Service Description (SD) is introduced in Chapter 2. A general service life cycle gives an overview on different phases of the service development and the different roles involved in it. The examples of statistics based, rule-based, hybrid, and formal semantic extraction techniques provide an overview of different semantic analysis techniques. The examples of these semantic analysis techniques demonstrate how semantics can be extracted from structured and unstructured data. The use of visualization and Visual Analytics (VA) techniques for a semantic analysis are also discussed.

Different manual Service Description (SD) modeling, semi-automatic SD modeling and semantic analysis approaches are discussed in Chapter 3 that serve as related work to this thesis and play a key role to identify the requirements for the VSA approach. First, an overview of different manual SD modeling approaches is presented. Then, semantic extraction techniques to support semi-automatic SD modeling are illustrated. Afterward, semantic analysis approaches introduce semantic extraction from structured and unstructured data. Finally, the Visual Analytics based semantic analysis techniques show how semantic analysis and visualization techniques can support the semantic analysis process.

A generic process model of the Visual Semantic Analysis (VSA) is introduced in Chapter 4. First, the terms used in this generic process model are defined and the requirements analysis for the VSA is discussed in detail. Then, the comparison of VSA approach with the existing related research efforts gives a brief overview of its features. The next section presents a generic process model of the VSA that describes the sub processes and their execution order.

Finally, a formal specification of the basic elements of this generic process model of the VSA is presented.

A conceptual framework for the visual semantic analysis is presented in Chapter 5. It describes the system behavior and system components of the VSA system. It manages tasks and sub processes, execution order of sub processes, application of semantic and visualization techniques in the sub processes and data generated and processed in the sub processes. The perspectives based semantic analysis is also introduced to deal with a large number of services and complexity of SDs, Furthermore, the interaction between VSA system and service platforms by using different platform services is described in detail that allows the VSA approach to be used for different service platforms.

A prototype of the VSA system based on the VSA conceptual framework is presented in Chapter 6. This prototype supports the aspects like analysis of available services, modeling of new services and reanalysis of new modeled services with the available services. Four application scenarios for the VSA approach "service matchmaking and annotation", "price plan analysis", "legal aspects analysis" and "Service Level Agreements (SLAs) analysis" is also presented briefly that serves as proof of concept for the VSA approach.

The evaluation of the Visual Semantic Analysis approach is described in Chapter 7. A user study was conducted to verify the application benefits referring to aspects efficiency, service analysis, service modeling, reanalysis of modeled services, scalability and user satisfaction. The prototype of the VSA system Sophie was compared with a service modeling tool USDL-Editor for the evaluation. After the introduction of the applied evaluation method and study design, the evaluation results are presented. A conclusion wraps up the discussions done within this chapter.

Finally, the outcome of this thesis is summarized and discussed with respect to the achieved contribution in Chapter 8. This chapter concludes with an out view on future work for further research in the field of Visual Semantic Analysis.

1.5 Publications

A large part of the work presented in this thesis has been peer-reviewed and published within the following conference proceedings.

1. Bhatti, Nadeem; Fellner, Dieter W. (2011): **Visual Semantic Analysis to Support Semi-Automatic Modeling of Service Descriptions**. In *Modern Software Engineering Concepts and Practices: Advanced Approaches.* Hershey, Pennsylvania : IGI Global, 2011

2. Oberle, Daniel; Bhatti, Nadeem; Brockmans, Saartje; Niemann, Michael; Janiesch, Christian (2009):**Countering Service Information Challenges in the Internet of Services.** In *Business & Information Systems Engineering.* 1 (2009), 5, pp. 370-390.

3. Oberle, Daniel; Bhatti, Nadeem; Brockmans, Saartje; Niemann, Michael; Janiesch, Christian (2009):**Effektive Handhabung von Service-Informationen im Internet der Dienste.** In *Wirtschaftsinformatik.* 51 (2009), 5, pp. 429-452.

4. Bhatti, Nadeem; Weber, Stefan Hagen (2009): **Semantic Visualization to Support Knowledge Discovery in Multi-Relational Service Communities.** In *Cruz-Cunha, Maria Manuela (Ed.); Oliveira, Eva Ferreira de (Ed.); Tavares, António José Videira (Ed.); Ferreira, Luis G. (Ed.): Handbook of Research on Social Dimensions of Semantic Technologies and Web Services.* Hershey, Pennsylvania : IGI Global, 2009, 23 p.

5. Burkhardt, Dirk; Nazemi, Kawa; Bhatti, Nadeem; Hornung, Christoph (2009): **Technology Support for Analyzing User Interactions to Create User-Centered Interactions.** In *HCI International 2009. Proceedings and Posters [DVD-ROM] : With 10 further Associated Conferences.* Berlin, Heidelberg, New York : Springer, 2009, LNCS 5614, pp. 3-12. (Lecture Notes in Computer Science (LNCS)).

6. Godehardt, Eicke; Bhatti, Nadeem (2008): **Using Topic Maps for Visually Exploring Various Data Sources in a Web-based Environment.** *In Maicher, Lutz (Ed.); Garshol, Lars Marius (Ed.): Scaling Topic Maps : Third International Conference on Topic Map Research and Applications, TMRA 2007.* Berlin; Heidelberg; New York : Springer, 2008, 6 p. (Lecture Notes in Computer Science (LNCS)).

7. Bhatti, Nadeem (2008): **Web Based Semantic Visualization to Explore Knowledge Spaces - An Approach for Learning by Exploring.** In *Luca, Joseph (Ed.); Weippl, Edgar R. (Ed.); Association for the Advancement of Computing in Education (AACE): Proceedings of ED-Media 2008 : World Conference on Educational Multimedia,* Hypermedia & Telecommunications [CD-ROM]. Chesapeake, 2008, pp. 312-317.

8. Bhatti, Nadeem; Godehardt, Eicke; Hornung, Christoph (2005): **Modular Virtual Learning World.** In *Kommers, Piet (Ed.); Richards, Griff (Ed.); Association for the Advancement of Computing in Education (AACE): Proceedings of ED-Media 2005 : World Conference on Educational Multimedia, Hypermedia & Telecommunications [CD-ROM].* Norfolk,

2005, pp. 146-153.

Chapter 2

Services and Semantic Analysis

The Service-Oriented Architecture (SOA) has become one of the most popular approaches for distributed business applications. A new trend service ecosystem is emerging, where service providers can augment their core services by using business service delivery related available functionalities, like distribution and delivery. The service providers may re-combine and mediate services of other providers in such way that have not been anticipated by their original providers [BD06]. The formalization of functional and non-functional parameters of services as service description will be needed to accomplish the vision of the service ecosystem as discussed in the chpater 1.

The manual modeling of service description is still tedious and cumbersome task. The semantic analysis aims at facilitating the semantic extraction by combining interaction between human modeler and machine learning methods. The knowledge experts should be able to analyze unstructured, semi-structured and structured data to extract semantic information by using semantic analysis techniques, which should lead to semi-automatic modeling of service descriptions. [Mae02].

This chapter introduces the definitions of the terms that are later used in this thesis. The terms related to services, service descriptions, and service lifecycles are first presented in this chapter. They are followed by a description of the semantic analysis methods and interactive visual techniques to support semantic analysis.

2.1 Services

The terms “service” and “Web service” are often used synonymously in computer science. The difference between these terms is often discussed and still there are different definitions of both terms in the research community. The definitions of "service" and "Web services" according to Fensel et al. [FLP+07, p. 39] are:

> *Service: A service is defined as the provision of a concrete product or abstract value in some domain.*
>
> *Web service: Web services are defined as computational entities accessible over the Internet (using Web service standards and protocols) via platform- and programming-language-independent interfaces.*

The following example describes both terms "service" and "Web service": Let us consider a person who wants to book a flight from Frankfurt to Madrid. The provision of service and contractual issues related to it are independent of how the supplier and the provider interact. It is not relevant if the person goes to the airline ticket office or uses the airline's website to book the ticket. There, provision of value is considered as *service*. An airline can offer a software component accessible via Web service standards to request a *Web service* to request the booking of a flight. Thus, Web service means to consume an actual service or place a contract for actual service via internet. The Service Description (SD) of both services and Web services will be considered in this work from different perspectives, e.g. technical and business perspectives. For the simplicity, the term "service" is used for the both terms "service" and "Web service" in this thesis.

2.1.1 Service Oriented Architecture (SOA)

Services offer access to functionalities via internet by using standardized interfaces and open standards. The use of open standards makes distributed applications independent of the operating system platform and programming languages. SOAs facilitate the development of service based distributed applications. Figure 2.1 shows the basic building blocks of a service-oriented architecture. Service providers have to publish a service, before a service can be discovered by a service consumer. Then service consumers can issue queries to find a desired service. The discovery process is regulated by different standards, e.g. UDDI (Universal Description, Discovery, and Integration). The description of a service interface is needed to invocate a service. The description of the interface is provided by a service description language e.g. WSDL (Web Service Description Language.) The message and data exchange between services is achieved by the SOAP (Simple Object Access Protocol) standard. These core standards (like UDDI, WSDL and SOAP) facilitate basic functionalities of service-oriented architecture.

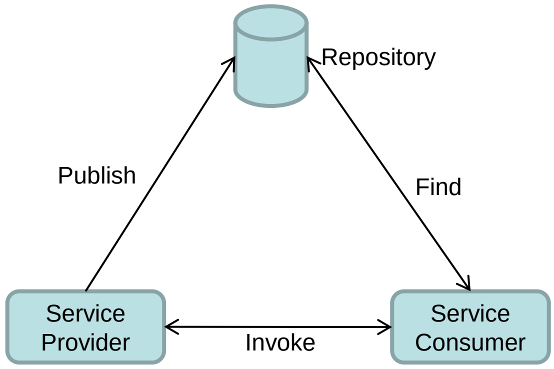


Figure 2.1: Basic building blocks of Service Oriented Architecture (SOA)

Only syntactical aspects are addressed so far in this section. These technologies offer a set of rigid services that can not be adapted to changing environment without keeping human programmers in the loop. Nevertheless, the service technologies and SOA represent a milestone in making cross enterprise or organization distributed applications possible on large scale. They provide first time a widely spread standard to manage communication between distributed applications and allow combination of them to develop more complex entities. It is the major advantage of services as compared to previous or existing middleware standards, which did not reach such level of consensus and acceptance. The lack of agreement between global industrial players and the lack of simple widespread application- and programming-language-independent protocols and standard data exchange formats were main reasons for the failure of previous and existing middleware standards [FLP+07].

The definition of service-oriented architectures is still heavily discussed. Fensel et al. [FLP+07] outline basic principles as listed abelow:

1. Loose coupling: Every service should be atomic, self-describing, accessible, declarative, stateless and composite in SOA.

2. Contracted: The inputs, outputs, access policies, quality-of-service requirements, and error-handling procedures are described as contract, which represent services in SOA.

3. Discoverable: At the time of execution, services should be able to be discoverable.

4. Addressable: Services should have unique identity in a network for their identification.

5. Distributed: Service should be good netizen application because they

are separated by geographical and machine boundaries i.e. they must be able to recover from loss of communication.

6. Point-to-Point: A service consumer uses one and only one procedure at any point in time

2.1.2 Service Standards

In recent years services have become a widely accepted common standard for the development of distributed applications. They introduced a new abstraction layer and a radically new architecture for software. They allow to reuse and combine software components via standardized interfaces. They facilitate the development of distributed business applications and integration of existing software within or across enterprise boundaries. The two application areas for services are Enterprise Application Integration (EAI) and E-Commerce. The demand for flexible EAI solutions instead of expensive reimplementation is increasing rapidly. The online stores and electronic marketplaces can profit by integrating their current service into more value added services by using SOAs [FLP+07, SP07, Sab06].

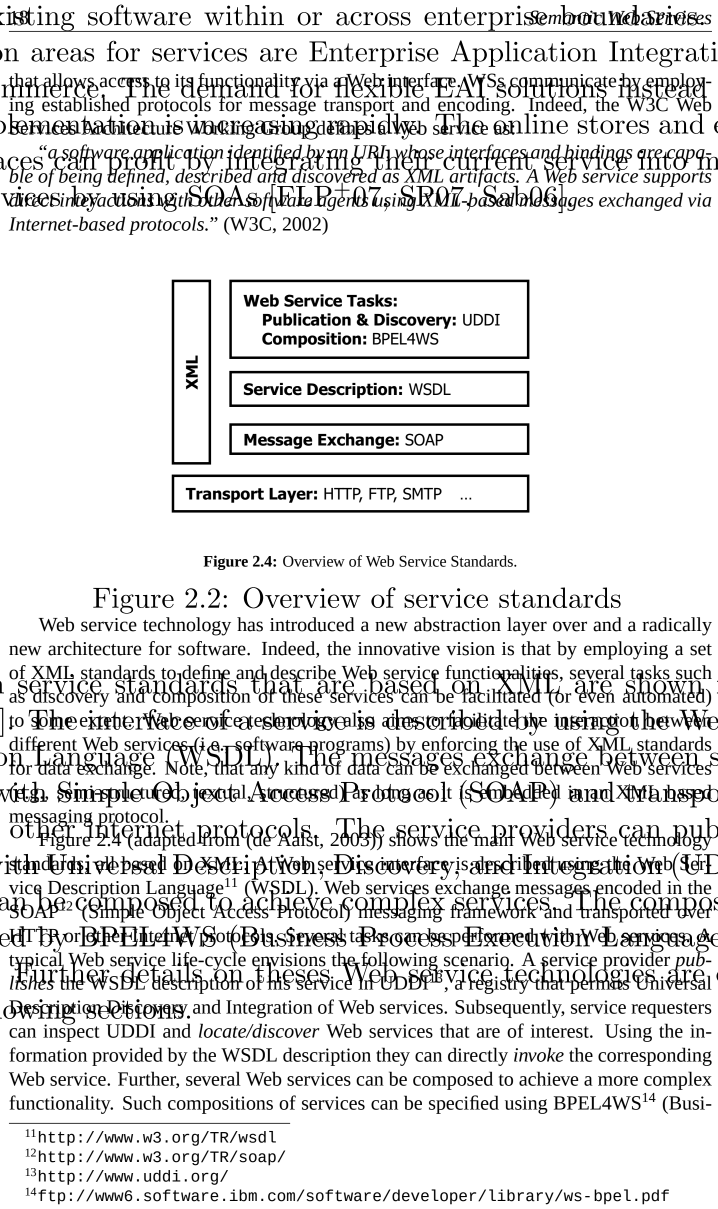


Figure 2.2: Overview of service standards

The main service standards that are based on XML are shown in Figure 2.2 [SP07]. The interface of a service is described by using the Web Service Description Language (WSDL). The messages exchange between services is encoded with Simple Object Access Protocol (SOAP) and transported over HTTP or other internet protocols. The service providers can publish their services with Universal Description, Discovery, and Integration (UDDI). The services can be composed to achieve complex services. The composition can be specified by BPEL4WS (Business Process Execution Language for Web Services). Further details on theses Web service technologies are described in the following sections.

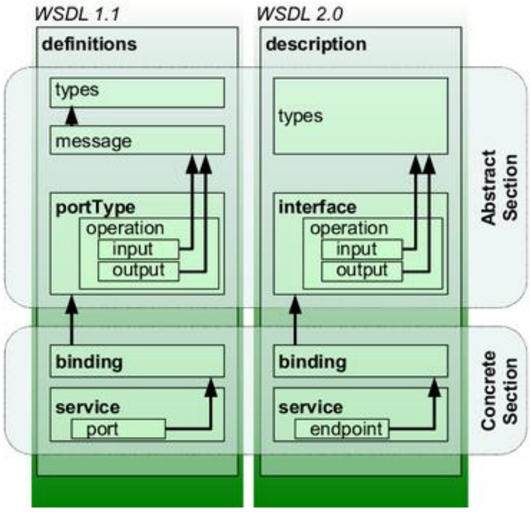


Figure 2.3: The elements of WSDL 1.1 and WSDL 2.0

SOAP

The Simple Object Access Protocol (SOAP) is a message exchange format that can be bound to different underlying transport protocols. SOAP aims to achieve maximum acceptance and flexibility by provisioning a framework in which application specific information can be exchanged in extensible manner, without any commitment to the semantics of application-specific data. It provides all required actions, which are needed to process the SOAP message. For example, the SOAP standard specifies [illegible] can be used [illegible] issue. The additional routing or fault-handling information is also available in the SOAP message. SOAP 1.2 is a W3C Recommendation since 27 April 2007 and has been adapted in most Web service based applications [Sab06].

WSDL

The Web Service Description Language (WSDL) describes interfaces, e.g. operations and their input and output message of Web services. WSDL 1.1 is a W3C standard since March 2001 and WSDL 2.0 is a recommendation of W3C since June 2007. Figure 2.3 shows various elements of WSDL 1.1 and WSDL 2.0.

The first three boxes in Figure 2.3 represent generic elements of WSDL 1.1/ WSDL 2.0. The purpose of the types/types in WSDL 1.1/WSDL 2.0 is to describe the data by using XML Schema. The message/N.A. includes defi-

The first three box in the figure ??? represent generic elements of WSDL 1.1/ WSDL 2.0. The purpose of the types/types in WSDL 1.1/WSDL 2.0 is to describe the data by using XML Schema. The message/N.A. includes definition of input and output that refers to types defined in generic part element of WSDL 1.1. The message element was removed in WSDL 2.0, where the definition bodies of inputs, outputs and faults can directly be refered to XML schema types. The portType element of WSDL 1.1 or interface element of WSDL 2.0 defined operation and messages that are needed to perform an operation of a service. The operations are comparable to methods or functions in programming languages.

The bottom two boxes in the figure ??? show binding of the interface of a service. The port/endpoint defines address or connection to specific endpoint of a service. The element service/service is a container for system functions. The endpoint defines the physical address where message can be sent. The binding specifies the interface by defining SOAP binding style (RPC/Document) and transport (SOAP protocol).

An example of WSDL file is shown in the listing ???. The definition of inputs and outputs and their linkage to operation are presented in the WSDL file. The type section contains element definition of getWeather and it child elements CityName and CountryName with type “String”. A PortType definitionwith GetWeather can be seen at the bottom of WSDL file. The input and outputs are linked to previously defined message in type element. Listing ??? shows binding element of WSDL. The GlobalWeatherSoap binding defines a binding style using SOAP and HTTP protocol. The actual endpoint (soap:address) is declared in service element

UDDI

The Universal Description, Discovery and Integration (UDDI) allow business partners to publish brows and query existing services. UDDI provide a data model for Web services and business entities. The data model contains application information like, categories, contacts, URLs or other necessary information. The description of group of services, which are used to compose a Web service, is described as service information.

The technical details like URLs, method names, arguments type and so on are available as binding information in the data model. The binding information contains all information, which is necessary to invoke a service. The metadata as Web service specific details specify various implementation of Web services. These information are called tModels in UDDI specification.

nition of input and output that refers to types defined in generic elements of WSDL 1.1. The message element has been removed in WSDL 2.0, where the definition bodies of inputs, outputs, and faults can directly be referred to XML schema types. The portType element of WSDL 1.1 or interface element of WSDL 2.0 defines operations and messages that are needed to perform an operation of a service. The operations are comparable to methods or functions in programming languages. The two boxes at the bottom in Figure 2.3 show binding of the interface of a service. The port/endpoint defines address or connection to a specific endpoint of a service. The element service/service is a container for system functions. The endpoint defines the physical address where messages can be sent. The binding specifies the interface by defining SOAP binding style (RPC/Document) and transport (SOAP protocol).

UDDI

The Universal Description, Discovery and Integration (UDDI) allows business partners to publish, browse and query existing services. UDDI provides a data model for Web services and business entities. The data model contains application information like categories, contacts, URLs or other necessary information. The description of group of services, which are used to compose a Web service, is described as service information. Technical details, like URLs, method names, argument types and so on, are available as binding information in the data model. The binding information contains all information which is necessary to invoke a service. The metadata as Web service specific details specify various implementation of Web services[Ber07]. UDDI provides three categories of information: white, yellow and green pages as described below:

- *White pages*: The name, description and all contact information of the provider, who has published services.
- *Yellow pages*: Categorization of service providers and their offered services.
- *Green pages*: The description of technical documentation of published service. For example, WSDL as an interface of a Web service.

BPEL4WS

The composition of complex services by using a set of services can be specified by Business Process Execution Language for Web Services (BPEL4WS). The BPEL4WS is an XML based standard to describe formal notation of workflows. The composition of different services to model a workflow can be specified by BPEL4WS. Such workflows can be executed by workflow engines, which support BPEL4WS. The workflow engines offer such specified workflows as a service. In this way, the specified workflows can also be used further in different workflows [Ber07, Sab06].

Workflows can be categorized as executable or abstract processes. The executable process models the behavior of business transactions of involved partners. The abstract processes specify just message exchange and message format without specifying internal behavior of business transactions. The abstract processes allows business partners to specify cross enterprise workflows without publishing their internal workflows [Ber07].

Limitations of Service Standards

The service standards SOAP, WSDL, UDDI, and BPEL4WS as discussed above to realize services and SOA based applications. However, they fail to achieve automation and interoperability, because they require a human in the loop. The service description with WSDL can be parsed automatically and invoked by machines, but the interpretation of their meaning is left for a human [Sab06].

To support automation and interoperability in service ecosystems, Service Descriptions (SDs) are needed, which contains the formalization of functional and non-functional parameters of services [Sab06]. The non-functional parameters allow service providers to define business related aspects also in SDs. These parameters allow service providers to re-combine and mediate services of other providers to compose new services [BD06]. The service description and service description language are described in the following sections.

2.1.3 Service Description

The service discovery, selection, and negotiation requires the description of functional and non-functional parameters, e.g. price models, general terms, and availability. The functional description is not sufficient for the service discovery and selection in today´s applications. Particularly price and Quality of Service parameters (QoS), e.g. availability, are most important. Typically, users will not decide for a service - even if it fulfils their functional requirements - if it can not meet their requirements about price or QoS parameters. They may compromise on their functional requirement rather than on nonfunctional requirements [HKRK07].

A separate consideration of functional and non-functional requirements is not the best way to approach the problems of service discovery, selection and negotiation. The distinction between functional and nonfunctional attributes of a service is artificial and often arbitrary: Should the prices be regarded as a functional or a non-functional parameter? Consider for example two offers, both offering map downloads for cities. Offer A offers map downloads in resolution of 1:10,000 over a slow network connection for a price of 0,50 Euro per map. Offer B offers map downloads in resolution of 1: 15,000 over a fast connection for a price of 0,60 Euro per map. In this case, the decision for functional and non-functional parameters is not clear. The ´´name of the city” is undoubtedly a functional parameter. The both other parameters ´´resolution” and ´´price” can be considered either as a functional or a non-functional parameter. This example shows also that just functional parameters are not sufficient for service consumers to decide for a service.

According to [HKRK07], there are three different categories of functional and nonfunctional attributes.

- Static attributes
- Dynamic attributes within the influence of the service provider
- Dynamic attributes beyond the influence of the service provider

The value of static attributes does not change over time and they can be a static part of the service description. Static attributes can be functional, e.g. types of notebooks sold by an online trader, or non-functional, e.g. price per picture offered by a photo printing service, the resolution offered by a printer, and the delivery time offered by a shipment company.

The value of dynamic attributes can change over time. The dynamic attributes are either within the influence or beyond the influence of service providers. The dynamic attributes within the influence of service providers allow the offering configurable services. The price attribute of an airline reservation service is an example of a dynamic attribute within the influence of the service providers. The price of airline reservation may change over time depending on holiday season or current booking status of flights. These attributes can again be functional or non-functional.

Dynamic attributes beyond the influence of service provider are most challenging. The value of these dynamic attributes change over time. Examples of such attributes are the available bandwidth, the response time or the reputation of the service provider. These are typically characterized as QoS parameters. They can be defined by service providers as estimated values, but the real values of these parameters can be provided by a monitoring service of the service providers. Typically, functional attributes do not belong to this category.

Service Description Languages

The service description captures static and dynamic attributes of services. The most commonly used languages for the service description are XML, Ontologies (OWL, RDF(S), etc.), and UML. These SD languages are described in this section, because the VSA approach has to support the different formats of service descriptions.

XML: The eXtensible Markup Language (XML) was developed to process Web content automatically. XML is a set of rules to decode documents with restrictions, e.g. tag names disallow certain symbols, all tags have to be closed, and names are case sensitive. These restrictions were chosen to achieve a widely accepted interchange format for various (semi-)structured data. Over the last few years, XML has been established as standard exchange format in the Web [FLP+07]. The examples for the XML-based service description approaches are Semantic Modeling Language [PPS10] and Universal Service Description Languages (USDL) [CWV09] etc.

Listing 2.1 shows the notion of a physical address of a person in USDL [IoS10]. The physical address is defined in the form of traditional postal address and represents a specific point-location in the physical world.

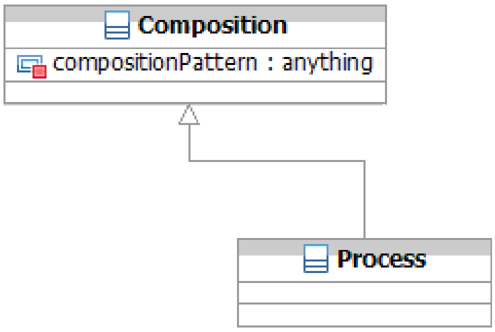
```
<naturalPerson>
...
        <contactProfiles>
                <physicalAddress>
                        <floorNumber> 26 </floorNumber>
                        <buildingNumber> A4 </buildingNumber>
                        <street> W Common Blvd. </street>
                        <streetNumber> 33 </streetNumber>
                        <city> Philadelphia </city>
                        <postcode> 19132 </postcode>
                        <state> Pennsylvania </state>
                        <country> USA </country>
                        <description>
                                <name> Office Address </name>
                        </description>
                </physicalAddress>
        </contactProfiles>
...
</naturalPerson>
```

Listing 2.1: An example of XML based service description

Ontology: The term Ontology originates from Philosophy and it is used as the name of subfield studying the nature of existence. However, in recent years, the term ontology is used by computer science and given a specific technical meaning that is different from the original meaning. According

```
</owl:Class>
```

Another key SOA concept is the notion of *process*. A *process* is a composition whose elements are composed into a sequence or flow of activities and interactions with the objective of carrying out certain work.[27] The concept of *process* is captured by the **Process** class, which is illustrated in the figure below.



Elements in process compositions can be things like Actors, Tasks, Services, other processes etc. A process always adds logic via the composition pattern, the result is more than the parts. According to their collaboration pattern, processes can be:

- Orchestrated: When a process is orchestrated in a Business Process Management System then the resulting IT artifact is in fact an orchestration, i.e. has an orchestration collaboration pattern. This type of process is often called a "Process Orchestration".
- Choreographed: E.g. a process model representing a defined pattern of behavior. This type of process is often called a "Process Choreography".
- Collaborative: No (pre)defined pattern of behavior (model), the process represents observed (executed) behavior.

5.5 Service Composition and Process examples

Include Service Composition, Process, Process/Service composition etc. as examples + Carwash example

[27] This definition is consistent with for instance the BPMN 2.0 definition of what a process is.

Figure 2.4: One example of UML based service description

to Noy and Rector [NR04], an ontology is a formal explicit description of concepts, or classes in a domain of discourse. Properties -or slots- of each class describe various features and attributes of the class, and restrictions on slots (called facets or role descriptions) state conditions that must always hold to guarantee the semantic integrity of the ontology. Each slot has a type and could have a restricted number of allowed values. Allowed classes for slots of type instance are often called a range of slot. The Service Ontology [OLG+06] and SOA Ontology [Har10] are examples of Ontology based service description approaches.

There are different languages for defining ontologies, e.g. RDF(S), OWL, and F-Logic [Mae02]. One example of OWL based service description (taken from SOA Ontology [Har10]) is depicted in Listing 2.2. This example shows the notion of a "process". A process is a composition whose elements are composed into a sequence of flow of activities.

```
<owl:Class rdf:ID="Process">
        <rdfs:subClassOf>
            <owl:Class rdf:ID="Composition"/>
        </rdfs:subClassOf>
    </owl:Class>
```

Listing 2.2: An example of XML-based service description

UML: The Unified Modeling Language (UML) is a language to specify the artifacts of software systems, business modeling and other non-software systems. It has become a de facto standard for the object-oriented modeling. The UML along with Meta Object Facility (MOF) offers a foundation for the Model-driven Architecture, which supports the whole service engineering process from modeling till deployment [Lar02, UML10].

The OWL example in the Listing 2.2 can be also modeled with UML as it is depicted in Figure 2.4. Figure shows the UML notion for the modeling of

SOA concept process [Har10].

2.1.4 Service Life Cycle

Different aspects of service life cycle have to be considered and specified in service description. Therefore, the service life cycles are discussed in this section. Different service life cycle models have been proposed by the research community. These service life cycle models range from very generic life cycle

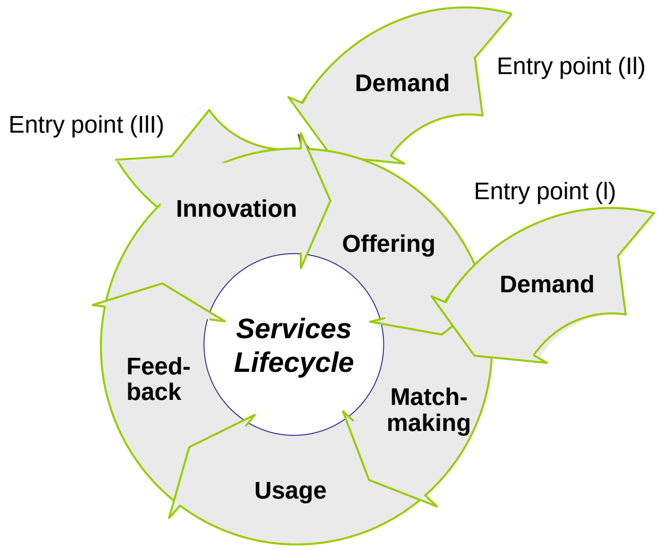


Figure 2.5: Service Life Cycle

The service lifecycle according to Oberle et al. [OBB+09] is described in the following, because it discusses service description modeling issues briefly. The service lifecycle comprises multiple phases innovation, offering, matchmaking, usage, and feedback as depicted in Figure 2.5 to offer new services in service ecosystems. Three possible entry points are offered in the service lifecycle. They can be initiated by a service provider, a service consumers or the business community. Service consumers can initiate entry point (I) in matchmaking phase of the service life cycle by ordering a new service to meet their needs. Service providers can initiate the entry point (II) in the service offering phase to meet the demand of service customers. The initiation of the entry point (III) takes place in the service innovation phase, where service providers or service brokers realize demands of new value added services. These three entry points initiate the design and development of new services. In the following paragraphs, the different phases of the service lifecycle are described in detail.

Innovation: The implicit and explicit feedback from the usage phase initiates the innovation phase. Service providers can analyze the feedback of service consumers and redesign services (if necessary) to meet the needs of service consumers. The innovation phase can also initiate new business models, consumption and development paradigms. Not only service consumers and service providers can initiate ideas for new services, but service brokers and service mediators can also initiate ideas for new services. The innovation phase plays an important role for the growth of service ecosystems by initiating new service ideas [RMF+09, Che06].

Offering: The design and implementation of services take place in the offering phase. The business process, interaction behavior and the user interface are designed here. The reuse, adaption and modification of existing services and the composition of different services are also done in this phase. These aspects require data mediation to harmony different service input and output formats, which are part of service offering. Furthermore, semantic enriched service description with functional and non-functional parameters such as the pricing model, general terms, penalties and the quality of service parameters [OBB+09, HKRK07, OSEH05] have to be specified during the service offering. These functional and non-functional aspects offer service consumers a flexible service discovery in the service ecosystems. Service providers have to publish the semantic description of services in such a way that they can be discovered by service consumers.

Matchmaking: The Service Matchmaking phase brings service the consumer's needs and service the provider's offers together. The matchmaking can take place manually or consumer-driven by using a discovery function, which allows a service consumer to search for services. The semantic enriched service description allows service consumers to formulate queries in automated manner to find adequate services according to their needs. Service consumers can use context information and the quality of service parameter from the semantic enriched service descriptions to search appropriate services [OBB+09, BRS+08]. Matchmaking is part of service delivery, as it is the first step in bringing services to consumers. After the selection of services, service consumers negotiate a Service Level Agreement (SLA) by specifying parameters, e.g. availability and response time with the service providers. The negotiation process typically requires a complex interaction between service consumers and service providers.

Phases of the service life cycle	Service life cycle roles
Innovation	Service provider, service broker
Offering	Service provider, service broker, service mediator
Matchmaking	Service provider, service broker, service consumer
Usage	Service consumer
Feedback	Service consumer

Table 2.1: The roles in the different phases of the service life cycle

Usage: The service usage phase is the main part of service delivery. Service consumers can buy a service and consume services through different channels, like mobile devices, web portals, Rich Internet Applications (RIAs), or business applications. The service consumption or invocation means interaction between the client and the services that involves messages being exchanged between them. The Service Level Agreements (SLA) specific parameters, like availability and response time, will be monitored in the usage phase to guaranty the quality of service for service consumers [HKRK07, BRS+08] .

Feedback: Service consumers can provide their personal rating as explicit feedback using text-forms for the functional and non-functional performances of services, e.g. three stars rating the response time of a service. Further feedback, like log files or usage statistics as implicit feedback, can be collected automatically. The explicit and implicit feedback helps service providers to redesign their services to improve the quality of service and achieve higher acceptance by service consumers. Beside monitoring of SLA specific parameters, the implicit and explicit feedback are the most important instruments to improve the quality of services [OBB+09, BRS+08].

Service Life Cycle Roles

The four main roles in the service life cycle are identified: service provider, service consumer, service broker and service mediator [RBRK09, BD06, BDB05] . Figure 2.6 gives an overview of the service life cycle roles. The table 2.1 depicts these roles in the different phases of the service life cycle.

Service providers produce and publish services in the Internet of Services. They are mostly the owner of services and responsible for implementation and maintenance of services. Service providers can develop new services from scratch and combine or extend existing services to offer value added services. They are also responsible for the process description and implementation of services. Furthermore, they provide functional and non-functional service descriptions, e.g. pricing models, general terms, and SLA paramen

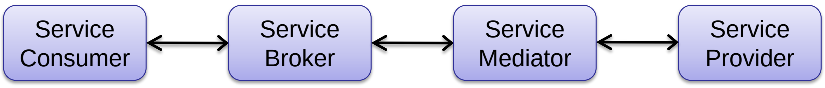


Figure 2.6: Service life cycle roles

Service consumers can find and consume offered services in the Internet of Services. They can be a guest, an expert or a business consumer. As guest, they can just browse through offered services without buying them. Business consumers have deeper technical knowledge of services and are interested in buying services. The functional and non-functional parameters from service descriptions offer flexible service search to service consumers. Once they have found a service, the service level agreement has to be negotiated between service consumer and service provider. When SLAs are concluded, service consumers can consume services in the usage phase of the service life cycle[OBB+09, BRS+08, Ber07]. They provide their feedback after the service consumption in the feedback phase of the service life cycle.

Service brokers bring service providers and service consumers closer. They offer service catalogs with different search functionalities to service consumers. Service catalogs contain published services from different service providers. Services brokers can also combine services from different service providers and offer them in their service catalogs. Service brokers offer services in their service catalogs according to constraints of service providers e.g. pricing models, quality of service parameters and SLAs. Service consumers can buy services either from service brokers or directly from services providers [RBRK09, Ber07]. Service brokers are involved in the phases innovation, offering, and matchmaking of the service life cycle.

The combination of services from different service providers to develop new value added services may need translation between different service formats and other routine functions. *Service mediators* offer translations between different service formats and other routine functions in the ofering phase of the service life cycle. Service providers, service mediators and service brokers have to work together to offer service to service consumers [RBRK09, BD06]. The interplay of supply, distribution and consumption roles is shown in Figure 2.6.

2.2 Semantic Analysis

The terms service, SOA and service description are introduced in the last section. The service life cycle and different roles in the service life cycle are also introduced. These roles e.g. service provider, and service broker have to be supported during the modeling of SDs by the VSA approach. They apply the VSA approach in the different phases of the service life cycle to model the different aspects e.g. price plans (offering), feedback (usage), service ideas (innovation) of services. The application of the VSA in the different service life cycle phases is discussed briefly in Chapter 6.

The semantic analysis techniques facilitate the analysis of available services to extract semantic information and their usage during the modeling of new services in the VSA approach. Furthermore, the semantic analysis techniques allow the analysis of new modeled services with available services to compare new services with available services in the VSA approach. Therefore, the semantic analyis techniques, and the usage of visualization and Visual Analytics (VA) techniques for the semantic analysis are discussed in this section.

Semantic Analysis relies on the paradigm *balanced cooperative modeling*, which is defined as a coordinated interaction between a human modeler and learning algorithms. According to this approach, semantic analysis supports cooperation between knowledge experts and machine learning algorithms. It also deals with learning techniques and algorithms that can be applied for an efficient support of semantic description modeling by using data from available knowledge sources, e.g. Web. Furthermore, it addresses refinements of semi-automatic generated semantic descriptions of information artifacts. There are different semantic analysis processes, which will be discussed in the Chapter 3 briefly.

2.2.1 Knowledge Sources for the Semantic Analysis

The structured, semi-structured, and unstructured data like ontologies, HTML-Documents or Natural Language Documents (NL Documents), can be considered as knowledge sources for semantic analyis. The NL documents are very important for the semantic analysis and freely available in large amounts on the World Wide Web (WWW). The documents are also available as domain independent and are domain specific form, which make them the most interesting source for semantic analysis. The NL Documents can be classified into two categories "pure natural language text" and "natural language documents enriched with semi-structured information". Pure natural language text exhibit morphological, syntactic, semantic, pragmatic, and conceptual constraints that interact to transport particular information to the reader. The documents in the web are natural language documents enriched with semi-structured information, e.g. the information contained in tables and lists [Mae02].

Ontologies or semantic models can be considered as a specific type of structured data. Different available ontologies e.g. WordNet [Fel98], GermaNet [HF97], Thesauri [Wer85] or domain specific web ontologies can be used to drive more comprehensive ontologies [Mae02]. For example, the enhancement of the semantic analysis process can be improved by the automatic discovery of the WordNet-style lexico-semantic relations by searching for corresponding lexico-syntactic patterns in large text corpus [Fel98]. Semantic descriptions are applicable and beneficial to various challenges, which are faced by the software engineering (SE) domain. It seems obvious that the engineering and modeling efforts contained in form of semantic description, e.g. WSDL [CCMW01], service ontologies [OBB+09], or USDL [CWV09] may be used for the semantic analysis. For example, a WSDL file contains input/output parameters, function names and textual information about a service, which may be used for semantic analysis.

To parse NL documents available in the World Wide Web (WWW), web crawlers are commonly used by search engines. By using ontological background knowledge, the crawler focuses the search in the web space and collects domain specific NL Documents. A detailed introduction to focus crawling to collect documents and an evaluation are provided in [Mae02]. Ontologies have different complex underlying representation languages, e.g. OWL. To import and process different sort of ontologies, they have to be transformed into a representation that may be used in the semantic analysis process. Furthermore, given ontologies have to be merged to achieve a common ontology.

The semantic descriptions are mostly available in XML formats with specific schemata, e.g. WSDL [CCMW01], service ontologies [OBB+09], or USDL [CWV09]. The wrappers allow extraction of particular information from the semantic description and deliver as a self describing representation. In our context, the wrapper converts semantic analysis related information implicitly stored in the semantic description into explicit information for further processing.

2.2.2 Semantic Extraction Algorithms

The main semantic extraction algorithms are classified into four different categories: statistics based, rule-based, hybrid (statistics and rule based), and formal [SP07, Zho07, CHS05, Mae02]. Semantic extraction techniques are mostly unsupervised learning methods because training data with background knowledge is not available. Therefore, statistical techniques are often used for the semantic extraction. A *statistical model* represents probabilistic dependencies between random variables. The concepts and relations are extracted by using statistical information, which is calculated by observing frequencies and distributions of the terms in the corpus. The *rule-based techniques* are pattern-based approaches and use heuristic methods using regular expressions. The pattern based approaches for extraction of semantic structures is introduced by Maedche [Mae02]. The *hybrid approaches* combine the strength of the statistics-based and rule-based techniques. *Formal Concept Analysis* (FCA) is an unsupervised learning technique for the semantic extraction. The semantic extraction algorithms are not the main focus of this thesis. Therefore, some semantic extraction algorithms, which are related to this thesis, are described in the following sections. For the detailed overview of semantic extraction algorithms, we refer to stat of the art report presented by Zhou [Zho07].

Lexical Entry and Concept Extraction

The lexical entry and concept extraction is *rule based semantic extraction technique* to acquire lexical entries and corresponding concepts. Web documents have to be morphologically processed by using text preprocessing techniques which are described below.

Part of Speech Tagging (POS-Tagging): The semantic analysis process may be applied without pre-processing, but additional linguistic preprocessing is used to enhance the semantic analysis process. The POS-Tagging determines the part of speech tag for each term, e.g. noun, verb, and adjective. It is based on the observation that the concepts in domain ontologies are usually nouns in the extracted information [CHS05, HNP05].

Reduction of stop words: The stop words, like articles and other terms that do not constitute the idea and concepts of documents, can be eliminated. For example words such as “a”, “an” and “the” can be eliminated from the extracted information. The preposition and conjunction should not be eliminated because they allow the usage of phrases within the semantic analysis process [Mae02, PF09b].

Stemming: The extracted information may conclude different writing styles or variations of the words to define one concept. The usage of plurality, verbal nouns and tenses can alter from the basic form of a word. In the task stemming, the different forms of the words are replaced by their root word. For example, the words “Fish”, “Fishes” “Fishing” and “Fished” are replaced by “Fish” [Mae02, PF09b].

Lexical analysis: The lexical analysis converts the extracted information into “n-gram”, i.e. arbitrary strings of n consecutive terms. This can be done by moving a window of n terms through extracted information. The n-gram models with n= 1 to 6 can be used to enhance the semantic analysis process. The lexical analysis allows the use of not only terms, but also phrases for semantic analysis. [PF09b, Gre00].

Document term matrix: After the lexical analysis, the extracted terms and documents related to terms can be inserted in the document term matrix that contains documents, terms and term frequency inverted document frequency (*tfidf*) . The *tfidf* weights of lexical entry in documents and are used to evaluate how important a word is to a document in a collection or corpus. The $tfidf_{l,d}$ defined is given by [Mae02]

$$tfidf_{l,d} = lef_{l,d} + \left(\frac{|D|}{d_{f,l}}\right)$$

where $lef_{l,d}$ is the lexical entry frequency of the lexical entry l in a document d, $d_{f,l}$ is the overall document frequency of lexical entry l and $|D|$ is the set of documents in the corpus.

Dictionary Parsing

Dictionary parsing is a *rule based semantic extraction technique.* The existing domain knowledge [Fel98] can be exploited for the extraction of domain conceptualization. For example, WordNet is a network of semantic relations among English words. It allows the extraction of synonyms by using the basic synset relation of WordNet. In addition to synonyms, the different semantic relations e.g. hyponyms help to extract real concepts of a word. These semantic relations allow mapping of words to more general concepts or merging synonyms [WH02]. For Example, two terms *car* and another one *truck* can be mapped to *vehicles* by using WordNet.

The pattern matching heuristic can be applied to extract lexical entries from morphological processing of text to identify semantic relations. For example, the dictionary from insurance company contains the following entry [MS01]:

Automatic Debit Transfer: Electronic service arising from a debit authorization of the Yellow Account holder for a recipient to debit bills that fall due direct from the account.

A simple heuristic relates definition term “automatic debit transfer” with the first noun phrase in the definition “Electronic Service”. It means both concepts are linked in the concept hierarchy.

Clustering

After lexical entries and concept extraction, the taxonomic classification of concept can be achieved by using concepts clustering techniques. Clustering is a *statistics based semantic extraction technique.* A simple clustering algorithm is k-means. The k-means algorithm clusters objects so that similarity between objects within one cluster is high and the similarity between objects from different clusters is low [SD08, CPSK07, HK01]. The k-means algorithm [HK01] aims minimize the within-cluster sum of squares that defined

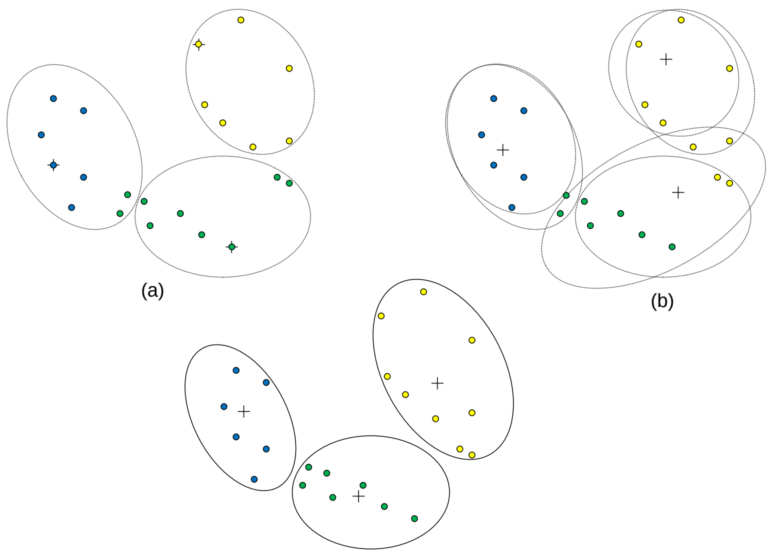


Figure 2.7: Clustering based on k-means algorithm

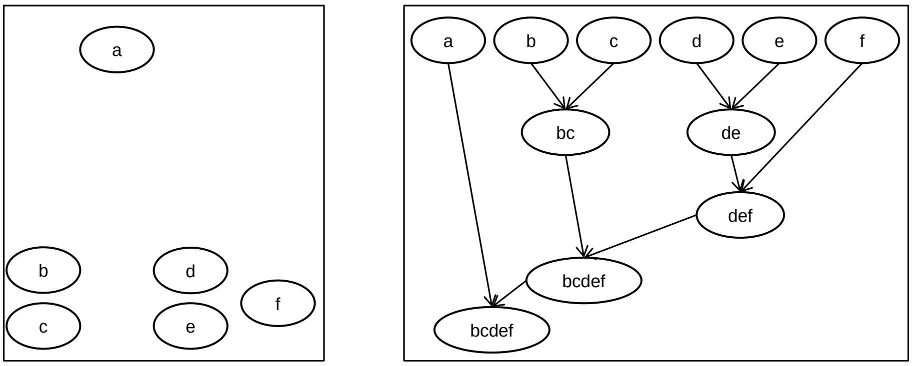
by the follwing equation.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \tag{2.1}$$

Where E is within-cluster sum of squares, p is the point of an object in space, m_i is the mean of the cluster C_i.

Figure 2.7 depicts an example of clustering with the k-means clustering algorithm [HK01]. The three arbitrarily chosen initial cluster centers are marked by "+". Each object is assigned to the nearest cluster based on the cluster center as shown in Figure 2.7a. By recalculation of the mean value of each cluster the cluster centers will be updated. Relative to these new centers, objects are reassigned to the nearest cluster based on the cluster center as shown in Figure 2.7b. This process iterates so long that no redistribution of objects in any cluster takes place. The resulting clusters are shown in Figure 2.7c.

The hierarchical concept clustering proposes a hierarchy of item categories based on similarity of items. The properties of items can be used to measure the similarity of items. In the context of natural language context and adjacency of terms, syntactical relationships between two terms can be used to extract a semantic hierarchy of concepts [Mae02]. There are different hierarchical concept clustering algorithms. The Agglomerative hierarchical clustering algorithm clusters the items according to Euclidean distance met-



(a) (b)

Figure 2.8: Clustering based on hierarchical concept clustering

ric. For example, Figure 2.8a shows items in Cartesian coordinates. These items can be assigned to a concept hierarchy based on the Euclidean distance metric as shown in Figure 2.8b. The items *bc* in Figure 2.8b are assigned as sub concept of *b* and *c* based on Euclidean distance. The Euclidean distance $d(p,q)$ between two points $p = \{p_1, p_2, ..., p_n\}$ and q $= \{q_1, q_2, ..., q_n\}$ is defined as [HK01]. The clustering techniques are introduced shortly, because they are not the main focus of this thesis. For the further clustering techniques, we refer to the research work of Han and Kamber [HK01]

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2.2}$$

Association Rules Learning

The association rule learning algorithms are *rule based semantic extraction technique* and typically used in data mining to find interesting associations in large-scale databases. For example, associations in sales transactions of a supermarket database as depicted in Table 2.2 can be found with association rule learning. An example rule for the supermarket can be $\{milk, bread\} \Rightarrow \{butter\}$. It means that if customers buy milk and bread, then they will also buy butter.

The *apriori* is an influential algorithm for the association rules learning [HK01]. One example of sales transactions mining illustrates the apriori algorithm. The nine sales transaction of a supermarket are shown in table 2.3 [HK01]. The apriori algorithm scans all transactions to count the number of occurrences of 1-itemset C_1 with minimum support count 2 as shown in Table 2.4. Then it scans 2-itemset C_2 with minimum support count 2 as

Transactin ID	Milk	Bread	Butter	Beer
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Table 2.2: Example data for supermarket transations

Transactin ID	List of items ID
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Table 2.3: Sale transactions of a supermarket

1-Itemset C_1	Support count
$\{I1\}$	6
$\{I2\}$	7
$\{I3\}$	6
$\{I4\}$	2
$\{I5\}$	2

Table 2.4: 1-Itemset C_1 with support count 2

1-Itemset C_2	Support count
$\{I1, I2\}$	4
$\{I1, I3\}$	4
$\{I1, I5\}$	2
$\{I2, I3\}$	4
$\{I2, I4\}$	2
$\{I2, I5\}$	2

Table 2.5: 2-Itemset C_2 with support count 2

shown in Table 2.4. Finally, it generates 3-itemset C_3 with minimum support count 2 as depicted in Table 2.6.

In the semantic analysis, a syntactically related classes (e.g. pair (Festival, Island) extracted as head-modifier relationship from the sentence "The fes-

1-Itemset C_2	Support count
$\{I1, I2, I3\}$	2
$\{I1, I2, I5\}$	2

Table 2.6: 3-Itemset C_3 with support count 2

tival on Island attracts tourists from all over the world") are given as input. The association rules learning algorithms generate rules and propose them to knowledge experts in order to model relationships between classes. The number of generated rules is typically very high, which can be restricted by using different strategies [Mae02].

Formal Concept Analysis

Formal Concept Analysis (FCA) is an unsupervised learning technique to form structures from data, which is based on the creation of natural clusters of objects and attributes. The purpose of the FCA is to offer data mining operations against data collections to analyze semantics within data collections [Duc07, Pri06]. FCA is a *formal technique* for data analysis, knowledge presentation and information management proposed by Wille [Wil82] and formalized by Ganter and Wille [GW99].

Philosophically, a concept is a unit of human thoughts. A concept can be defined by all attributes which describe the concept (the intent), or all objects which are members of the concept (the extent). Numerous objects and attributes can be used to define real world concepts, which make the representation of the real world units of thoughts difficult. That is why Formal Concept Analysis works within a *context*, which has fixed objects and attributes [Duc07].

Table 2.7 depicts an example of a formal context of the real estate domain [Haa04]. The objects of the context are real estate objects and the attributes are real estate specific attributes like family house, country house, or summer house. The rows represent objects and columns represent attributes in the formal context of real estate domain. The "X" in the table indicates, which attributes describe an object (represented by a row) or which object has an attribute (represented by a column).

The formal context can be defined as $K(G, M, I)$ where G is set of objects, M is set of attributes and $I \subseteq G \times M$. A formal concept of the context $K(G, M, I)$ is defined as pair (A, B). For $A \subseteq G$ and $B \subseteq M$:

$$A' = \{m \in M | (g, m) \in I \forall g \in A\} \tag{2.3}$$

$$B' = \{g \in G | (g, m) \in I \forall m \in B\} \tag{2.4}$$

Objects	Attributes					
	Real	Family	Country	Summer	Blockhouse	Skyscraper
	estate	house	house	house		
A1	x	x				
A2	x	x	x	x		
A3	x		x			
A4	x				x	x
A5	x				x	
A6	x			x		

Table 2.7: Real estate domain context [Haa04]

The A' is the set of attributes, which is common to all objects in A and B' is the set of objects, which is common to all attributes in B. The concept of the context $K\left(G, M, I\right)$ is (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$ and $A = B'$. The extent of the concept (A, B) is A and its intent is B.

For the concepts (A_1, B_1) and (A_2, B_2) from the set S of all the concepts of $K\left(G, M, I\right)$:

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \left(\Leftrightarrow B_1 \supseteq B_2\right) \tag{2.5}$$

The relation $\leq$ is an order on S e.g. the concept (A_1, B_1) is less general as compared to the concept (A_2, B_2), if the extant of the concept (A_1, B_1) is contained by (A_2, B_2).

The complete set of concepts S of a context $K\left(G, M, I\right)$ and the order between concepts $\left(S\left(K\right), \leq\right)$ is a complete lattice and it is known as concept lattice of the context $K\left(G, M, I\right)$ as shown in [GSW05].The concept lattice of the context of real estate domain as depicted Table 2.7 is shown in Figure 2.9 [Haa04]. The rows and columns in Table 2.7 represent real estate objects and real estate specific attribute correspondingly.

The formal concepts are represented by nodes and sub- and super-concept relationships are represented by edges in the lattice (see Figure 2.9). For example, one of the formal concept in the lattice is defined by attributes $\{Realestate, Blockhouse, Skyscraper\}$ and the object $A4$ is assigned to it. The formal concept with the eattributes $\{RealEstate, Blockhouse\}$ is its super-concept and the objects $A4$ and $A5$ are assigned to this super concept.

The use of the Formal Concept Analysis as data analysis technique is still not fully studied for the design of Information Systems. Eklund et al. [EGSW00] introduced a development method based of Toscana System and Hereth and Becker [BC05] introduced the term Conceptual Information System, which is based on Cernato [Pri09] and ConExp [Pri09]. A variety of applications have been developed on the base of Formal Concept Analysis to support knowl-

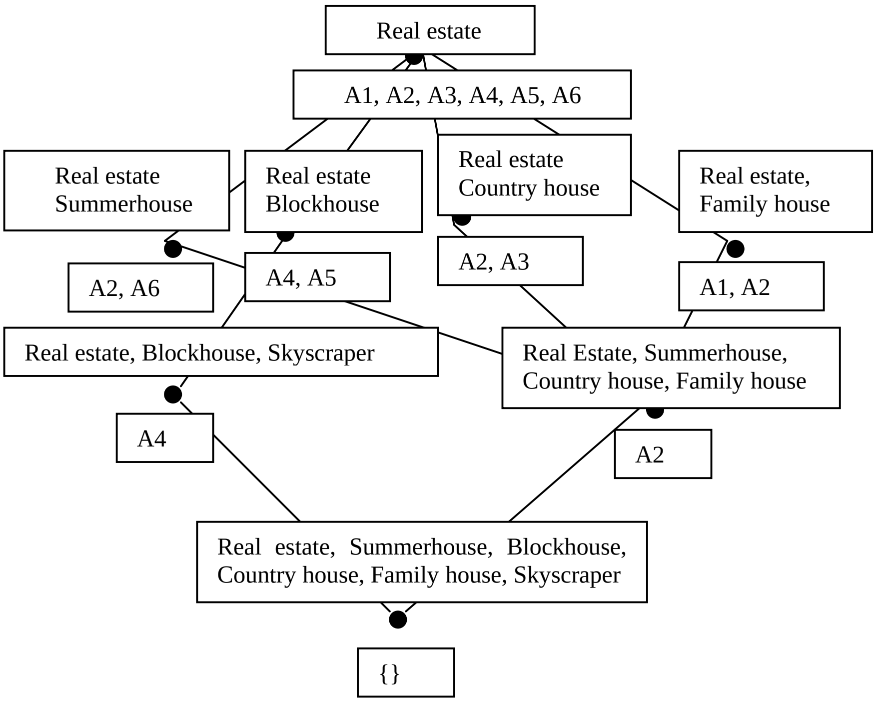


Fig. 1. The concept lattice of a real estate domain

Figure 2.9: The concept lattice of the real estate domain [Haa04]

3.2 Concept Lattice Based Ontology Expression

There is redundant information in concept lattice. The two kinds of redundancy can be eliminated from concept lattice without loosing any information as shown in [3]: redundant elements in formal concepts intents and redundant objects in formal concepts extents. Our reduction procedure has 2 steps: elimination of redundant elements from formal concepts intents and elimination of lattice of extents.

Elimination of redundant elements. For a pair (A, B), B (intent) will appear in every descendant. The inherited elements may be eliminated. Let B´ be the set of elements in B that do not appear in any descendant of B. Then we can consider the concept lattice, where the nodes contain the pairs (A, B´) instead of pairs (A, B). This lattice is a result of the first step of our reduction procedure.

Elimination of lattice of extents. After elimination of redundant elements from concept intents, we eliminate lattice of extents L_O and get as a result a reduced lattice of intents L_{CR} of formal concepts.

We call the resulting lattice L_{CR} of reduction procedure as concept lattice based ontology expression. Fig. 2 shows the lattice L_{CR} of our example.

Naming concepts. After reducing concept lattice to its intentional part, we need to give formal concepts the names. Naming in our case can be done as follows:

edge management and discovery e.g. MailSleuth [Duc07, CES03, EC02], SurfMachine, DSift [Pri09], ImageSleuth [Pri09], SearchSleuth [Pri09].

2.2.3 Information Visualization for Semantic Analysis

Visual techniques take into account human perceptual capabilities to detect patterns and outliers in visual information. The application of visualization techniques in the semantic analysis process allow to view the results of semantic analysis process and facilitate experts' understanding about extracted concepts, their structures, hierarchy and relations in semantic analysis process [Sab05]. The visualization techniques support knowledge experts to validate extracted semantic information or refine it. Sabou has presented the visualization techniques "Cluster Map" to support semantic analysis process. She purposes investigation of new visual techniques for semantic analysis and their integration in the semantic analysis process [Sab05]. The usage of the information visualization techniques is an essential part of the VSA approach, because the VSA approach aims to combine semantic analysis and semantic analysis techniques to support semi-automatic modeling of SDs. That is why, the terms related to information visualizations and some examples of information visualization are introduced in this section.

Information visualization is derived from several communities. Playfair (1786) is seemed to be among earliest, who uses abstract visual properties such as lines and areas to represent data visually. Since then the classical data plotting methods have been developed. Bertin published his theory of graphics in 1967 [CMS99]. This theory identified basic elements of diagrams and described a framework for the design of diagrams. Tufte introduced his theory of data graphics that highlighted maximizing density of useful information. Both well-known theories from Bertin and Tufte led to the development of information visualization as discipline [CMS99].

Foyel [FR94] has defined visualization as follows:

> *A useful definition of visualization might be the binding (or mapping) of data to a representation that can be perceived. The types of binding could be visual, auditory, tactile, etc., or a combination of these.*

The main focus of Foyel and Ribarsky´s definition of visualization is the mapping of data to visual forms to improve understanding of users about data. The visualization can be seen as communication medium between data and users according to Foyel and Ribarsky´s definition. The definition of visualization provided by Card et al. [CMS99] underlined the objectives of visualization:

> *The use of computer-supported, interactive, visual representations of data to amplify cognition.*

According to them visualization can be categorized into the two subfields scientific visualizations and information visualizations. This categorization is based on the type of data to be visualized. Scientific visualizations are based on physical data, e.g. the human body, the earth, molecules etc. The physical data allow spatial mapping, e.g. ozone concentration in the atmosphere, can be visualized on a physical 3D representation of the earth. The non-physical data, such as financial data, business information, collection of documents and abstract conceptions, do not have an obvious spatial mapping. The visualization of non-physical data has two major problems: visualization of objects´ properties and mapping of non-spatial abstractions into an effective visual format. Therefore, information visualization is defined as follows:

> *The use of computer-supported, interactive, visual representations of abstract data to amplify cognition [CMS99].*

According to several authors [God09, Dom00, CMS99], the objectives of information visualization can be summarized as:

- Improvement of exploration/exploitation of data and information by using visualizations;
- Enhancing understanding of concepts and processes by using knowledge and process visualization;
- Gaining new (unexpected, profound) insights with the help of visualization techniques;
- Making the invisible visible;
- Effective presentation of significant features by using visual forms;
- Increasing scientific productivity by assisting experts with visual techniques;
- Improving communication/collaboration by using visualization as medium.

The following paragraphs describe information visualization fundamentals like human visual perception and the visual information seeking mantra, and present different visualization techniques.

Human Visual Perception

The visual structures in the information visualization are dependent upon the properties of human perception. According to Ware [War04] a simplified information-processing model of human visual perception consists of three stages as shown in Figure 2.10. In the stage one, billions of neurons work in parallel and extremely fast to extract features such as orientation, color, textures and movement pattern from visual fields. This parallel process proceeds whether we like it or not and it is independent of what we choose to attend to. To make information visualization understandable quickly for people, it should be presented in such a way that it can be detected in this stage by the large and fast computational system in the brain. At the second stage, the visual fields are divided into regions and simple patterns, e.g. continuous contours, regions of the same color and regions of the same textures. The slow serial processing of visual fields takes place in this stage. The pattern-finding stage of visual process is in state of flux, a combination of bottom up features processing from the first stage and a top down attention mechanism.

At the highest level of perception, the objects are held in visual working memory by the demands of active attention. At this level, only few objects can be held at a time that are related to available patterns providing answers

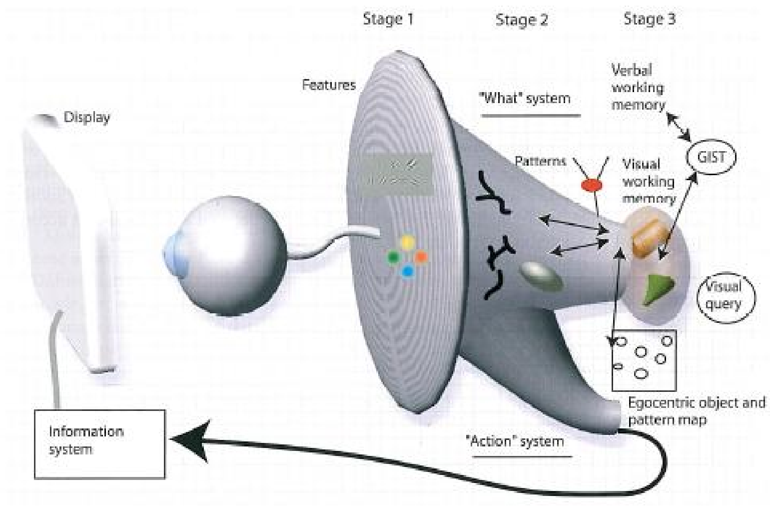


Figure 2.10: Human Visual Perception [War04]

to the visual queries. For example, the visual query triggers search for major highways connecting two locations during the route planning with a road map.

Some suggested approaches by Ware [War04], which improve effectiveness of information visualization for users:

- Reveal fine details where it is necessary
- Usage of luminance contrast is essential.
- Consider color, elements of form (orientation, size), position, simple motion and stereoscopic depth as separate channels for the dimensions coding.
- Reflection of relationships between data entities or glyphs (graphical objects) by visual structures. For example, linking graphical objects with lines or enclose them within a contour represent a relationship among them. The perception of contours is likely to be especially important to see patterns.

Visual Information Seeking Mantra

Shneiderman [Shn96] has presented the “Visual Information Seeking Mantra” as visual design guidelines or basic principles, which can help during the design and conception of information visualization application. The “Visual Information Seeking Mantra” summarized as:

Overview first, zoom and filter, then details-on-demand [Shn96].

According to the "Visual Information Seeking Mantra", Shneiderman [Shn96] also suggests high-level tasks that an information visualization application should support:

1. Overview: Gain an overview of the entire collection.
2. Zoom in on items of interest.
3. Filter out uninteresting items.
4. Details-on-demand: Select an item or group and get details when needed.
5. Relate: View relationships among items.
6. History: Keep a history of actions to support undo, replay, and progressive refinement.
7. Extract: Allow extraction of sub-collections and of the query parameters.

Overview and zoom offer users a detailed view or an abstract view of the data. The zooming factors allow users to see also intermediate views. The overview shows the entire data collection plus details. The zooming strategies, e.g. zoom, focus and zoom factor features, should helps users to focus on their point of interest in data collections. Smooth zooming helps users to preserve their sense of position and context.

Filter out uninteresting items from data collections is the key feature in information visualization. It enables users to focus on their interest by eliminating uninteresting items. Usually the detailed information is hidden in information visualization applications. They will be offered to the user by the details-on-demand feature. Users can select a single item or a group of items to get detailed information on demand. The relationships among items enable users to understand specific patterns in the data collections.

Information exploration to achieve the desired outcome is a complex process with many steps. Keeping history of users´ interaction enables user to retrace their steps and go back to previous steps, if the desired outcome is not achieved during information exploration. The extraction of the desired sub-collections during information exploration would be useful for users to facilitate other uses, e.g. email or print.

Information Visualization Examples

Different visualization techniques (ca. 300 - 400) [NBB+10] are available right now, which can be suitable for a specific task, aspect, data or data

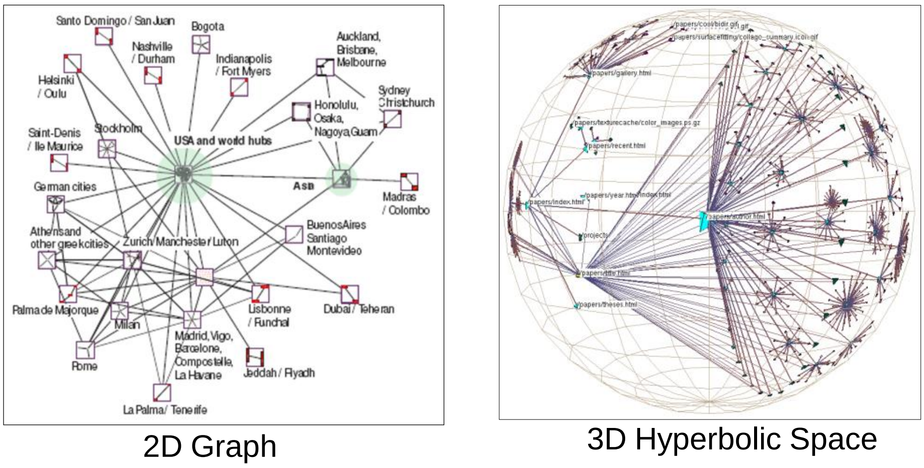


Figure 2.11: 2D graph [KAF$^+$08] and 3D hyperbolic space visualization [Mun97]

format [God09]. The main focus of this work is on semantic structures such as abstract conceptions, which plays an important role for the semantic analysis. We now give an overview of visualization techniques, which can be used for the semantic analysis tasks.

Graph visualizations are used in various areas of application. For example, graphs are applied in biology and chemistry to visualize evolutionary trees, molecule structures, chemical reactions or biochemical pathways. In computing, graph visualizations are used for data flow diagrams, entity relationship diagrams, semantic networks and knowledge representation. Furthermore, social network visualization is also one of the important applications of graph visualizations [HMS00].

Graph structures, e.g. oriented/non-oriented graphs, trees or cyclic graphs and size of graphs are key research issues of graph visualizations. The size of the graph is key issue in the graph visualization research. Large graph visualization is a challenging issue for the performance and layout. Even if it is possible to layout and display large graphs, the usability issues still arises. Large graphs with a huge number of nodes and links requires new visualization techniques, e.g. clustering or focus-plus-context [LKS$^+$10]. 3D hyperbolic space layouts are one example for the visualization of large graphs [Mun97]. Figure 2.11 shows an 2D graph [KAF$^+$08] and 3D hyperbolic space visualization [Mun97].

Trees are classical layouts to visualize hierarchical structures. The classic tree layout positions child nodes below their common ancestor [RT81]. The treemaps can also be used to visualize hierarchical structures as pre-

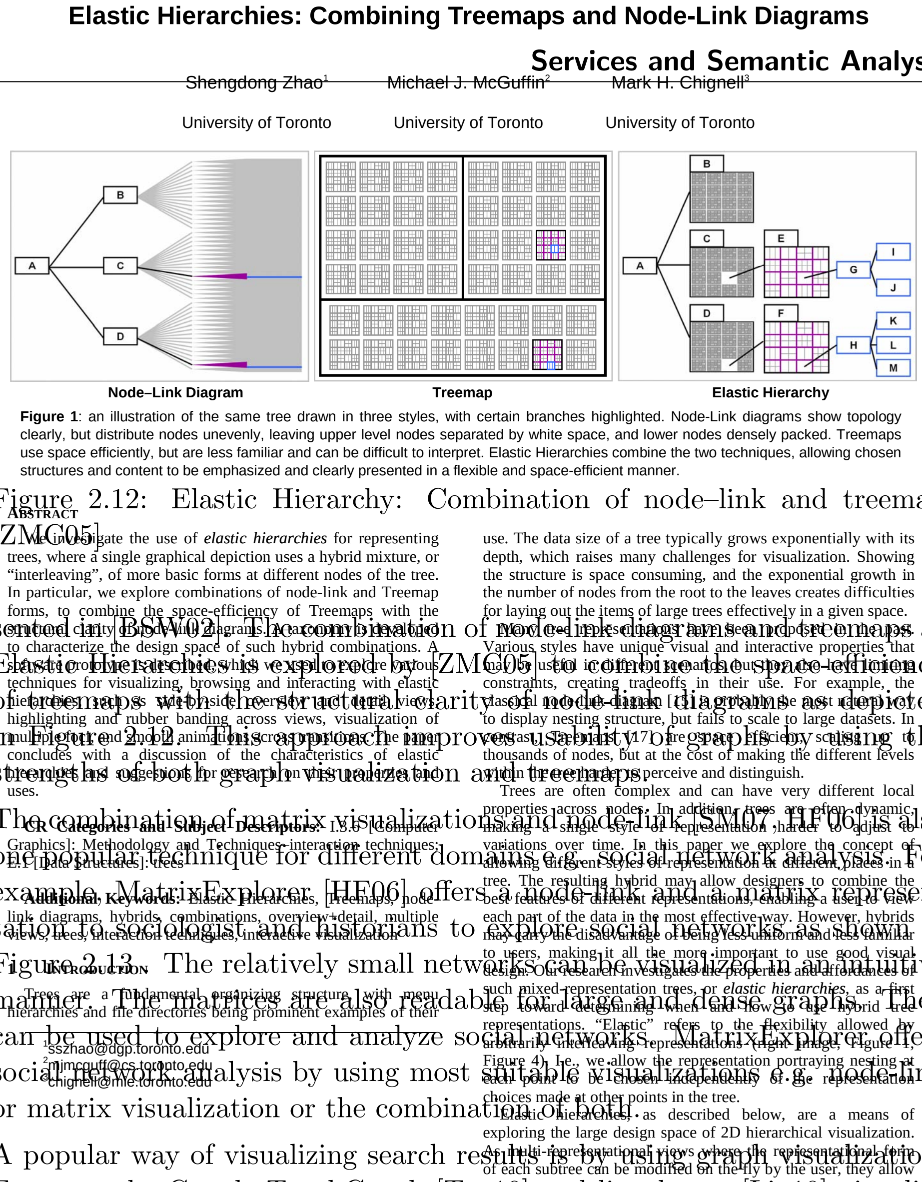


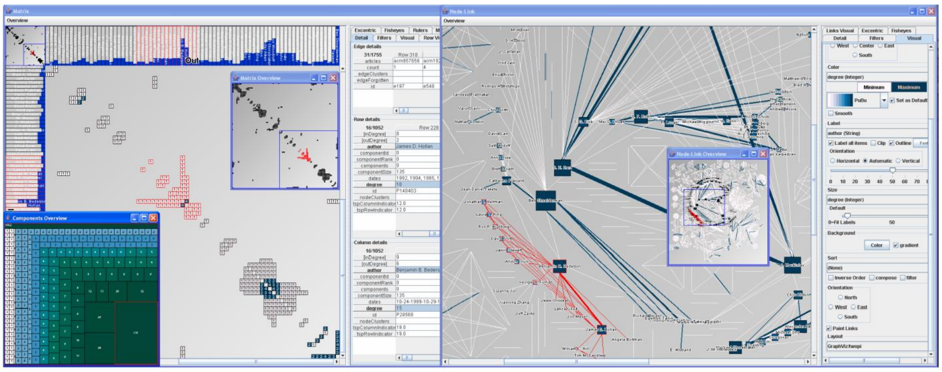
Figure 2.12: Elastic Hierarchy: Combination of node–link and treemap [ZMC05]

sented in [BSW02]. The combination of node-link diagrams and treemaps as Elastic Hierarchies is explored by [ZMC05] to combine the space-efficiency of treemaps with the structural clarity of node-link diagrams as depicted in Figure 2.12. This approach improves usability of graphs by using the strengths of both graph visualization and treemaps.

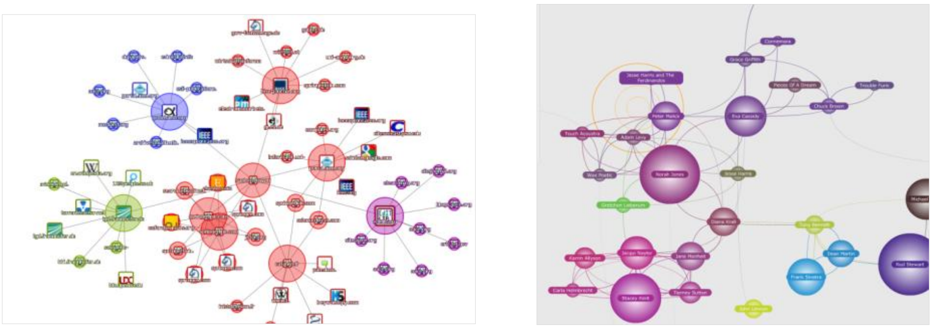
The combination of matrix visualizations and node-link [SM07, HF06] is also one popular technique for different domains e.g. social network analysis. For example, MatrixExplorer [HF06] offers a node-link and matrix representation to sociologist and historians to explore social networks as shown in Figure 2.13. The relatively small networks can be visualized in an intuitive manner. The matrices are also readable for large and dense graphs. They can be used to explore and analyze social networks. MatrixExplorer offers social network analysis by using most suitable visualizations e.g. node-link or matrix visualization or the combination of both.

A popular way of visualizing search results is by using graph visualization. For example, Google TouchGraph [Tou10] and liveplasma [Liv10] visualize search results with graph visualizations as shown in Figure 2.14. Beside nodes and edges, the color and size of nodes represent grouping or popularity of results.

The visualization technique ThemRiver [HHN00] presents how temporal data (e.g. the change of headline stories in the news) can be mapped onto a time scale. The appearance of a specific keyword in a number of articles is visualized over time in the ThemRiver. It shows also the co–appearance of specific themes over time as shown in Figure 2.15. The scatterplot technique [CWR06, Nor00] is often used for the multidimensional data analysis as shown in Figure 2.15. Scatterplot work best for numerical data, which is mapped on x and y coordinates respectively. Because of their simple metaphors, the points in n-dimensional space become points in a 2–dimensional space. They are mostly used to visualize projections of n-dimensional data-space onto 2–dimensional display-space [God09].



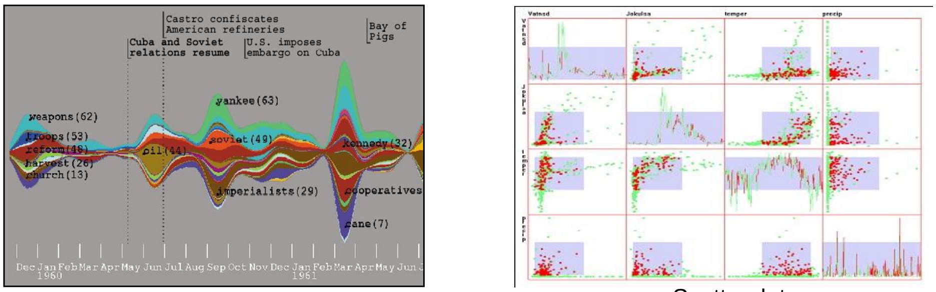
Matrix Visualisation Node-link Diagram



Google TouchGraph Live-plasma

Figure 2.14: Search results visualization by Google TouchGraph [Tou10] and liveplasma [Liv10]

Landscapes or maps [Dod07, Cha93] metaphors are popular way to visualize documents. For example, ThemeScape information analysis and mapping techniques produce "thematic terrains" showing topics or themes of a large collection of documents as map. It uses landscape metaphor e.g. mountains, hills and valleys as shown in Figure 2.16. The elevation of hills and mountains reveals the relative prevalence of different themes. The valleys represent transitions between topics. The closeness of hills represents similarity of their information content. The documents are indicated by small black dots. Self-organizing maps (SOM) are a well developed technique to create interactive information maps to explore a large collection of documents. For example, WEBSOM is developed by using SOM techniques as shown in Figure 2.16. SOM algorithms organize documents automatically



ThemRiver Scatterplot

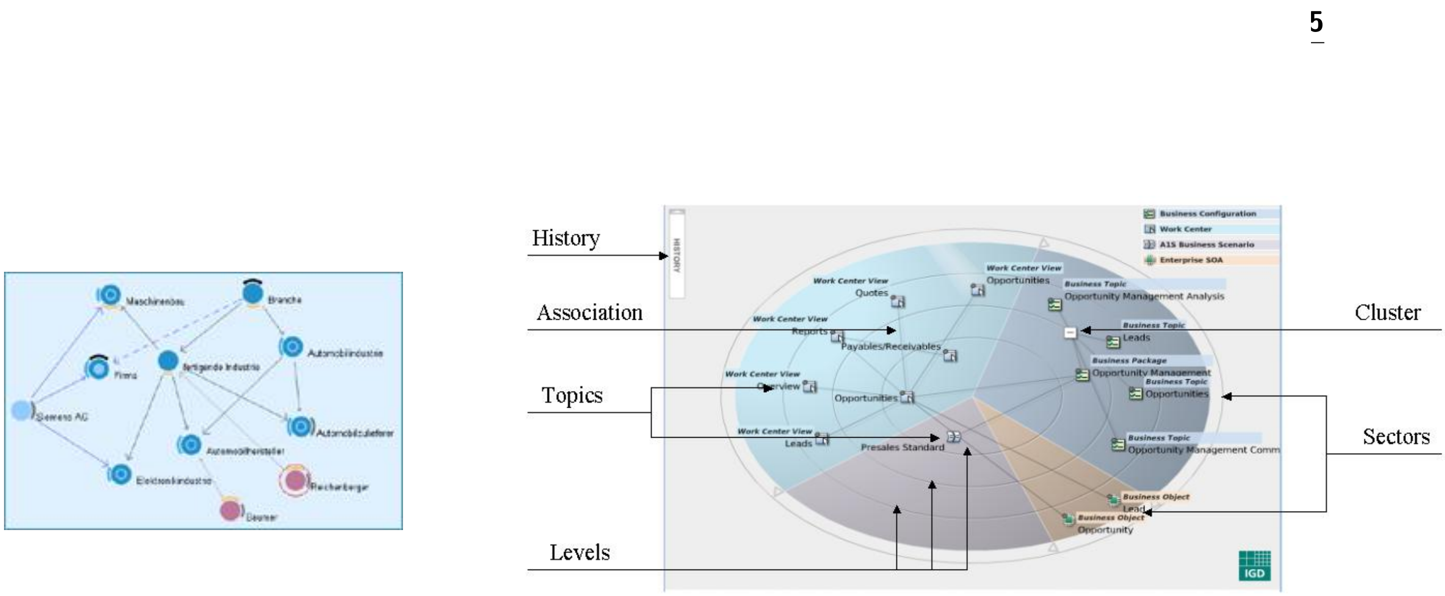
Figure 2.15: ThemRiver [HHN00] and Scatterplot [CWR06, Nor00] visualization techniques

onto a two-dimensional map grid in such a way that similar documents lie near to each other. It can scale to cope with tens or hundreds of thousands of documents

Ontologies allow to structure data by defining a hierarchy of concepts, attributes of concepts and relationship between concepts as described in section 2.1.3. The number of concepts and instances related to them can range from one or two to thousands. The semantic relations between instances can also increase tremendously. Therefore, Ontology visualizations (semantic visualizations) are not an easy task. The complex multi-dimensional semantic relationships between concepts and instances demand innovative visualization techniques to present semantics in a personalized manner, where users can interact with semantics and navigate through networks of semantic information [BW09]. Some semantic visualization examples are described in detail.

Graph visualizations are often used for ontology visualization. For example, the K–Infinity from Intelligent Views [iVi10] and TopicMap Viewer [God09, GB07] visualize semantic networks. Both visualization techniques allow users to navigate through these networks. The K-Infinity as shown in Figure 2.17 places a selected node in the center and positions connected nodes around the selected node. It causes an extreme display change during the user interaction. The TopicMap Viewer minimizes display changes during the user interaction. Furthermore, the TopicMap Viewer offers sector and levels within sectors to visualize affiliation of nodes to specific sectors and abstraction level beside network view of nodes as shown in Figure 2.17 (right). These visualization techniques can be used for social network or business data, e.g. product catalogs.

SemaSpace [BW09, Bha08] offers a knowledge space based visualization for a sophisticated way to explore semantics. It offers knowledge spaces (concepts)



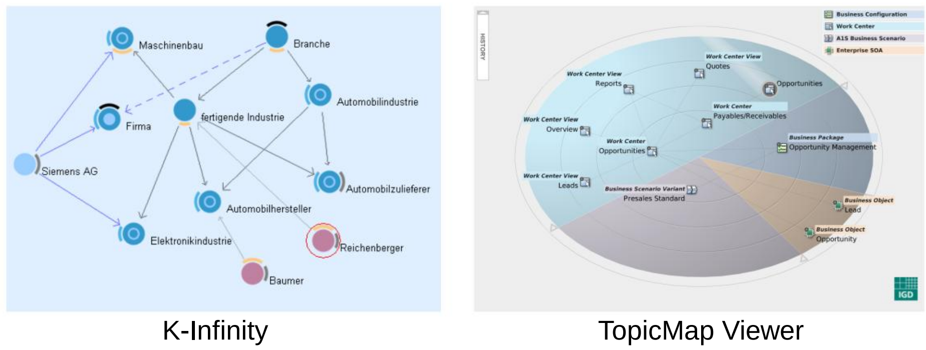
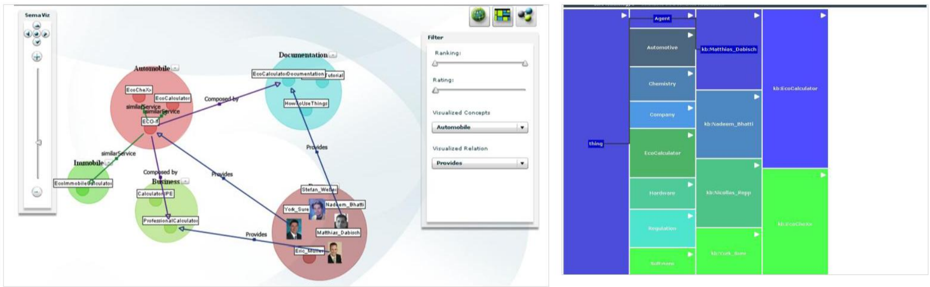


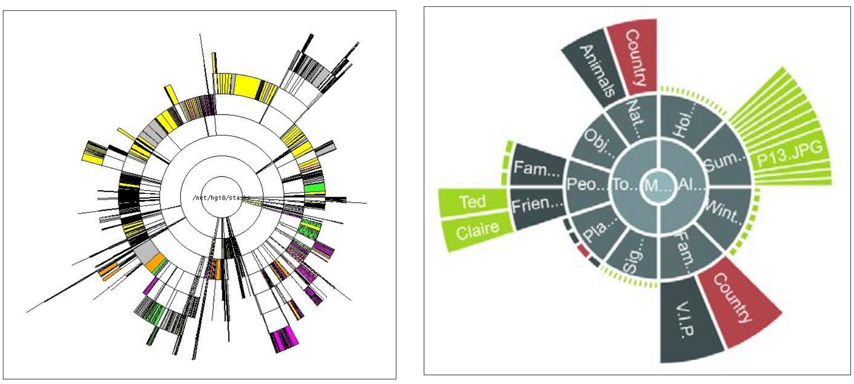
Figure 2.17: K–Infinity [iVi10] and TopicMap Viewer [God09, GB07] visualizations

and related knowledge items to them and interrelation between knowledge spaces and knowledge items as semantic relations. It visualize concepts, sub concepts and instances related to them at the same metaphor as shown in Figure 2.18. SeMap [NBH09] is a treemap based semantic visualization technique, which allows the incremental exploring of the semantic hierarchy. This incremental exploration approach helps to reduce the cognitive load of the user. Users can start with root node and navigate through semantic hierarchy by choosing entities of their interest as depicted in Figure 2.18 (right). The drawback of this approach is that users can just view one single path of semantics at a time.



SemaSpace SeMap

9]



SunBurst SemaSun

Figure 2.19: SunBurst [Sun10] and SemaSun 2.19 visualizations

SunBurst [Sun10] is also an alternative space-filling visualization that uses a radial rather than a rectangular layout as introduced by Shneiderman as shown in Figure 2.19. SunBurst places the hierarchy radially and places the top of the hierarchy at the center and deeper levels away from the center as shown in Figure 2.19. The angle and color represent the attributes of visualized data in SunBurst. For example, in a file system visualization, the angle may represent file/directory size and color may represent file type. A SunBurst based semantic visualization techniques SemaSun is presented by Stab et al. [SBN+10]. SemaSun extends the SunBurst metaphor for semantic visualization. SemaSun Fcannot just visualize hierarchical data, but also multiple inheritance and semantic relations as shown in Figure 2.19. It also allows incremental semantic exploration to reduce the cognitive load of users.

2.2.4 Visual Analytics for Semantic Analysis

The visual Analytics (VA) approach [KAF+08, LKS+10] proposes combination of automated data analysis and interactive visualizations to support effective understanding, reasoning and decision making on the base of very large and complex data sets. It allows knowledge expert to analysis data and visualize the analysis results. The focus of VA approach is the data analysis and semantic extraction, but it doesn't deal with validation, refinement and merging of semantics. The data analysis has to be done, before the semantics can be extracted, validated and refined. The VSA approach uses the VA techniques for the data analysis and semantic extraction, and processes extracted semantic further for the semantic analysis, validation and refinement. Therefor, the VA techniques and some examples are introduced in the following paragraphs.

In the last decade, the way of dealing with information has been influenced by continuously improving data storage devices. Almost every branch of industry or business, and political or personal activity creates enormous amounts of data. The data collection techniques allow collection and storage of data more easily that increase this problem. The huge amount of data leads to an information overload problem that refers to the danger of getting lost in data. For example, irrelevant data available for tasks or inappropriate processing or presentation of data can lead to information overload. Visual analytics techniques provide different solutions to overcome the information overload problem.

The field of visual analytics [KAF+08] proposes combination of automated data analysis and interactive visualizations to support effective understanding, reasoning and decision making on the base of very large and complex data sets. Visual analytics allows users to enter into a loop and inactively manipulate data to gain insight on the data and the representation itself. The visual analytics approaches plays also an important role in the semi-automatic semantic analysis.

According to [KAF+08], visual analytics is defined as:

> *"Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets".*

The goals of visual analytics is to provide techniques to enable people to [KAF+08]:

- Analyze massive, dynamic, ambiguous, and often conflicting data to synthesize information and derive insight.

Figure 2.20: The KVMaps visualization technique

- Pattern analysis and matching to detect the expected and discover the unexpected.
- Formal data analysis methods to provide timely, defensible, and understandable assessments.
- Better presentation to communicate assessment effectively for action.

The difference between visual analytics and information visualization is not always clear, because there are certainly some overlays and some information visualization task are related to visual analytics. Visual analytics is more than visualization. It is an integral approach to combine visualizations, human factors and data analysis. The challenge is to develop a solution by combining best fit automated analysis algorithms, visualizations and interaction techniques. The application of advanced knowledge discovery algorithms in the information visualization community has been limited. Visual analytics tries to solve problems that can not be solved automatically. The process of producing views and creating valuable interaction techniques for a given class of data are the main objectives of information visualization. The tuning of underlying analytical processes is less in focus. Visual analytics aims to give higher priority to data analytics from the start and through all iterations. The learning from users' behavior and effective use of the visualization should play an important role in the analytical process [KAF+08]:

Shneiderman [Shn96] have introduced visual information seeking mantra as design guideline and basic principles as it is described above. Keim et al. [KAF+08] has adjusted the visual information seeking mantra for visual analytics as follows.

> *Analyze first, Show the Important, Zoom, filter and analyze further, Details on demand.*

This mantra proposes intelligent combination of analytical approaches and advanced visualization techniques. Many scalable graph-drawing algorithms try to optimize graph drawing in terms of readability. Some approaches offer a visual abstraction of clusters to improve the readability. The challenge is to introduce a representation that is as faithful as possible to avoid introducing uncertainty. It is naive to assume that visualizations can offer a virgin view on the data. Visual analytics allows users to enter into a loop and inactively manipulate data to gain insight on the data and the representation itself. The visual analytics approaches can also play an important role in the semi-automatic semantic analysis, because they deal with data analysis and visualization techniques to have effective understanding about data.

Visual Analytics Examples

KVMaps visualization techniques [MKJ08] allow knowledge experts to establish an iterative semantic analysis process by combining classification and visualization techniques. They can refine classifiers by using interactive visualization techniques. The knowledge experts start with an initial setting of the classifiers, select data from the display, proceed further with an update of classification and end with visual feedback of the new categories. Figure 2.20 shows the classification process for the demographic data. In every cycle (a), (b) and (c), the selected partition by user is colored black and visual feedback of classification is presented in original color.

Keim et al. [KAF+08] analyze the movement data of people and objects in large scale to optimize location and mobility oriented infrastructures and services. The visual analytics techniques should support human analyst to analyze massive collections of movement data to understand movement behaviors and mobility patterns. The clustering techniques are used to detect frequently visited places and the visualization techniques enable human cognition and reasoning to control the further analysis. Temporal histograms as depicted in Figure 2.21 allow human analyst to explore the temporal distribution of the stops in the frequently visited places with respect to the weekly (left) and daily (right) cycles. Keim et al. [KAF+08] use the same method as KVMaps [MKJ08], but different visualization technique to visualize the analysis results of the massive collections of movement data.

2.3 Summary

The terms related to services, SOA, Service Descriptions (SDs), and service life cycle, which are later used in this thesis, are introduced in this chapter. The examples of statistics based, rule-based, hybrid, and formal semantic extraction techniques provide an overview of different semantic analysis

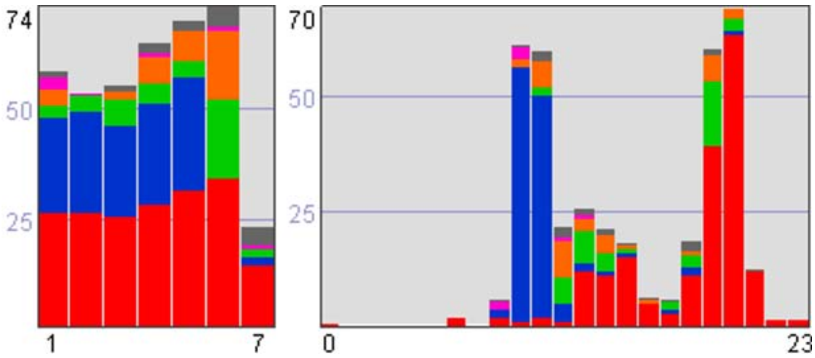


Figure 2.21: The Temporal histograms

Fig. 7. The temporal histograms show the distribution of the stops in the frequently visited places (Figure 6) with respect to the weekly (left) and daily (right) cycles.

techniques. The usage of visualization and Visual Analytics (VA) techniques for the semantic analysis are also discussed in this chapter. These semantic analysis, visualization and VA techniques can be used for the VSA approach. The VSA approach combine semantic analysis and interactive visualization techniques to analyze, model and reanalyze services. Service providers and service brokers apply the VSA approach in different phase phases of the service life cycle to model the ss.

Different manual 3D modeling, semi-automatic 3D modeling, and semantic analysis approaches are introduced in the next chapter that serve as related work for the VSA approach. These approaches help to identify requirements for the VSA approach. The terms related to services and semantic analysis techniques described in this chapter are used to describe and analyze these approaches. The analysis of these approaches is presented in the next chapter.

a map (Figure 6) or 3D spatial view. Temporal histograms (Figure 7) are used to explore the temporal distribution of the stops throughout the time period and within various temporal cycles (daily, weekly, etc.). These complementary views allow a human analyst to understand the meanings or roles of the frequently visited places.

In order to detect and interpret typical routes of the movement between the significant places, the analyst first applies a database query to extract sequences of position records between the stops, from which trajectories (time-referenced lines) are constructed. Then, clustering is applied with the use of specially devised similarity measures. The results are computationally generalized and summarized and displayed in the spatial context (Figure 8).

7.2 Multilevel Visualization of the Worldwide Air Transportation Network

The air transportation network has now become more dense and more complex at all geographical levels. Its dynamic no more rests on simple territorial logics. The challenge is to gain insightful understandings on how the routes carrying the densest traffic organize themselves and impact the organization of the network into sub-communities at lower levels. At the same time, subnetworks grow on their own logic, involving tourism, economy or territorial control, and influence or fight against each other. Because of the network size and complexity, its study can no more rely on traditional world map and requires novel visualization. A careful analysis of the network structural properties, requiring recent results on small world phenomenon, reveals its multilevel community structure.

The original network is organized into a top level network of communities (Figure 9(a)). Each component can then be further decomposed into sub-

Chapter 3

Service Description Modeling Approaches and Semantic Analysis Process Models

The related work to manual and semi-automatic modeling of service descriptions and semantic analysis process models is described in this chapter. It gives an overview of challenges faced during the modeling of service descriptions and manual and semi-automatic approaches for the service description modeling. Various semantic analysis process models are also discussed here in detail, which provides an overview of most prominent semantic analysis process models. The related work for the service description modeling and semantic analysis process models serves as state of the art for this thesis. It provides also a solid scientific base to define the process model for the Visual Semantic Analysis approach that is discussed in the next chapter.

3.1 Manual Modeling of Service Description

The manual modeling of service descriptions is challenging task. It covers different aspects regarding the content level, e.g. legal information comprising general terms and conditions or technical information contained in the WSDL description. Different user roles provide and use different aspects for describing service. Legal experts provide the general terms and conditions and business experts provide the pricing information. Different kinds of information are modeled and used in different phases of the service lifecycle as described in Chapter 2. General terms and conditions, and pricing information might exist as soon as the service is offered, and the broker might add special kinds of quality of parameters later on. The users' feedback is usually available only after the service usage. The service description related information might be spread throughout the Internet of Services (IoS). The Quality of service parameters may reside centrally on a platform or users´ feedback may reside on a consumers' community portal. Different kinds of service description related information might be available in structured and unstructured formats. The variety of tools might be available to capture service description related information for different aspects and different roles. The following eight challenges are identified by several authors [BDMDG08, PMLZ08, BR08, HV07, Nad04] for the manual modeling of service description:

- Modeling
- Documentation
- Interlinkage
- Interoperability
- Querying
- Compliance
- Inconsistency
- Cooperation

The service description have to be *modeled* in a holistic and consistent way and enable traceability of the modeling decision. The modeling decisions have to be grounded in scientific literature of the corresponding domain by domain experts. For example, the modeling and business experts are needed to model pricing schemes for services and business experts can just justify how a pricing scheme must look like.

The *documentation* of the modeled information without media break is also an important issue. This includes formal knowledge representation, natural language description and UI labels in multiple languages.

Because of the different roles, phases, locations and structures, the modeled information have to be *interlinked* with each other. For example, users' feedback on the community portal should be linked to service offering information on the service broker's site.

Different user roles use heterogeneous tools along the service lifecycle to model different aspects of service description, which raise the *interoperability* issues. Obtaining information should be available e.g., "give me all services which are cheaper than 5 Euro, and have at least an average of a 3 star rating".

The modeled service description have to be *compliance* with (international) specifications or policies, standard or laws.

Different users provide the same kind of information independently and differently during the modeling of different aspects of service description. A sensible information management is needed to avoid *inconsistency*. The information has to be modularized to allow different user roles to maintain and contribute information corresponding to different aspects in different phases of the service lifecycle.

Although the pricing model and the technical description belong to same service, business experts are not interested in technical description. The support for *cooperative* modeling is one of the most import issues to allow different user roles to model different aspects of service in the different phases of the service lifecycle.

There are many approaches for manual modeling of service descriptions. These approaches are analyzed according to following five criteria:

1. **Black box or glass box view**: The black box view based efforts focus on the aspects related to data and control flow considering mainly Web services. The glass box view based approaches describes the internal structure such as service level agreements and non-functional attributes.

2. **Method**: Approaches present just a model or the corresponding method as well. Most approaches present just model without describing method

3. **Interdisciplinary**: interdisciplinary means, if the service description covers single domain or experts from different domains e.g. legal and price plan experts are involved in the modeling.

4. **Best practices**: The criterion best practices indicates the usage of foundation ontology as sound modeling starting point and the usage of ontology patterns to avoid arbitrariness in modeling [GGM+02]. In fact, best practice is not often applied in existing approaches.

5. **W3C Semantic Web**: The last criterion reflects whether W3C Semantic Web recommendations are used for the modeling of service description.

Manual service description modeling approaches are grouped in five categories.

The first category represents approached in the field of semantic Web services. The most prominent efforts OWL-S [AA01] and WSMO [RBM+06] are listed in the table, but also other approaches are available in literature.

According to the black box view, the purpose of these approaches is to automate tasks such as discovery and composition.

The second category shows other service-related ontologies outside the field of semantic Web dervices. The OASIS reference ontology for Semantic Service Oriented Architectures [DZ07] is however not built on best practices and in a very early stage. Similarly, The Open Group ontology for Service-Oriented Architectures (SOA Ontology) [Har10] covers both aspects business and technical perspectives. The approach presented by Ferrario et al. [FG08] is based on basic ontological analysis. This effort captures an ontologically sound glass box view on a service. Oberle et al. [OLG^{+}06] have introduced a black box ontology "Core Ontology of Web Services" based on best practices and Semantic Web standards. The OBELIX ontology [ABG^{+}04] focuses on the ecosystem and value chain relationships between services and it is based on best practices. The approaches [DKG08a, DKG08b] introduce the e^{3}Service ontology to model services from the perspective of the user needs. The main focus of this approach is to generate service bundles under the consideration of user needs. The Service ontology proposed by Oberle et al. [OBB^{+}09] follows a glass box view of services with their internal structure, service level agreements and non-functional attributes. It proposes also a Semantic Business Web approach as method and service governance framework. The Service Ontology is constructed by an interdisciplinary team of experts from different domain such as legal, rating, business model etc. It is build by applying best practices and W3C semantic Web recommendations.

UML-based efforts aim to support model-driven software engineering for services. Bitsaki et. al. [BDH^{+}08] present the Service Network Notion (SNN), which covers similar aspects as the e^{3}Service ontology. The UML Profile and Metamodel for Services (UPMS) [Ber08] is an OMG (Object Management Group) to support top down and bottom up modeling and utilizes UML collaboration diagrams. It is also linked with the business modeling framework with business process modeling (BPMN). The Service-oriented architecture Modeling Language (SoaML) [Ber08] for UPMS describes a UML profile and metamodel for the design of services in SOA. The main goal of SoaML is to support activities of service modeling and model-driven development approach [Ber08]. The survey UML-based Modeling of Web Service Composition (WSC) presents different approaches, e.g. structure-based WSC Modeling, behavior-based WSC Modeling and a hybrid WSC modeling. A UML based approach presented by Emmerich [Emm05] covers product related services such as maintenance without considering model-driven software engineering.

Service Modeling Language (SML) [PPS10] is the most prominent XML based effort. SML offers support to build a rich set of constructs for creating and constructing models of complex IT services and systems. An SML-Model consists of interrelated XML documents containing information

Approach	Glass Box	Method	Inter-disci-plinary	Best practice	W3C Semantic Web
1. Semantic Web Services					
OWL-S [AA01]	−	−	−	−	+
WSMO [RBM+06]	−	−	−	−	−
...					
2. Other service-related ontology					
OASIS Reference Ontology [DZ07]	−	−	−	−	+
The Open Group SOA Ontology [Har10]	+	−	−	−	+
Ferrario et al. [FG08]	+	−	−	+	+
Core Ontology of Web Services [OLG+06]	−	−	−	+	+
OBELIX ontology [ABG+04]	+	+	−	−	+
e3Service ontology [DKG08a, DKG08b]	+	+	−	−	+
Service Ontology [OBB+09]	+	+	+	+	+
3. UML-based Approaches					
Service Network Notation (SNN) [BDH+08]	+	+	−	N/A	N/A
UPMS [Ber08]	+	+	−	N/A	N/A
SOAML [Ber08]	+	+	−	N/A	N/A
Emmrich [Emm05]	+	−	−	N/A	N/A
4. XML-based Approaches					
SML [PPS10]	+	−	−	N/A	N/A
USDL [CWV09]	+	−	−	N/A	N/A
ÓSullivan [Ò06]	+	−	−	N/A	N/A
WS-*	−	−	−	N/A	N/A
5. Informal Approaches					
Alter [Alt08]	+	−	−	N/A	N/A
DIN PAS 1018 [DIN02]	+	−	+	N/A	N/A
Baida et al. [BGA01]	+	−	−	N/A	N/A
+ = yes, − = No, N/A = Not available					

Table 3.1: Comparison of Service Description Approches

about parts of IT services and constraints. The SML model could contain information about parts of IT services e.g. configuration, deployment, monitoring and capacity planning etc. The Universal Service Description Language (USDL) [CWV09] covers technical, business and operation perspectives. This approach defines and formalizes the metamodel Meta Object Facility (MOF) for the USDL to support model-driven service engineering. O Sullivan [Ò06] analyzes a set of existing informal service descriptions to identify functional properties of a service. WS-* specifications such as WSDL or WS-BPEL consider only Web services and have black box view.

Furthermore, the informal efforts such the approaches by Alter's [Alt08], Baid et al. [BGA01] or the DIN PAS 1018 [DIN02] are listed in Table 3.1. They capture a glass box view from business perspective. For example, the German norm PAS1018 (Public Available Specification) describes elements like service, provider, category, location.

The Visual Semantic Analysis approach aims to support manual service description modeling with semi-automatic approaches. Therefore, challenges faced by the manual service description modeling and analysis of different service description modeling approaches play a key role to identify requirements for the Visual Semantic Analysis. The knowledge experts, who are facing the above discussed challenges during the SD modeling and dealing with different SD modeling approaches, provide the requirements for the Visual Semantic Analysis. The requirements from the perspective of SD modeling are discussed briefly in the next chapter.

3.2 Semi-Automatic Modeling of Service Description

Different approaches for the semantic description of software components have already been explored to supports aspects like program understanding [BMW93] or reusable software libraries. The acquisition of semantics has always been a critical issue: The survey of software reuse libraries [MMM98] concludes that the use of ad-hoc and low tech methods is more practical than semantically sophisticated methods. Several approaches to acquire semantics automatically were presented: The GURU tool introduced by [MBK91] uses software documentation as knowledge source and uses a hierarchical agglomerative cluster method to extract an hierarchy of terms contained in the software documentation. Helm et al. [HM91] have combined this approach with code analysis methods. Milli et al. [MAKGM97] have introduced an Information Retrieval (IR) based approach, that uses noun phrases in the corpus and extracts semantics by using co-occurrence of terms in the software documentation.

A subtopic of acquisition of semantics for software is the acquisition of semantics for services, which is a time consuming and complex task. The automation of the semantic acquisition task is desirable [SP07], [Sab06], [SWGS05] and [WGG$^+$04]. Four research teams have introduced semi-automatic semantic acquisition techniques for services. Hess and Kushmerick [HK04, HK03, HJK04] employ Naïve Bayes and SVM machine learning methods to WSDL files. They have also introduced a tool that allows users to assign classes and properties to operations, parameters and complex types. The machine learning algorithm suggests a probable classification to users to annotate services.

Patil et al. [POSV04] have introduced an interesting semi-automatic semantic acquisition technique that allows matching of WSDL files with relevant ontologies. The conversion of WSDL and ontology formats, e.g. RDF(S), to a common representation format SchemaGraph to facilitate better matching. The common representation format SchemaGraph allows the usage of different ontology formats DAML, RDF-S or OWL. The mapping between WSDL concepts and ontology was achieved by applying match score techniques. The both techniques by Hess and Kushmerick [HK04, HK03, HJK04] and by Patil et al. [POSV04] use existing domain ontologies.

Sabou et al. [SP07, Sab06, SWGS05] have presented a semi-automatic semantic acquisition of service. They used textual description of services as knowledge source and deployed pattern based extraction rules (lexical analysis) for the semantic acquisition. They observed that noun phrases in the textual description of services denote the parameters of services and the verbs indicate the functionality of services. For example, the noun phrases "image" and "url address" and the verbs "extract" in the following textual description of the a service "Extract images from a given url address" denote parameters and the functionality of the service correspondingly. . Sabou has also presented the visualization technique "Cluster Map" to support semantic acquisition for services as shown in Figure 3.1 . She purposes investigation of other visual techniques, interlinking and integrating the presented visualization techniques in semantic acquisition techniques for the services [Sab05]. Additionally, she has also introduced a generic ontology to model service description.

Wei et al. [WWWC08] have deployed heuristic rules to extract semantic constraint from description text of services. The constraint types were defined to extract semantic constraint from the text description of services. For example, a constraint type $\langle A, isPropertyObjectof, B\rangle$ means concept A is a property object of concept B. According to this constraint type the constraint $\langle Price, isPropertyObjectOf, Book\rangle$ can be extracted from the text description of a service " The service returns the price of the book published by Springer". Furthermore, they have used Constraint Graph Matching [WWWC08] for the service discovery.

The above described semi-automatic SD modeling approaches are not prothrough a set of classes (concepts) and their hierarchical relation. The technique visualizes *instances* of a set of *classes* according to their classification into these classes. Due to the specialization relationship that is encoded in the hierarchy, the set of objects in a subclass is a subset of its superclass. The set of subclasses of a class is *incomplete* when their union does not contain all the objects of the superclass. Classes that share instances are *overlapping* if no specialization relationship holds between them. These characteristics are quit common for taxonomies. However, they are difficult to show satisfactorily in textual representations or in schema like techniques that are currently used in the Semantic Web. The Cluster Map offers an alternative in this matter.

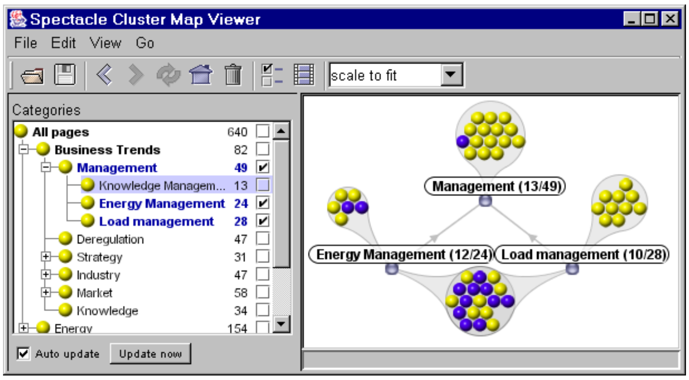


Figure 7.8: Cluster Map example.

Figure 3.1: Cluster Map [Sab05] visualization technique to support semi-automatic modeling of semantic web description

cess based approaches. They allow extraction of semantics from the WSDL-files and textual descriptions of services by using semantic analysis techniques. The usage of semantic analysis and interactive visualization techniques to analyze available services, model new services and reanalyze newmodeled services with available services is not the focus of these approaches. The validation, refinement, and merging of extracted semantics, and usage of interactive visualization techniques for the semantic analysis are also not discussed in these approaches, but these approaches provide an overview of initial approaches towards semi-automatic modeling of SSD by using semantic analysis techniques.

3.3 Semantic Analysis Process Models

The most prominent process models based on semantic analysis approaches are discussed briefly in this section. These semantic analysis approaches do not deal directly with the semi-automatic modeling of service description, but they provide a theoretic base to derive a process model for the Visual Semantic Analysis to support semi-automatic modeling of service descriptions. The usage of information visualization techniques for these process model based semantic analysis approaches will also be discussed briefly.

Maedche et al. [Mae02, MS01] have presented an ontology learning process model (or semantic analysis process model) with four steps: import / reuse, extraction, prune, and refine, as depicted in Figure 3.2. First, the available knowledge sources with their (schema) structures are imported and merged together by defining mapping rules. Second, semantic is extracted by using

Figure 7.8 shows an example Cluster Map that visualizes a set of documents classified according to topics discussed in those documents. Each small sphere represents an instance. The classes are represented as rounded rectangles, stating their names and cardinalities. Directed edges connect classes and point from specific to generic (e.g., *Load Management* is a subclass of *Management*). Balloon-shaped edges connect instances to their most specific class(es). Instances with the same class membership are grouped in clusters (similar to Venn Diagrams). Our example contains four clusters; one of them represents the overlap of the *Load management* and *Energy Management* classes.

The organization of the graph is computed using a variant of the well-known spring-embedder algorithm (Eades, 1984). On the one hand the class and cluster nodes repel each other. On the other hand the edges, connecting two classes or clusters to their classes, produce an attractive force between the connected nodes; i.e., they work as "springs".

The added value of our visualization lies in its expressivity. The classes and their relationships (the vocabulary of the domain) are easy to detect. Also, it is immediately apparent which items belong to one or multiple classes, which classes overlap (e.g., *EnergyManagement* and *LoadManagement*) and which do not. The subclasses of the root class are *incomplete* as their union does not cover the superclass: some members of *Management* were not further classified.

Another interesting aspect of the visualization is that geometric closeness in the map is related to semantic closeness. This is a consequence of the graph layout algorithm.

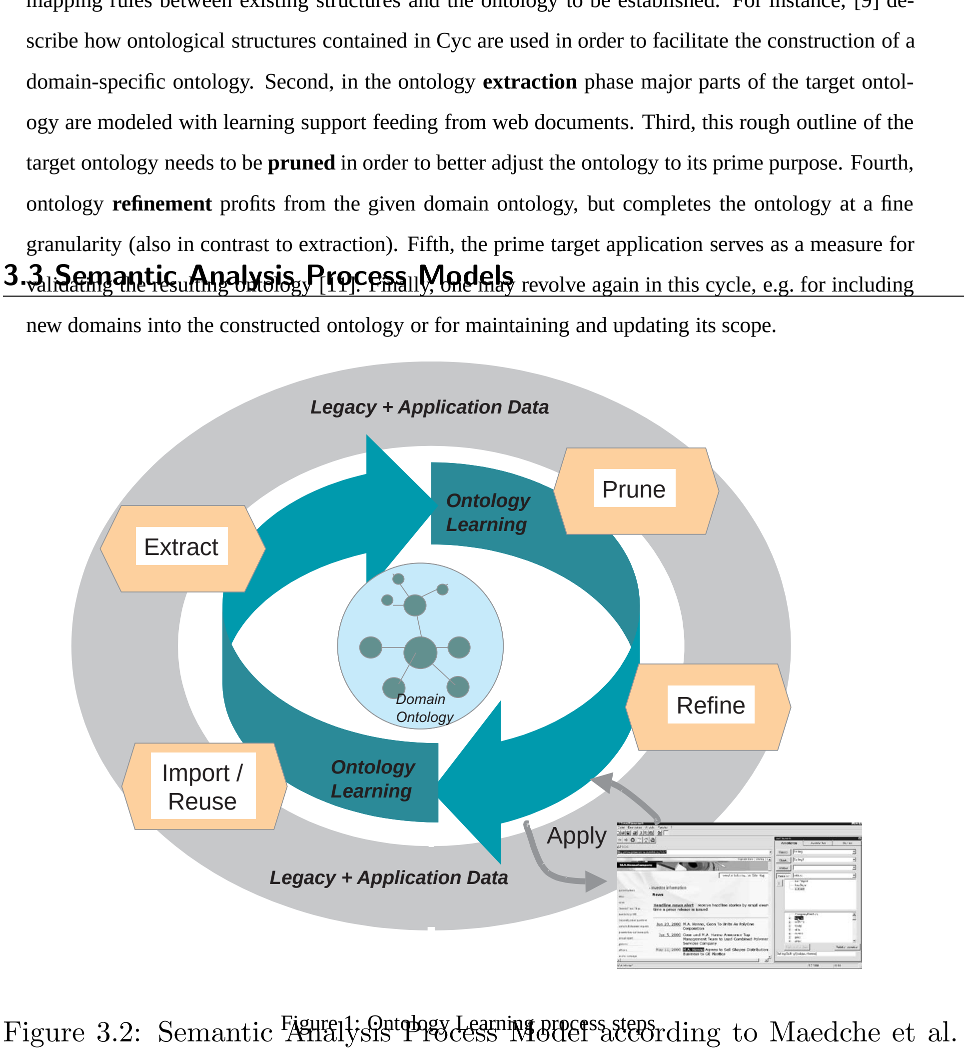
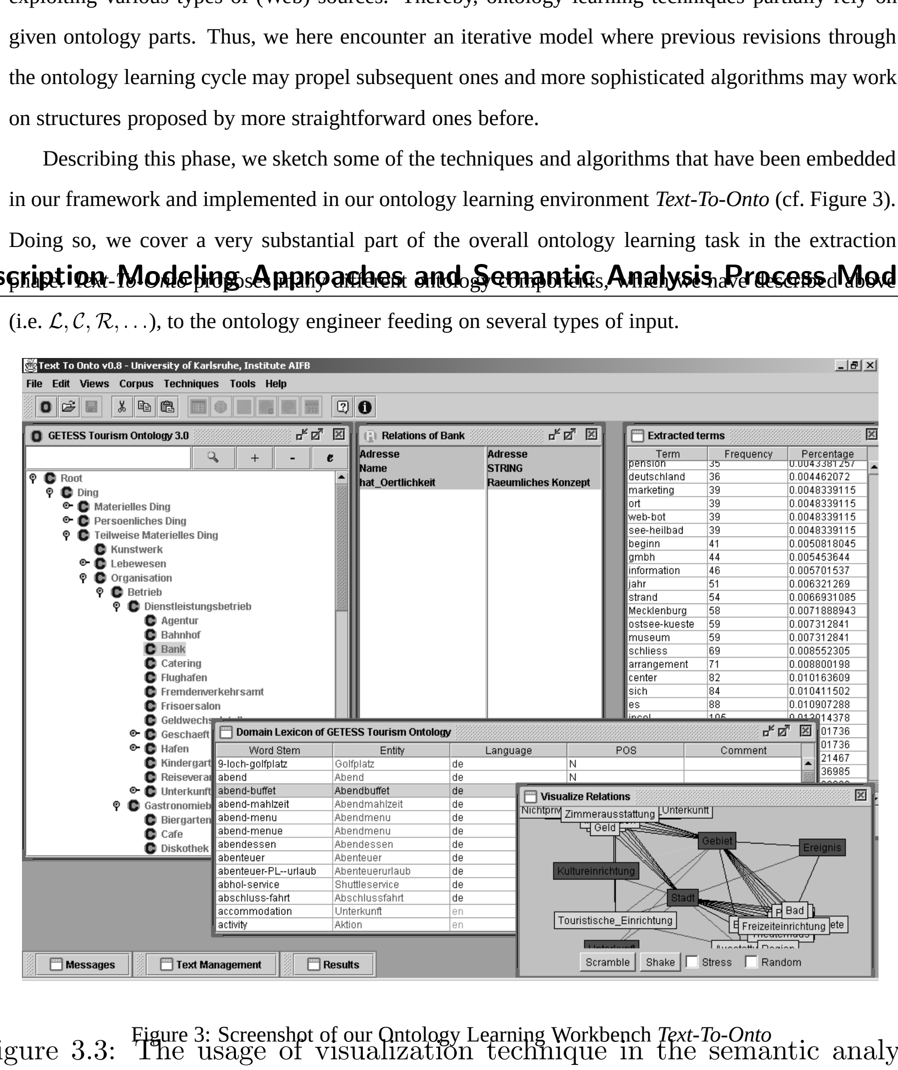

mapping rules between existing structures and the ontology to be established. For instance, [9] describe how ontological structures contained in Cyc are used in order to facilitate the construction of a domain-specific ontology. Second, in the ontology **extraction** phase major parts of the target ontology are modeled with learning support feeding from web documents. Third, this rough outline of the target ontology needs to be **pruned** in order to better adjust the ontology to its prime purpose. Fourth, ontology **refinement** profits from the given domain ontology, but completes the ontology at a fine granularity (also in contrast to extraction). Fifth, the prime target application serves as a measure for validating the resulting ontology [11]. Finally, one may revolve again in this cycle, e.g. for including new domains into the constructed ontology or for maintaining and updating its scope.

Figure 1: Ontology Learning process steps.

Figure 3.2: Semantic Analysis Process Model according to Maedche et al. [Mae02, MS01]

different learning techniques e.g., lexical entry & concept extraction, clustering and, association rules. Third, the extracted semantic is pruned to adjust extracted semantic to its prime purpose. Fourth, the refinement completes pruned semantic at fine granularity. Finally, more iterations of the process can be performed to include new domains into extracted and refined semantics and maintain its scope. The information visualization technique, e.g. graph visualization, is used to visualize the results of semantic analysis as shown in Figure 3.3, but does not allow interactive analysis or refinement.

Cimiano et al. [CHST04] used existing ontologies and extracted instances from a given set of domain specific text documents by applying Natural Language Processing (NLP) techniques, as discussed in chapter 2. The concepts of the ontologies are assigned to text documents. In this way, a number of type of context information can be derived as depicted in Figure 3.4. The documents and concepts of different ontologies define these different context information types. The Formal Concept Analysis (FCA) , as discussed in chapter 2, merge techniques allow semantic extraction from domain specific text documents by using different existing ontologies. Finally, the semantic can be generated from extracted semantic information. Cimiano et al. [CHS05] have also introduced another process model as shown in Figure 3.5 to extract semantics from the text without using existing ontologies. This process model describes NLP process steps in detail. Both process model


exploiting various types of (Web) sources. Thereby, ontology learning techniques partially rely on given ontology parts. Thus, we here encounter an iterative model where previous revisions through the ontology learning cycle may propel subsequent ones and more sophisticated algorithms may work on structures proposed by more straightforward ones before.

Describing this phase, we sketch some of the techniques and algorithms that have been embedded in our framework and implemented in our ontology learning environment *Text-To-Onto* (cf. Figure 3). Doing so, we cover a very substantial part of the overall ontology learning task in the extraction phase. *Text-To-Onto* proposes many different ontology components, which we have described above (i.e. $\mathcal{L}, \mathcal{C}, \mathcal{R}, \ldots$), to the ontology engineer feeding on several types of input.

Figure 3: Screenshot of our Ontology Learning Workbench *Text-To-Onto*

Figure 3.3: The usage of visualization technique in the semantic analysis approach of Maedche et al. [Mae02, MS01]

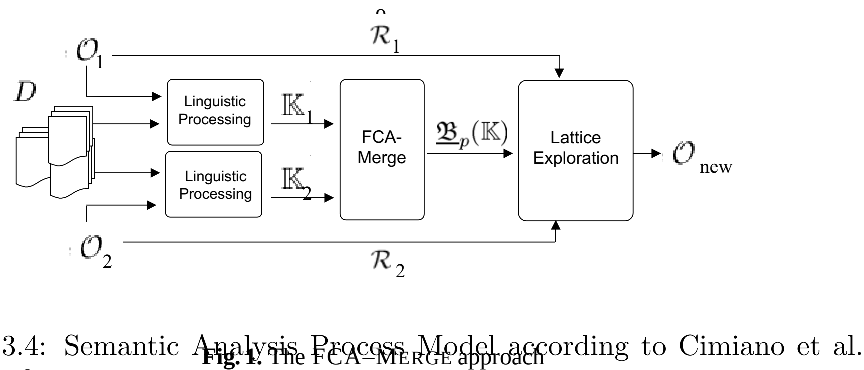
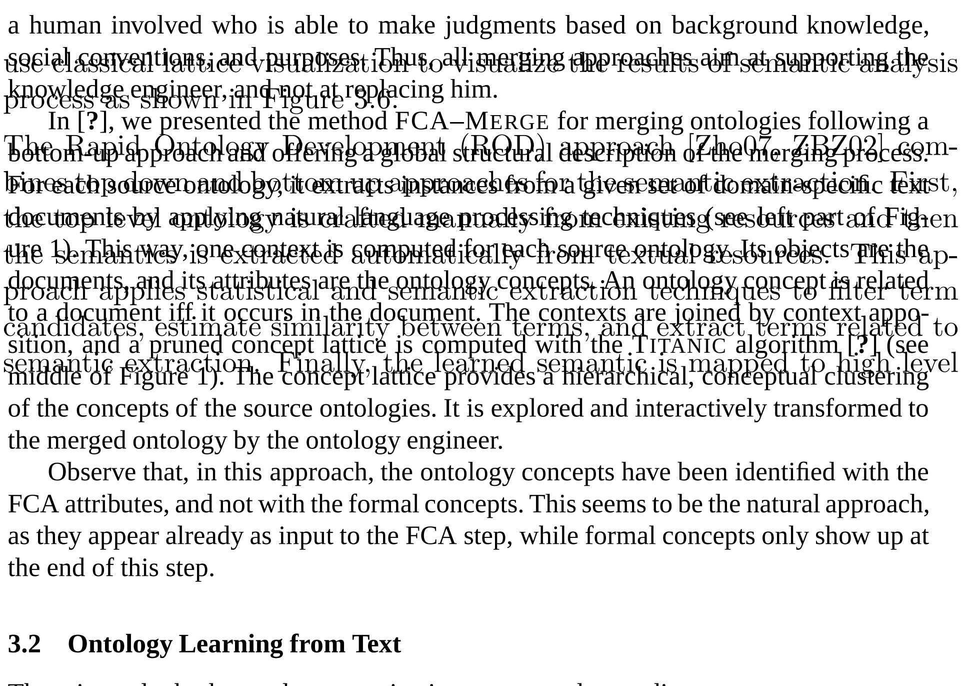


Fig. 1. The FCA–MERGE approach

Figure 3.4: Semantic Analysis Process Model according to Cimiano et al. [CHST04]

use classical lattice visualization to visualize the results of semantic analysis process as shown in Figure 3.6.

The Rapid Ontology Development (ROD) approach [Zho07, ZBZ02] combines top-down and bottom-up approaches for the semantic extraction. First, the top level ontology is created manually from existing resources and then the semantics is extracted automatically from textual resources. This approach applies statistical and semantic extraction techniques to filter term candidates, estimate similarity between terms, and extract terms related to semantic extraction. Finally, the learned semantic is mapped to high level


a human involved who is able to make judgments based on background knowledge, social conventions, and purposes. Thus, all merging approaches aim at supporting the knowledge engineer, and not at replacing him.

In [**?**], we presented the method FCA–MERGE for merging ontologies following a bottom-up approach and offering a global structural description of the merging process. For each source ontology, it extracts instances from a given set of domain-specific text documents by applying natural language processing techniques (see left part of Figure 1). This way, one context is computed for each source ontology. Its objects are the documents, and its attributes are the ontology concepts. An ontology concept is related to a document iff it occurs in the document. The contexts are joined by context apposition, and a pruned concept lattice is computed with the TITANIC algorithm [**?**] (see middle of Figure 1). The concept lattice provides a hierarchical, conceptual clustering of the concepts of the source ontologies. It is explored and interactively transformed to the merged ontology by the ontology engineer.

Observe that, in this approach, the ontology concepts have been identified with the FCA attributes, and not with the formal concepts. This seems to be the natural approach, as they appear already as input to the FCA step, while formal concepts only show up at the end of this step.

3.2 Ontology Learning from Text

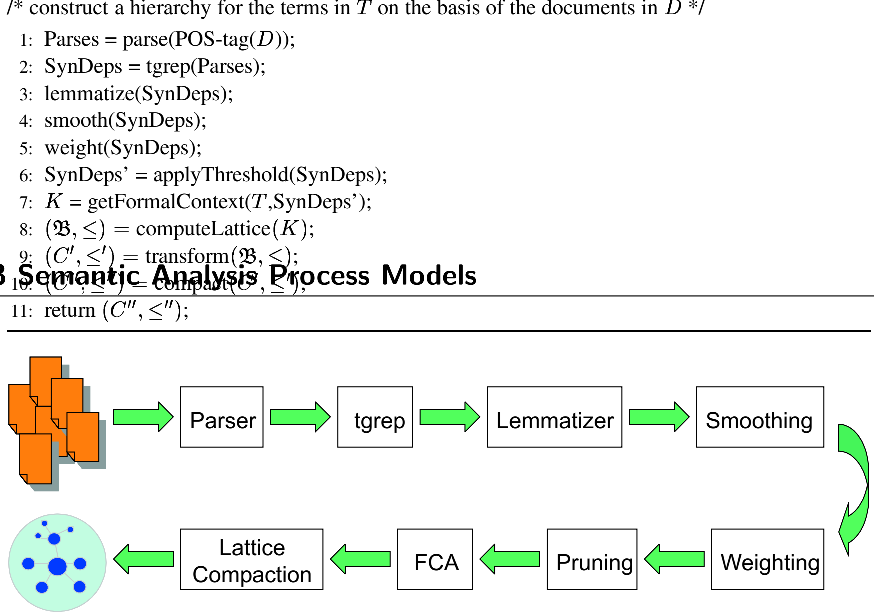


Figure 3.5: Semantic Analysis Process Model according to [CHS05]

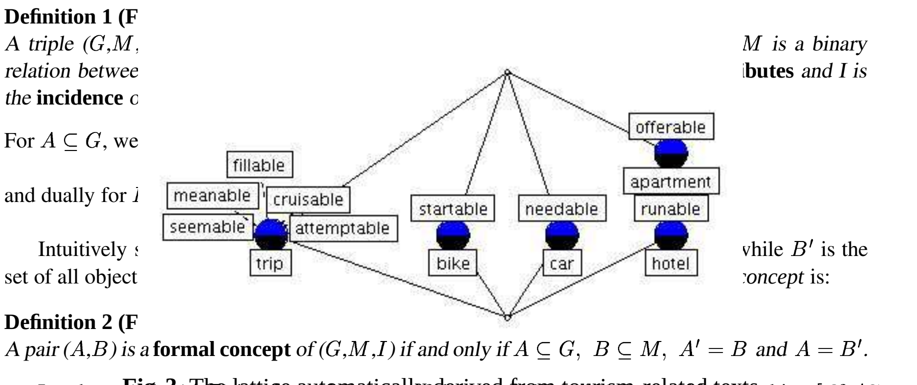


Figure 3.6: The lattice visualization used by Cimiano et al. [CHS05, CHST04]

semantically as shown in Figure 3.7.

The PACTOLE methodology [BTN08] allows semi-automatic semantic acquisition from astronomy related texts as depicted in Figure 3.8. The process model for the PACTOLE methodology is shown in Figure 3.8. The first step is Natural Language Processing (NLP) to extract pairs (object, property). In the second step, Formal Concept Analysis (FCA) is used to extract an hierarchy of concepts from the pairs (object, property). The third step converts the existing knowledge sources like dictionary/data base into a concept hierarchy by using FCA. In the fourth step, both semantic models (from the text and from existing knowledge sources) are merged together and enriched. During the fifth step, the semantic is generated from lattice and formalized. This semantic analysis method uses also lattice visualization to visualize the extracted semantic.

Pan et al. [PF09a] have introduced a semantic analysis process model for radiology reports as shown in Figure 3.9. This approach was developed for the Picture Archiving and Communication System (PACS) to improve image diagnosis accuracy and reducing patients' radiation exposure. The process model is shown in Figure 3.9. In the first step, radiologists create a radiology report text during diagnosis of medical images. After the term parsing

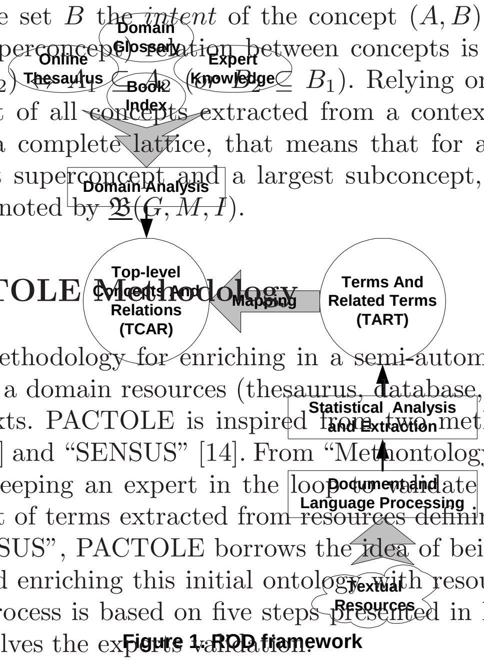


Figure 3.7: Semantic Analysis Process Model according to ROD Methodology [Zho07, ZBZ02]

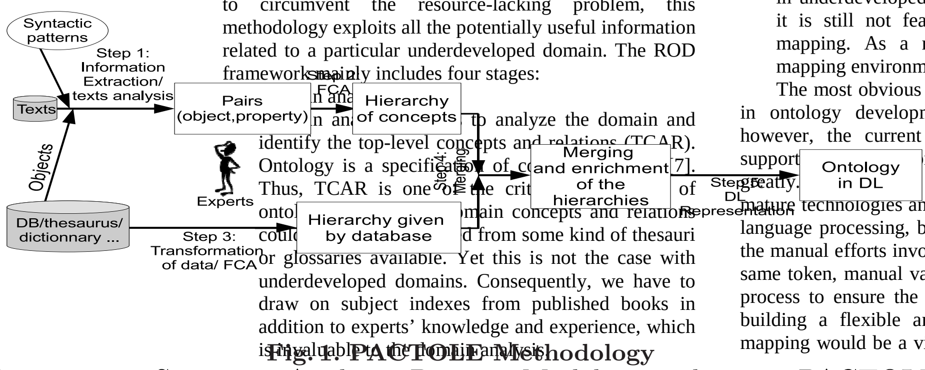
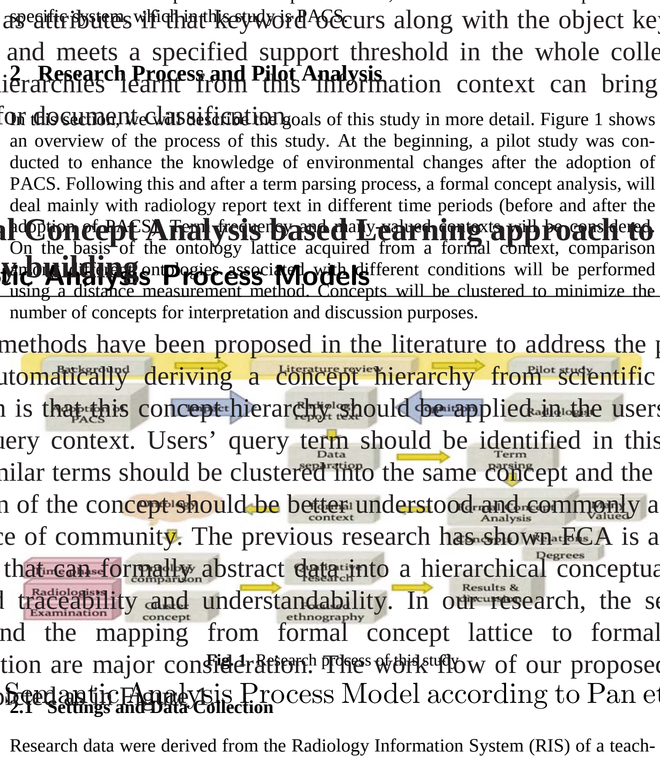


Figure 3.8: Semantic Analysis Process Model according to PACTOLE Methodology [BTN08]

process, FCA was applied to extract semantic from the radiology report text. Finally, the extracted semantic is compared with related ontologies by using a distance measurement method. The lattice visualization is used to visualize the extracted semantic.

Jia et al. [JNT09] have introduced a process model as depicted in Figure 3.10 to extract semantic semi-automatically from domain specific scientific documents. In the first step, keyword extraction from the research corpus takes place. Keyword which were semantically similar, are clustered together. After the statistical filter and NLP processing, formal context is generated.



specific systems, which in this study is PACS.

2 Research Process and Pilot Analysis

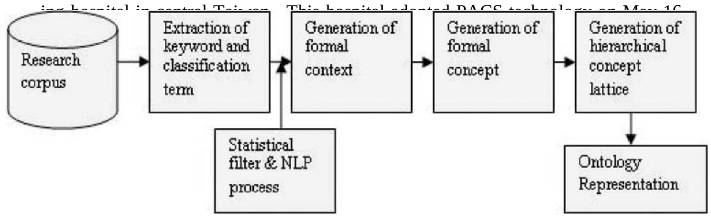
In this section, we will describe the goals of this study in more detail. Figure 1 shows an overview of the process of this study. At the beginning, a pilot study was conducted to enhance the knowledge of environmental changes after the adoption of PACS. Following this and after a term parsing process, a formal concept analysis, will deal mainly with radiology report text in different time periods (before and after the adoption of PACS). Term frequency and daily-valued contexts will be considered. On the basis of the ontology lattice acquired from a formal context, comparison among different ontologies associated with different conditions will be performed using a distance measurement method. Concepts will be clustered to minimize the number of concepts for interpretation and discussion purposes.

Fig. 1. Research process of this study

2.1 Settings and Data Collection

Research data were derived from the Radiology Information System (RIS) of a teaching hospital in central Taiwan.

Figure 3.9: Semantic Analysis Process Model according to Pan et al. [PF09a]



keywords as attributes if that keyword occurs along with the object keyword in a document and meets a specified support threshold in the whole collection. The concept hierarchies learnt from this information context can bring improved precision for document classification.

3 Formal Concept Analysis based Learning approach to Ontology building

Different methods have been proposed in the literature to address the problem of (semi-) automatically deriving a concept hierarchy from scientific text. Our motivation is that this concept hierarchy should be applied in the users' query to expand query context. Users' query term should be identified in this structure; topical similar terms should be clustered into the same concept and the intentional description of the concept should be better understood and commonly accepted by the practice of community. The previous research has shown FCA is an effective technique that can formally abstract data into a hierarchical conceptual structure with good traceability and understandability. In our research, the selection of context and the mapping from formal concept lattice to formal ontology representation are major consideration. The work flow of our proposed approach can be depicted in Figure 1.

Fig.1. Ontology Learning Work Flow

3.1 Formal Concept Analysis

In order to better interpret our approach, we briefly recall some basic terminology and definition of FCA and further detail can be found in [12].

Definition 1. A Formal Context is a triple *(G, M, I)*, where G is a set of objects, M is a set of attributes, and I is a binary relation between G and M (i.e. $I \subseteq G \times M$). $(g, m) \in I$ can be read as object g has attribute m.

Figure 3.10: Semantic Analysis Process Model according to Jia et al. [JNT09]

Then the FCA is applied to extract the lattice structure. Finally, the semantics can be extracted from the lattice structure. The approach from Haav et al. [HHN00] used a similar approach to extract semantic from domain specific text. This approach introduces additionally rules and facts to enrich the semi-automatically extracted semantics. The lattice visualization was also used by Jia et al. [JNT09] to visualize extracted semantics.

As already discussed in Section 2.2.3, visual techniques take into account human perceptual capabilities to detect patterns and outliers in visual information. They can facilitate experts' understanding about complex data, structures, hierarchy and relations in semantic descriptions [Sab05]. The visual analytics approaches [LKS+10, KAF+08] propose also a combination of automated data analysis and interactive visualizations to support effective understanding, reasoning and decision making on the base of very large and complex data sets. The process models which deals with visualization techniques for data analysis are described in the following paragraphs to consider visualization techniques for the semantic analysis as it is also proposed by Sabou [Sab05].

Card et al. [CMS99] have introduced a simple reference model for the information visualization to map raw data to visual forms as shown in Figure 3.11. The arrow flow from raw data to the human indicates a series of data transformations. The arrow flow from the human into transformations rep-

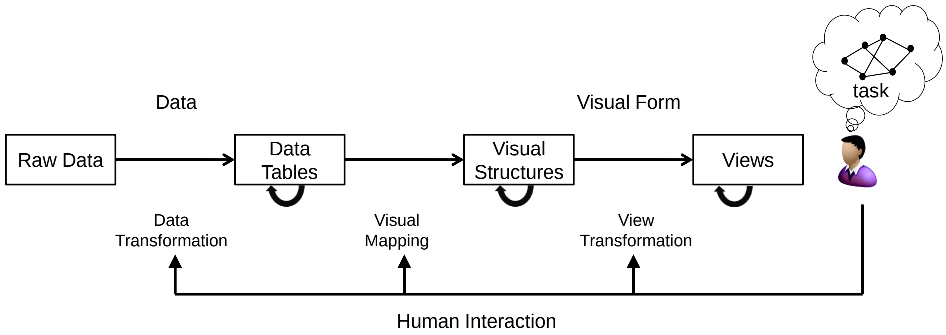


Figure 3.11: Reference Model for information visualization according to Card et al. [CMS99]

resents the adjustment of the transformation by user interaction. The user can transform the raw data to accomplish a desired task. Raw data can consist of various forms, from spreadsheets to text. The usual strategy is to map the raw data into a relation or set of relations that are more structured. The data transformation facilitates the transformation of various forms of raw data to more structured forms, like data tables. The visual mapping of data tables into visual structures, structures that combine spatial substrates (such as x- and y-coordinates), marks (points, lines and areas), and graphical properties (color, size and texture)[Mac86]. Finally, the view transformation generates views by specifying graphical parameters of visual structures such as position, scaling and clipping. User can control the parameters of the transformations e.g., restricting the view to certain data ranges. The information visualization, user interaction, and transformation facilitate users to perform a desire task.

May et al. [MKJ08] have presented the process model shown in Figure 3.12 to combine statistical data classification and interactive visualization techniques in order to group data items into categories. This approach extends the information visualization reference process [CMS99] as discussed above and establishes an iterative cyclic process. A single cycle starts with the selection of data, proceeds with the update of the classifier, and ends with the visual feedback of the new categories by using the KV-Map as shown in Figure 3.12. The user may continue the iterative refinement process by modifying the selection and analyzing the visual feedback.

Keim et al. [KAF$^+$08] have introduced a visual analytics process that is based on the simple model of visualization by van Wijk [Wij05], shown in Figure 3.13. According to Keim et al. [KAF$^+$08], this process model has to be evaluated and measured in terms of efficiency of knowledge gained. After initial data analysis by using statistical and mathematical techniques (e.g.

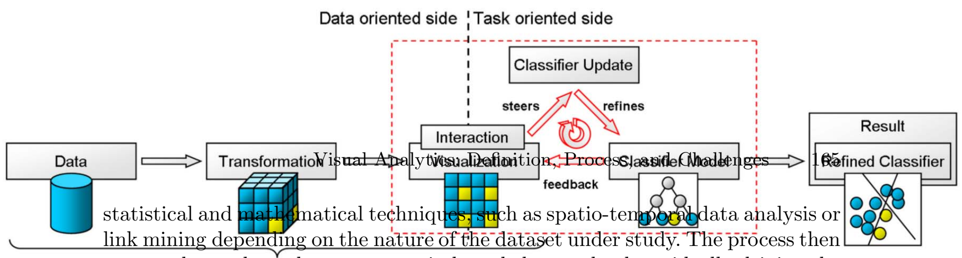


Figure 3.12: Semantic Analysis Process Model according May et al. [MKJ08]

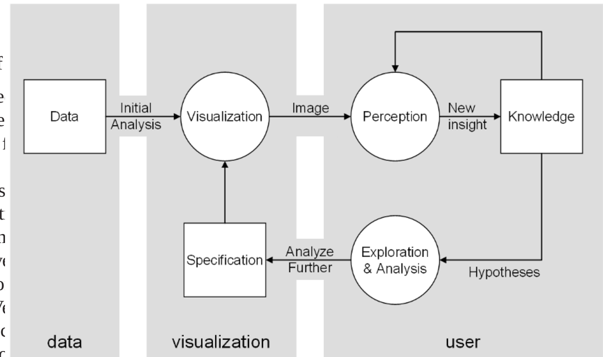


Figure 3.13: The Visual Analytics Process according to Keim et al. [KAF+08]

spatio-temporal data analysis or link mining depending on the dataset), the initial data analysis results can be visualized with appropriate visualization and interaction techniques. After initial analysis, users can enter into a loop to gain knowledge on the data and drive the system towards more focused analysis and adequate analytical techniques. The interaction on the visual representation facilitates users to gain a better understanding of visualization itself. Different views help them to go beyond visual representation and confirm or reject the hypotheses of initial analysis iteration. They can keep analyzing data within this loop till they find answers to their question or task from the dataset understudy. Visualization techniques such as [CMS99], [MKJ08] and Visual Analytics mostly deal with interactive data analysis, but semantic analysis can also profit from these techniques to support interactive semi-automatic semantic analysis.

The most of above described semantic analysis approaches deal with semantic extraction, and validation, refinement, and merging of extracted semantics. The semantic analysis approaches except Visual Analytics (VA) based semantic analysis techniques don't use visualization techniques intensively to support the semantic analysis process. The VA based semantic analysis techniques allows the combination of semantic analysis and interactive visualization techniques to support the semantic analysis process, but they don't deal with validation, refinement and merging of extracted semantics.

None of these semantic analysis techniques deals with analysis of available services, modeling of new services and reanalysis of new-modeled services with available services. These semantic analysis process models provide a scientific base for the definition of a generic process model that allows the combination of semantic analysis and visualization techniques to analyze, model and reanalyze services.

3.4 Summary

The Visual Semantic Analysis approach aims to combine semantic analysis and visualization techniques to support semi-automatic modeling of SDs. Different manual SD modeling approaches are discussed in this chapter that plays a key role to identify the requirements for the VSA approach. The semi-automatic approaches for the SD modeling are also presented, which allow semantic extraction from the WSDL files or textual description of services. These approaches don't deal with analysis, modeling and reanalysis of non-functional parameters of services. Furthermore, semantic analysis approaches, which allow semantic extraction from structured and un-structured data, are discussed. VA based semantic analysis techniques show how semantic analysis and visualization techniques can support the semantic analysis process. The focus of these semantic analysis approaches is not the SD modeling, but they provide a base for the analysis of available services. The VSA approach considers the process models of these semantic analysis approaches as scientific base to define a generic process model of the VSA that allows analysis, modeling and reanalysis of SDs.

Different manual SD modeling, semi-automatic SD modeling, and semantic analysis approaches described above serve as related work and help to identify requirements for the VSA approach. The knowledge experts provide requirements form the perspectives of users, semantic analysis, and SD modeling. The requirements analysis for the VSA approach and comparison of these approaches with the VSA approach are presented in the next chapter. Furthermore, a generic process model of the VSA approach ant its formal specification are defined under the consideration of the requirements analysis in the next chapter.

Chapter 4

A Process Model for the Visual Semantic Analysis

In the service ecosystem, service providers can re-combine and mediate their core services and deliver to service consumers by using available services of Web service ecosystem, e.g. service delivery and billing. One of the main challenges faced by Web service ecosystems is *flexible service discovery*. Service providers have to provide explicit and formalized description of non-functional parameters of services, e.g. guarantees, pricing, payment, penalties, and delivery modes, as Service Description (SD) to support *flexible service discovery* in Web service ecosystems. According to Barros and Dumas [BD06, BDB05], the modeling of Service Description (SD) will become a bottleneck for Web service ecosystems. The manual modeling of the SDs is a tedious, cumbersome and time consuming task[OBB+09, WWWC08, SP07]. We present the Visual Semantic Analysis approach to support semi-automatic modeling of SDs, and thus also facilitate the establishment of service ecosystems.

A generic process model of the Visual Semantic Analysis (VSA) approach to support semi-automatic modeling of SD is described in this chapter. First, the definitions of VSA related terminologies are introduced and then requirements analysis is performed to define requirements for the VSA approach. Then the comparison of the VSA approach with related research efforts is presented that illustrates the positioning of VSA approach to the related work. Afterwards, a generic process model of VSA developed on the base of requirements analysis is discussed in detail. Finally, the formal specification of the generic process model of the VSA is introduced.

4.1 Definitions of terms

The definitions of VSA related terms are presented in this section that help to define a generic process model for the VSA precisely.

Service Descriptions

The Service Description (SD) contains textual service description, and functional and non-functional parameters, e.g. input, output, service annotation, price plans, legal aspects, Service Level Agreements (SLAs), security and user feedback [OBB+09, CWV09]. The textual service description and non-functional parameters such as service annotation, price plans, legal aspects and SLAs of SDs are considered red for the VSA approach, therefore the SD for the VSA is defined as follows.

Definition 4.1.1 (Service Description)**.** *The Service Description (SDs) contains textual service description and non-functional parameters such as, service annotation, price plans, legal aspects and SLAs. The textual service description serves as unstructured data and the non-functional parameters are considered as structured data for the VSA.*

The SDs of the available services is used as knowledge source in the VSA approach to analyze and extract semantic information from them. The annotations of the available services and extracted semantic information from the textual service descriptions of the available services support semi-automatic annotation of new services. The analysis of non-functional parameters (price plans, legal aspects and SLAs) facilitates the modeling of these non-functional parameters for new services. The SD modeling for the VSA is defined as follows:

Definition 4.1.2 (Service Description Modeling)**.** *The Service Description (SD) Modeling means to define service annotation, price plans, legal aspects and SLAs.*

The SD modeling is a complex and interdisciplinary task and the knowledge experts from different domains, e.g. service engineers, service modeling experts, price plans experts, legal experts and SLAs experts, takes part in the SD modeling [OBB+09, CWV09]. That is why, the knowledge experts for the VSA are defined as follows:

Definition 4.1.3 (Knowledge Experts)**.** *The term "knowledge experts" represents service engineers, service modeling experts, price plans experts, legal experts and SLAs experts in the VSA approach. They define service annotation, price plans, legal aspects and SLAs of the SDs.*

Semi-Automatic Service Annotation

In the most recent years, the research field annotation of information artifacts has become a promising research field to support sharing and reuse of information artifacts. The information artifacts range from multimedia data to services. The underlying idea is to have semantic description of information artifacts in order to support reuse and exchange of information artifacts across various applications or organizations. It brings enormous challenges and promises immediate practical benefits [DVGT10, Sab06].

The easy service discovery based on the functional and non-functional semantics is necessary to use the existing service. Different research efforts investigate possibilities to automate service tasks like discovery and composition by enriching services with semantic descriptions. The semantic analysis and visualization techniques are becoming very important to extract semantics from the textual service descriptions and annotate services with new facets [WWWC08, SP07].

It is difficult for knowledge experts to annotate services with the best-fit terms, because they rely on their own view of domain and knowledge to annotate services instead of investigating all available terms. The semantic analysis techniques can support them to identify most frequent terms used by community. The annotation of services with the best-fit terms can lead to improve service matchmaking and support service tasks, e.g. service discovery and reuse of existing services during the service composition. The annotation of services supported by semantic analysis and visualization techniques is denoted as semi-automatic service annotation and defined for the VSA as follows:

Definition 4.1.4 (Semi-Automatic Service Annotation)**.** *The semi-automatic service annotation during the SD modeling is defined as annotation of services with the best-fit terms extracted from unstructured data such as textual description of services by using a combination of semantic analysis and visualization techniques.*

Price Plans

The pricing plays a key role in the strategy of most companies. It determines long term turnover and profit of a company. It plays also very important role for the companies' reputation and customer relationship [Sim92, Rul08]. The erroneous decisions for the pricing model may affect companies' reputation and their customer relationship.

Nowadays, software is offered and obtained in various ways. Besides traditionally on-premise software, an increasing trend toward on-demand solution Software as a Service (SaaS) is establishing in IT-landscapes. Free and open source software is also available. In case of open source software, the turnover is generated by offering supplementary services, e.g. consulting, implementation and documentation [BHL08].

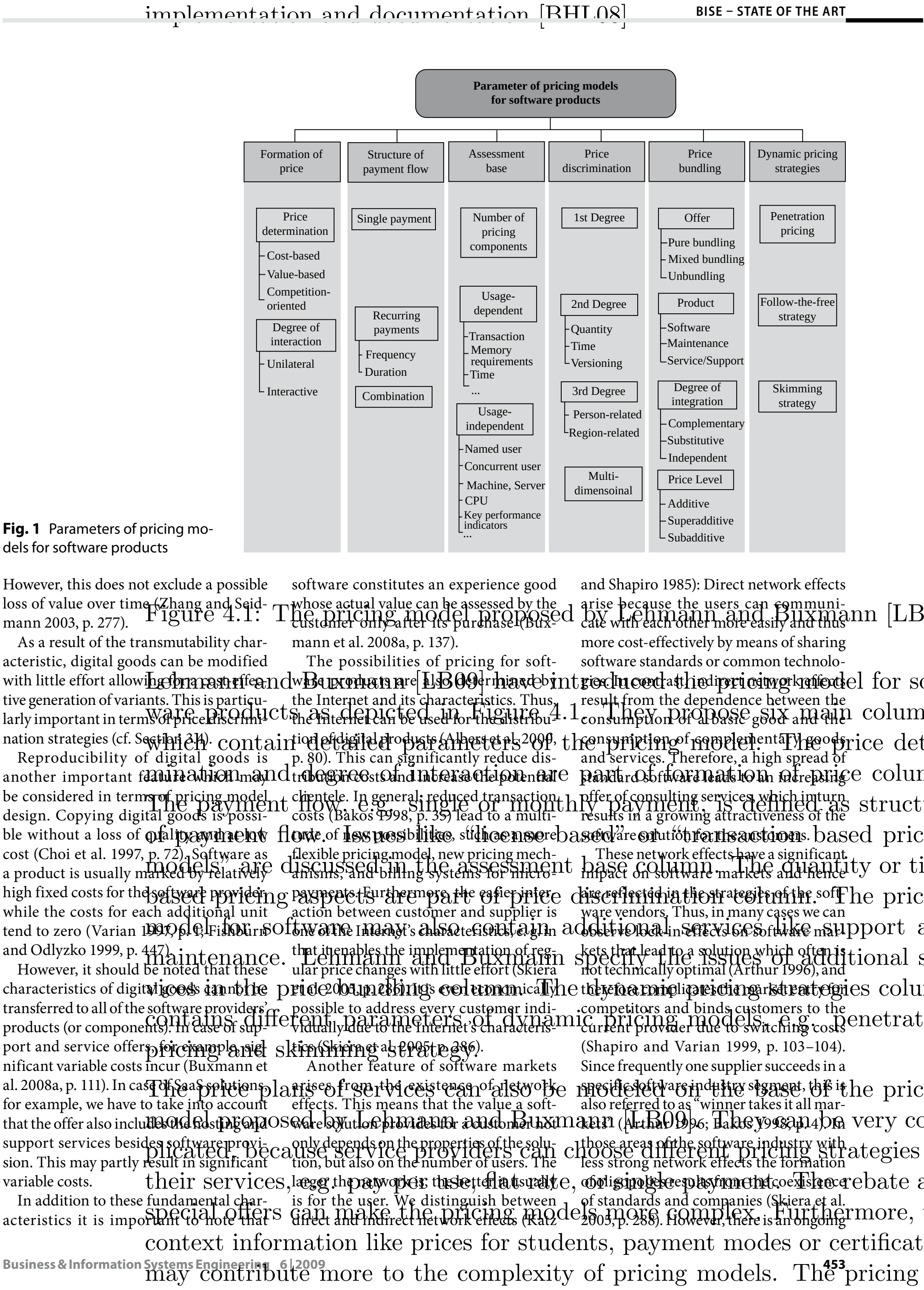


Figure 4.1: The pricing model proposed by Lehmann and Buxmann [LB09]

Lehmann and Buxmann [LB09] have introduced the pricing model for software products as depicted in Figure 4.1. They propose six main columns, which contain detailed parameters of the pricing model. The price determination and degree of interaction are part of formation of price column. The payment flow, e.g. single or monthly payment, is defined as structure of payment flow. Issues like "license based" or "transaction based pricing models" are discussed in the assessment base column. The quantity or time based pricing aspects are part of price discrimination column. The pricing model for software may also contain additional services like support and maintenance. Lehmann and Buxmann specify the issues of additional services in the price bundling column. The dynamic pricing strategies column contains different parameters of dynamic pricing models, e.g. penetration pricing and skimming strategy.

The price plans of services can also be modeled on the base of the pricing model proposed by Lehmann and Buxmann [LB09]. They can be very complicated, because service providers can choose different pricing strategies for their services, e.g. pay per use, flat rate, or single payment. The rebate and special offers can make the pricing models more complex. Furthermore, the context information like prices for students, payment modes or certification may contribute more to the complexity of pricing models. The pricing related aspects of SDs are denoted as price plans for the VSA and defined as follows.

Definition 4.1.5 (Price Plan)**.** *A price plan of SDs defines pricing components such as price, special offers and payment modes according to different pricing strategies, e.g. penetration pricing and skimming strategy.*

4.1.1 Legal Aspects

The service technology based distributed environments, where services can be atomic or composition of services, arise new legal challenges. Service composition and consumption lead to establishment of legal relationship between legal entities, e.g. service providers and consumers. The need for agreements to private law has to be considered in this case. The liability questions in case of claims have to be clarified for service composition and consumption. Especially for the composed services, where different service providers are involved, liability issue is very critical. The copyright issues concerning the compatibility of different licenses [J̈O8] in the case of service composition has to be considered very carefully too.

Moreover, the usage of consumers' personal data in composed services reduced transparency for service consumers. Even if the personal data is collected for a specific task within a distributed application, it is not clear for service consumers who else receive their personal data in complex and distributed applications. It raises privacy law issues for the service consumers. These issues are very critical for the success of distributed applications based on the composed services. The legal uncertainty and lack of trustworthiness may hinder commercial appliance of distributed applications. Unfortunately, the service composition makes it more difficult to foresee legal requirements [BPWR10, RWB+10].

The service description provides functional and non-functional parameters. This information can be used as starting point for the legal assessment. Furthermore, the European laws are codified [BPWR10] in a comprehensive and systematic way and include rules for legal consequences. It provides the opportunity to model legal norms and legal assessment procedure. The legal assessment can be considered as a service during development, composition and usage of services, which can identify legal obligation and requirements. The contracts, copyright and privacy law issues can be considered for legal assessment. Baumann et al. [BPWR10] introduces a legal methodology to formalize legal norms and process of legal assessment to obtain legal rights and obligations. The legal methodology motivates the separation of service description and formalized law. The mapping of formalized law to service description elements should bridge the gap as shown in Figure 4.2. The C_i and E_i in Figure 4.2 represent non-functional parameters, and context

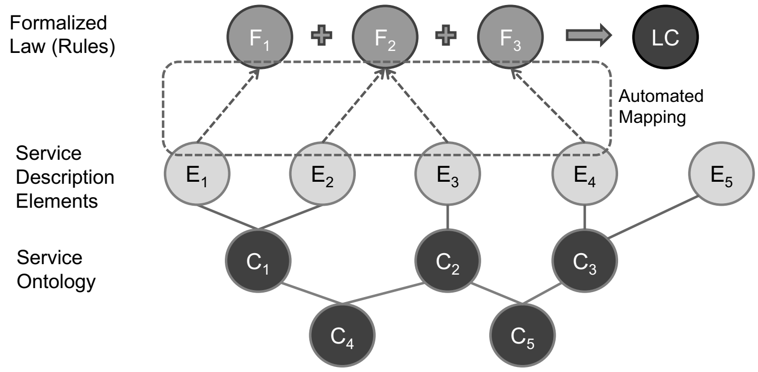


Figure 4.2: The legal methodology of Baumann et al. [BPWR10]

information from the data exchanged during the service execution (service description elements)respectively. The definition of formalized law (rules) is represented by F_i in Figure 4.2.

According the methodology of Baumann et al. [BPWR10], the functional parameters, e.g. inputs and outputs, and the non-functional aspects, e.g. service provider, are collected from the service description in the first step. Then the context information (service description elements) from the data exchanged during the service execution and business process model is collected, e.g. name and address of a service consumer. Finally, the mapping of formalized law to service description elements takes place. The legal aspects for the VSA are defined as follows:

Definition 4.1.6 (Legal Aspects). *Legal aspects of SDs define legal aspects such as agreement, obligations and penalties of a service according to national and international law.*

4.1.2 Service Level Agreement

The quality of service is one of the most important aspects to differentiate between similar services of service providers. The Service Level Agreement (SLA) between service providers and customers assures quality of service to customers and obligate service provider to achieve promised quality of service. The SLA violation may result serious financial consequences and affect the reputation of service providers. Hence service providers are interested in providing promised quality of service to achieve acceptance of service consumers [JS02].

The functional and non-functional parameters of services can be classified in three categories as it is already discussed in Chapter 2: Static attributes, dynamic attributes within the influence of service provider and dynamic attributes beyond the influence of service provider. An example of these parameters for a service is price and response time. The dynamic attributes within or beyond the influence of service providers are complex. They may change over time depending on market situation, competitors' offers and IT-infrastructure. For example, the price may change because of special offers of competitors and response time of a service may change because of peak hours. The service providers have to analyze the SLAs over time and have to adapt according to dynamically changing market conditions [JS02, HKRK07]. The SLA for the VSA is defined as follows:

Definition 4.1.7 (Service Level Agreement)**.** *A Service Level Agreement (SLA) of SDs defines all quality of service related non-functional parameters such as response time and availability.*

4.2 Requirements Analysis

A study was conducted to derive requirements for the VSA. The knowledge experts from different domains were interviewed in this study. The requirements for the VSA were derived from these interviews. The following sections give an overview of requirements from users' perspective, semantic analysis perspective and SD modeling perspective.

4.2.1 Requirements from Users' Perspective

The most important requirement of the knowledge experts is to have a process model to analyze the SDs of the available services and extract semantic information from them, model new services by using extracted semantics and reanalyze new modeled services with the available services. The analysis of available service helps them to identify similar services or compare different aspects of SDs to extract semantics. The usage of extracted semantic information enables them to reuse or modify different aspects of available services to model new services. The reanalysis allows knowledge experts to compare annotations, price plans, legal aspects and SLAs of new modeled services with available services. The iterative analysis, modeling and reanalysis of services must support semi-automatic modeling of SDs.

The SD modeling is a complex and interdisciplinary task, where the knowledge experts from different domains take part. The interdisciplinary tasks of the SD modeling demand multiple perspectives based SD modeling. The VSA must provide them different perspectives for the semi-automatic service analysis and modeling of price plans, legal aspects and SLAs. The multiple perspectives support them to model either each aspect of the SD separately in a perspective or different aspects of SD simultaneously in a single perspective. The selection of a sub set of services for the VSA in different perspectives must allow them to deal with a large number of services.

The analysis, modeling and reanalysis process with a large amount of available services can lead to information overload for knowledge experts. The knowledge experts require visual support during the analysis, modeling and reanalysis process to reduce information overload. The combination of semantic analysis and interactive visualization techniques allow them to have a effective understanding and reasoning about complex extracted semantic information as it is also proposed by the Visual Analytics [KAF+08]. The interactive visualization techniques must facilitate knowledge experts to focus on aspects of SDs according to their point of interest to avoid information overload during the analysis, modeling and reanalysis process.

- A process model for the VSA to support analysis, modeling and reanalysis of services
- Support for the analysis of available services to extract semantic information
- Support for the semi-automatic modeling of new services based on the extracted semantics from the available services
- Support for the reanalysis of new modeled services with the available services to compare new modeled services with available services
- Iterative analysis, modeling and reanalysis to support semi-automatic SD modeling
- Multiple perspectives to deal with complexity of SDs and a large number of services.
- Visual support for analysis, modeling and reanalysis of services

4.2.2 Requirements from Semantic Analysis Perspective

The semantic analysis process models present different abstraction levels as discussed in Chapter 3. An abstract process model for the semantic analysis is necessary to consider different process steps and different semantic analysis aspects. An abstract process model of the semantic analysis to support analysis of services is the most important requirements to achieve a generic process model for the Visual Semantic Analysis.

The SDs are mostly modeled with different SD modeling languages, e.g. XML, OWL, RDF(S), and UML, as it is already discussed in Chapter 2. The analysis of available SDs modeled with different SD modeling languages requires support for different data input formats of SDs in the VSA process model. It allows knowledge experts to analyze SDs modeled with different formats. The visual techniques must facilitate knowledge experts to have an overview and better understanding about complex context and semantics extracted in semantic analysis process.

Different semantic analysis algorithms cannot perform always the best results. The combination of different semantic analysis algorithms as multi-strategy seems to be a promising approach [Mae02]. The VSA approach must support multiple algorithms for the semantic analysis to improve the semantic analysis process. It allows knowledge experts to apply different semantic analysis algorithms to analyze available SDs and extract semantic information from them.

The integration of external context or semantic information must facilitate knowledge experts to use their own knowledge or extracted semantic information from different semantic analysis sessions for the semantic analysis. It allows them to use their own knowledge during the analysis of available services. Furthermore, the export of context or semantic information allows saving the results of semantic analysis and using them in different sessions. It helps knowledge experts to analyze available services in different sessions and use extracted semantic information during the analysis of available SDs.

- Abstract process model for the semantic analysis to support analysis of services
- Support for the different data input formats
- Visual support for the context analysis
- Visual support for the semantic analysis
- Support for the multiple semantic extraction algorithms
- Support for the import of external context or semantics in the semantic analysis process
- Support for the export of the extracted context and semantics in the semantic analysis process

4.2.3 Requirements from SD Modeling Perspective

According to Barros and Dumas [BD06], the SD modeling will become a bottleneck for service ecosystems. The semi-automatic semantic acquisition techniques have to be introduced to facilitate semi-automatic modeling of service description. The extension of semantic analysis process to facilitate semi-automatic modeling of SDs is one of the most important requirements for the VSA approach. Additionally, the reanalysis of new modeled services must support the comparison of new modeled services with available services.

The modeling of service description is an interdisciplinary task, where the experts from different domains take part. A modularization approach is mostly used to define modules for different domains, where domain experts are responsible for domain specific modules [OBB+09, CWV09]. A governance team monitors the modeling activities of knowledge experts to assure the consistency. The governance team assures consistency of SDs according to SD modeling guidelines. These aspects lead to the requirement that VSA must support modular approaches and governance for the SD modeling. Furthermore, the governance aspects have to be considered for the extension of semantic analysis process to assure consistency of new modeled SDs. The knowledge experts model SDs with different SD modeling languages that stipulate the support for the different SD modeling languages to model SDs in the VSA process model.

The SD modeling and governance efforts can be at risk if the semantic presentation and interactive manipulation cannot be accomplished in an intuitive and user-friendly way [OBB+09]. Therefore the visual support must facilitate knowledge experts to model SDs in an interactive way by using extracted semantics.

- An abstract process model for the SD modeling
- Support for the reanalysis to compare new modeled services with the available services
- Support for the Modularization approach of SD modeling
- Support for governance to assure consistency of new modeled SDs
- Support for different SD modeling languages
- Visual support for the SD modeling and reanalysis of new modeled services

4.2.4 Requirements Overview

This section gives an overview of concrete requirements as described in sections 4.2.1, 4.2.2, and 4.2.3. The requirements are grouped to nine requirements categories. They are used later on to compare the VSA approach with related efforts. They are assigned further to two main category types: “semantic analysis” and “service modeling”.

The category type “semantic analysis” consists of six requirements categories as shown in Table 4.1. The “process model/method for semantic analysis” indicates the need for process or method based approach and comprises all requirements that are directly related to semantic analysis process. The "data sources" related requirements for the semantic analysis are represented by the categories "support for unstructured data" and "support for semi-structured or structured data". The semantic analysis algorithms related requirements are depicted by the category "multiple semantic analysis algorithms". The categories "visual context analysis" and "visual semantic refinement" capture requirements related to the usage of visualization techniques for the context and semantic analysis in order to improve semantic analysis process.

Three requirements categories are assigned to the category type "service modeling" as depicted in Table 4.2. The "process model/ Method for the SD modeling" represents all requirements to extend semantic analysis process with the modeling specific aspects. The "visual support for SD modeling" manifests the need of visualization for the SD modeling to support knowledge experts. The requirement category "Analyze - Model - Reanalyze" demonstrates the need of techniques for the knowledge experts to analyze, model and reanalyze services.

Nr.	Categories	Requirements	Perspective
1	Process model/ Method	A process model for the VSA to support analysis, modeling and reanalysis of services	User
		Abstract process model for the semantic analysis to support analysis of services	Semantic Analysis
		Iterative analysis, modeling and reanalysis to support semi-automatic SD modeling	User
		Support for the analysis of available services to extract semantic information	User
		Support for the reanalysis of new modeled services with the available services to compare new modeled services with available services	User
		Visual support for analysis, modeling and reanalysis of services	User
		Support for the import of external context or semantics in the semantic analysis process	Semantic Analysis

Continued on next page

Nr.	Categories	Requirements	Perspective
		Support for the export of the extracted context and semantics in the semantic analysis process	Semantic Analysis
2	Support for unstructured data	Support for the different data input formats	Semantic Analysis
3	Support for semi-structured or structured data	Support for the different data input formats	Semantic Analysis
4	Multiple semantic analysis algorithms	Support for the multiple semantic extraction algorithms	Semantic Analysis
5	Visual Context Analysis	Visual support for the context analysis	Semantic Analysis
		Visual support for analysis, modeling and reanalysis of services	User
6	Visual Semantic Refinement	Visual support for the semantic analysis	Semantic Analysis
		Visual support for analysis, modeling and reanalysis of services	User

Table 4.1: Requirements Overview for the category semantic analysis

Nr.	Categories	Requirements	Perspective
1	Process model/ Method based	A process model for the VSA to support analysis, modeling and reanalysis of services	User
		An abstract process model for the SD modeling	SD Modeling
		Support for the semi-automatic modeling of new services based on the extracted semantics from the available services	User
		Multiple perspectives to deal with complexity of SDs and a large number of services	User

Continued on next page

Nr.	Categories	Requirements	Perspective
		Support for the Modularization approach of SD modeling	SD Modeling
		Support for governance to assure consistency of new modeled SDs	SD Modeling
		Support for different SD modeling languages	SD Modeling
2	Visual support for the SD modeling	Visual support for the SD modeling and reanalysis of new modeled services	SD Modeling
		Visual support for analysis, modeling and reanalysis of services	User
3	Analyze - Model - Reanalyze	Iterative analysis, modeling and reanalysis to support semi-automatic SD modeling	User
		Visual support for analysis, modeling and reanalysis of services	User
		Support for the reanalysis to compare new modeled services with the available services	SD Modeling

Table 4.2: Requirements Overview for the category service modeling

4.3 Comparison of Visual Semantic Analysis Approach with the Related Efforts

There are four approaches that deal with semi-automatic service description modeling directly. Further semantic analysis and visualization supported semantic analysis approaches are also related to this research work. The nine different requirement categories as discussed in Section 4.2.4 set our Visual Semantic Analysis approach apart from related efforts. The comparison of the Visual Semantic Analysis approach with related efforts according to nine requirements categories is presented in Table 4.3.

The Visual Semantic Analysis is a process model based approach for the semantic analysis and service modeling. Visual Semantic Analysis approach

		Semantic analysis						SD modeling		
Nr.	Approaches	Process model/ Method	Support for unstructured data	Support for semi-structured or structured data	Multiple semantic analysis algorithms	Visual Context Analysis	Visual Semantic Refinement	Process model/ Method based	Visual support for SD Modeling	Analyze - Visualize - Model - Reanalyze
1	Visual Semantic Analysis approach	•	•	•	•	•	•	•	•	•
2	Hess and Kushmerick [HK03]	◦	•	◦	◦	◦	◦	◦	◦	◦
3	Patil et al. [POSV04]	◦	•	•	◦	◦	◦	◦	◦	◦
4	Sabou [Sab06]	◦	•	◦	◦	◦	⋄	◦	◦	◦
5	Wei et al. [WWWC08]	◦	•	◦	◦	◦	◦	◦	◦	◦
6	Maedche et al. [Mae02]	•	•	•	•	◦	⋄	◦	◦	◦
7	Cimiano et al. [CHST04]	•	•	•	◦	◦	⋄	◦	◦	◦
8	Cimiano et al. [CHS05]	•	•	•	◦	◦	⋄	◦	◦	◦
9	ROD Methodology [ZBZ02, Zho07]	•	•	•	•	◦	◦	◦	◦	◦
10	PACTOLE methodology [BTN08]	•	•	•	•	◦	⋄	◦	◦	◦
11	Pan et al. [PF09b]	•	•	•	•	◦	⋄	◦	◦	◦
12	Jia et al. [JNT08]	•	•	◦	◦	◦	⋄	◦	◦	◦
13	Haav [Haa04]	•	•	◦	◦	◦	⋄	◦	◦	◦
14	Card et al. [CMS99]	•	•	•	◦	•	◦	◦	◦	◦
15	May et al. [MKJ08]	•	•	•	•	•	◦	◦	◦	◦
16	Keim et al. [KAF+08]	•	•	•	•	•	◦	◦	◦	◦
• = Supported, ◦ = Not supported, ⋄ = Not enough supported										

Table 4.3: Comparison of the VSA approach with existing research approaches

combines semantic analysis and visualization techniques to support semi-automatic modeling of SDs. It allows knowledge experts to analyze SDs of available services to extract semantic information, model new service on the base of extracted semantic information and then reanalyze the new-modeled services with available services to compare them. It allows knowledge experts to enter into a loop, where they can analyze, model and reanalyze service iteratively by using semantic analysis and visualization techniques. The requirements category Analyze - Model - Reanalyze is the most prominent difference between Visual Semantic Analysis and other related efforts, because it is only supported by the Visual Semantic Analysis.

The approaches 2 to 5 support both semantic analysis and SD modeling, but they are not process model/method based approaches. The approach offered by Hess and Kushmerick [HK03] supports just unstructured data for the semantic analysis. It employs Naive Bayes and SVM machine learning methods to extract semantic from WSDL files. The approach of Patil et al. [POSV04] supports unstructured, semi-structured and structured data for the semantic analysis. This approach presents conversion of WSDL and exiting ontology formats to a common representation format SchemaGraph and then maps the WSDL concepts to existing ontologies by applying match score techniques. These both approached do not allow the combination of different semantic analysis algorithms and offer visualization techniques to support SD modeling. The approaches proposed by Sabou [SWGS05][Sab06][SP07] and Wei et al. [WWWC08] both support unstructured data for semantic analysis. Both techniques use lexical analysis techniques for the semantic extraction and do not allow the combination of different semantic analysis algorithms. Wei et al. do not offer any visualization technique, but Sabou offers visualization technique to improve semantic analysis.

The approaches 6 to 16 allow process model based semantic analysis and do not deal with the SD modeling. Maedche et al. [Mae02], Cimiano et al. [CHST04] and Cimiano et al. [CHS05] facilitate usage of different data formats such as unstructured, semi-structured and structured data. The approach offered by Maedche et al. [Mae02] supports multiple semantic analysis algorithms and combination of different algorithms to improve semantic extraction. The both approaches from Cimiano et al. [CHST04] and [CHS05] apply Formal Concept Analysis (FCA) for the semantic extraction and do not allow the combination of different semantic analysis algorithms. These three semantic analysis techniques offer visual support, e.g. graph visualization or lattice visualization, to improve semantic extraction, but they do not exploit different interactive techniques for the semantic analysis.

The approaches ROD [ZBZ02, Zho07], PACTOLE [BTN08] , and Pan et al. [PF09b] allow different data formats and combination of different semantic analysis algorithms. The approaches of Jia et al. [JNT08] and Haav [Haa04] supports just unstructured data and do not support multiple semantic analysis algorithms. The methodology Rod does not offer visual support, but the other approaches PACTOLE, Pan et al. Jia et al. and Haav offer lattice visualization without using sophisticated interactive visualization techniques for the exploration of extracted semantic.

The approaches introduced by Card et al. [CMS99], May et al. [MKJ08] and Keim et al. [KAF+08] propose visualization techniques as essential part of data and semantic analysis. Card et al. propose usage of visualization technique for data to have better understanding about data and discuss visualization issues like overview, zoom and level of details without handling semantic analysis issues. May et al. [MKJ08] and Keim et al. [KAF+08] suggest that the visualization techniques, and data and semantic analysis have to go hand in hand to achieve better data and semantic analysis results. These both approaches originate from visual analytics, which focuses more on the data analysis rather than semantic analysis. They deal with data same like semantic analysis, but it doesn't deal with validation, refinement and merging of extracted semantics. The usage of visualization techniques for the data and semantic analysis presented by [CMS99], May et al. [MKJ08] and Keim et al. [KAF+08] play an important role for the Visual Semantic Analysis approach.

4.4 Visual Semantic Analysis Process Model

A generic process model for the Visual Semantic Analysis is developed on the base of the requirements analysis. The generic VSA process model is divided into two parts: semantic analysis specific sub processes and Service modeling specific sub processes as shown in Figure 4.3. The semantic analysis specific sub processes are "visual context acquisition", "visual context analysis", "semantic extraction" and visual semantic refinement". The sub processes "visual SD modeling" and "governance" belong to service modeling specific sub processes. The sub processes are described in the following sections.

4.4.1 Semantic analysis specific sub processes

Different semantic analysis process models were discussed in Chapter 3. They consist of different sub processes and tasks with different abstraction levels. The VSA process model defines an abstract semantic analysis process model. The semantic analysis specific sub processes in the generic VSA process model are described below.

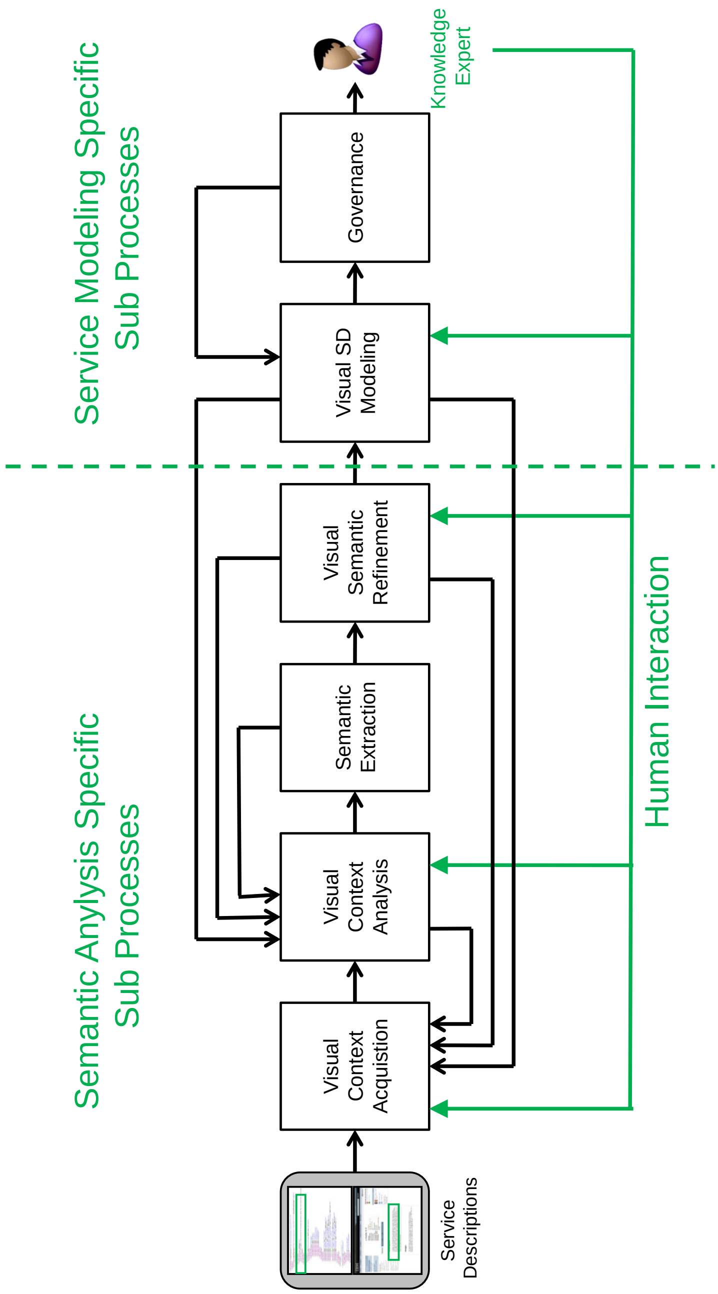


Figure 4.3: A Process Model for the Visual Semantic Analysis

Visual Context Acquisition

The context acquisition is an essential part of all semantic analysis approaches. All semantic analysis approaches support unstructured, semi-structured and/or structured data input formats as it is also perceptible

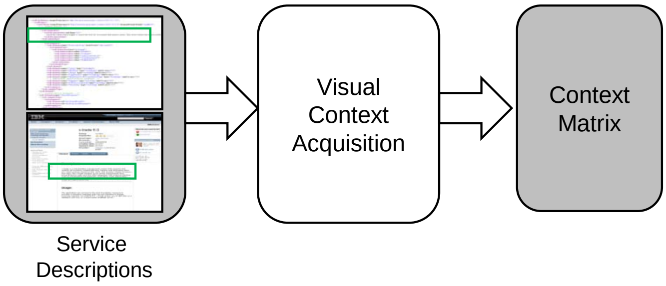


Figure 4.4: The sub process visual context acquisition

The extracted context information from the non-functional parameters of SDs is mostly very complex and the application of semantic analysis techniques to the complex context is very difficult. In this case, the knowledge experts transform complex context information into suitable context information in the task conceptual scaling. For this purpose, the non-functional parameters of SDs are considered as objects that are described by new scale attributes. The transformation is not unique and allows different interpretations and views of knowledge experts [Wol94, FMCNFV98]. The conceptual scaling can be done with different levels of details or granularity. The transactions from the sub processes visual context analysis, visual semantic refinement and visual SD modeling to the visual context acquisitions facilitate to reinitialize conceptual scaling.

The textual service description and non-functional parameters of SDs with different formats serve as input of this sub process. The output of this sub process is a context matrix as shown in Figure 4.4, where the rows and columns represent services and attributes (terms extracted from textual service description and non-functional parameters) respectively. The values in this context matrix are the *tfidf* (see section 2.2) values of attributes extracted from textual service description or real values of the non-functional parameters.

Visual Context Analysis

The knowledge experts visualize the extracted context matrix by using different visualization techniques in this sub process. The interactive visualization techniques allow them to have an overview of the extracted context matrix to have better understanding about the extracted context. The functionalities of interactive visualization techniques such as Overview, zoom, filter and details-on-demand facilitate the knowledge experts to explore the extracted context and focus on the part of it according to their needs. The two sub processes visual context acquisition and visual context analysis, and transitions between them (see Figure 4.3) represent an abstract form of the reference model of the information visualization introduced by Card et al. [CMS99] as discussed in Chapter 3.

The Knowledge experts prune or refine the extracted context matrix in this sub process by using different semantic extraction techniques from the sub process semantic extraction as it is shown in Figure 4.5. They apply semantic extraction techniques to analyze context information and then visualize the pruned or refined context matrix to view it. The three sub process visual context acquisition, visual context analysis, and semantic extraction, and the transition between sub processes semantic extraction and visual context analysis (see Figure 4.3) represent abstract form of the Visual Analytics (VA) process models presented by May et al. [MKJ08] and Keim et al. [KAF$^+$08]. These both VA process models are introduced in Chapter 3 and depicted in figures 3.12 and 3.13.

The VSA techniques allow knowledge expert to combine semantic extraction techniques and appropriate visualization technique to reduce the dimensionality of the context matrix by clustering similar services and visualizing the clusters. They apply these semantic extraction techniques from the sub process semantic extraction and view the results in the sub process visual context analysis. The feedback loop between the sub processes semantic extraction and visual context analysis allows them to enter into a loop to gain insight into context information and perform more focused analysis. The visual representation of interactive context analysis facilitates them to gain better understanding about data and confirm or reject previous iteration of context analysis.

In the case of the textual service description of SDs, the semantic analysis techniques, e.g. dictionary parsing and pattern matching, as discussed in section 2.2 facilitate knowledge experts to extract real concepts (attributes).Additionally, the lexical analysis and lexical pattern matching as introduced in section 2.2 allow to refine the context matrix. The combination of semantic analysis and visualization techniques allow the knowledge experts to discover hidden facts or patterns form the context extracted from the textual service description of SDs.

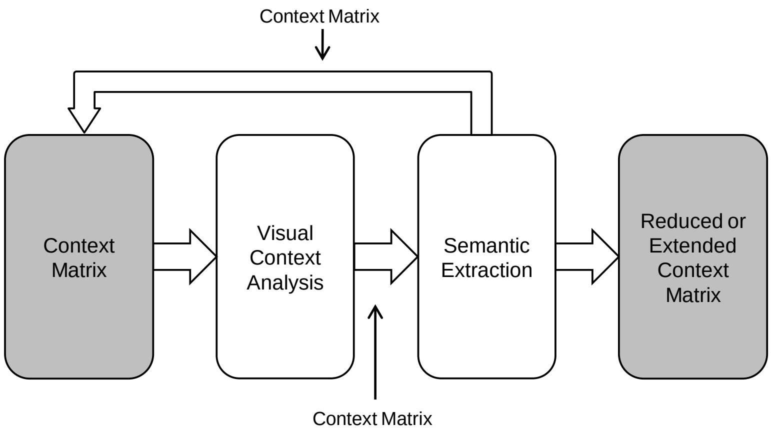


Figure 4.5: The sub processes visual context analysis and semantic extraction

The visual context analysis for a large number of available services can lead to information overload for knowledge experts. The multiple perspectives allow knowledge experts to consider a partial context and a sub set of services for the Visual Semantic Analysis. They facilitate the consideration of context from different perspectives. Additionally, the perspectives based VSA allow the knowledge experts to deal with a large number of services by considering sub sets of services for the VSA in different perspectives.

The input of this sub process is a context matrix as shown in Figure 4.5. The output of this sub process is a refined context matrix. The perspective based VSA generates one or more refined context matrixes. The import and export of context matrixes facilitate integration of external context matrixes and storage of the results of the visual context analysis respectively.

Semantic Extraction

The main semantic analysis techniques can be classified in four different categories "statistics based", "rule-based", "hybrid" (statistics and rule based) and "Formal Concept Analysis (FCA)" [SP07, CHS05, Zho07, Mae02] as discussed in the section 2.2. A bundle of semantic extraction algorithms are offered to the knowledge experts in this sub process. They can apply a single semantic extraction technique or a combination of these techniques to improve the results of the visual context analysis or semantic analysis.

The input of this sub process is a context matrix. The outputs of this sub process are either a context matrix or semantics (graph) depending on the successor sub process. The usage of different semantic analysis techniques determines the output formats. The output formats of all semantic extraction algorithms used in the VSA have to be mapped to the input formats of the successor sub processes. The input formats of the successor sub processes visual context analysis and visual semantic refinement are context matrix and semantics respectively.

Visual Semantic Refinement

Semantic extraction techniques such as the Formal Concept Analysis (FCA) and dictionary parsing can be applied to extract semantics from the context matrix as described in Chapter 2 . The extracted semantics has to be validated and refined by the knowledge experts in this sub process. The semantic visualization techniques facilitate the knowledge experts to visualize the extracted semantics in order to understand complex hierarchies of concepts and their semantic relations in the extracted semantics. The knowledge expert validate and refine extracted semantics by adding or removing new concepts, attributes and relations in this sub process. The sub processes visual context acquisition, visual context analysis, semantic extraction and visual semantic analysis, and transition between them (see Figure 4.3) represent an abstract semantic analysis process model as combination of process models of semantic analysis and Visual Analytics techniques presented in Chapter 3.

Additionally, the knowledge experts model their own knowledge manually in this sub process. The feedback loop between visual semantic refinement and the visual context analysis allows the merging of extracted semantics and manual generated semantics. The merging is done by using semantic merging techniques, e.g. FCA, from the sub process semantic extraction. The merged semantic can be analyzed again in this sub process.

The input and output of this sub process are extracted semantics, and validated and refined semantics respectively. The import and export of external semantics allows knowledge experts to save the extracted semantics and integrate external semantics in this sub process. The extracted semantics from different perspectives can also be integrated in the extracted semantics of the current perspective.

4.4.2 Service Modeling Specific Sub-Process

The sub processes visual SD modeling and governance are the extension of semantic analysis process in the VSA to support semi-automatic modeling of SDs. It supports the SD modeling by offering extracted semantic and visualization techniques. It helps them not only to rely on their own view of the domain, but also use semi-automatically extracted semantic during the service modeling. The following sections describe service modeling specific sub processes.

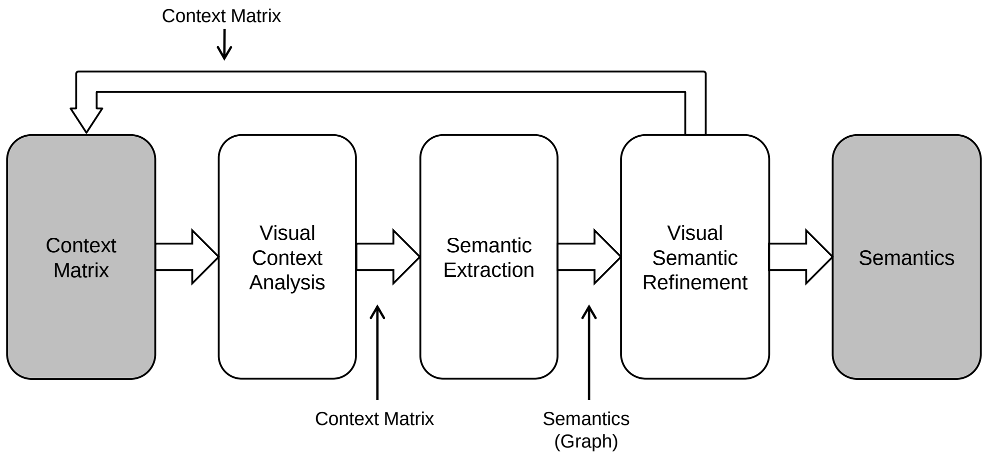


Figure 4.6: The sub process visual semantic refinement and its related transitions

Visual SD Modeling

The semantic analysis specific sub processes allow the knowledge experts to analyze SDs of available services and extract semantics from them. The extracted semantics facilitate them to identify services similar to their planed new services. The knowledge experts model SDs of their new services in this sub process by using the extracted semantics. The reuse or adaption of the extracted semantics for the modeling of new services support the semi-automatic SD modeling of new services and accelerate the service modeling process. The interactive visualization techniques as visual modeling techniques facilitate the knowledge experts to understand the extracted semantics, and reuse or adapt it for the SD modeling of new services. Afterwards, the feedback loop between visual SD modeling and visual context analysis allows them to reanalyze the new modeled services with the available similar services as depicted in Figure 4.7. The reanalysis makes the comparison of new modeled services with the available services easier. The analysis of the available services, semi-automatic SD modeling based on the extracted semantics and reanalysis of new modeled services support the metaphor Analyze - Model - Reanalyze. The knowledge experts can iterate analysis, modeling and reanalysis of services until they are satisfied with the new-modeled services.

The perspectives based VSA allows the knowledge experts to analyze, model and reanalyze different aspects of SDs, e.g. service annotation, price plan, legal aspects or SLAs, simultaneously or separately. A large number of services can lead to complex semantic. The perspective based VSA permit the

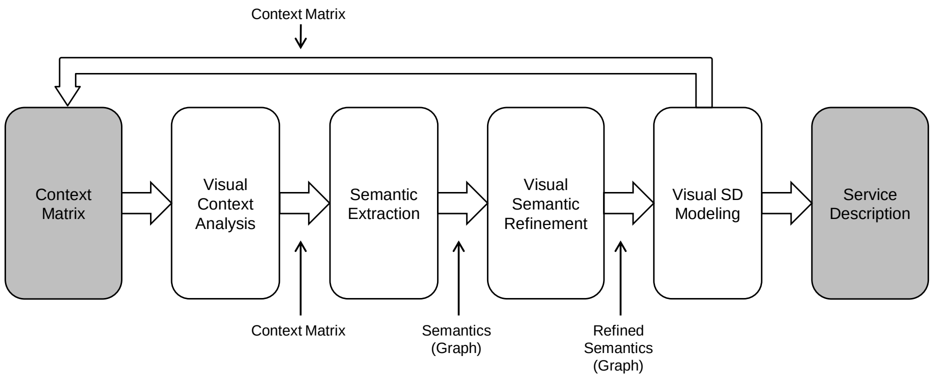


Figure 4.7: The sub process visual SD modeling and its related transitions

knowledge experts to consider sub sets of services in different perspectives in order to deal with a large number of services. The import of different aspects of SDs from different perspectives into a new perspective helps them to integrate them as SDs of new services.

The sub process governance validates the SDs of the new modeled services and provides the feedback as SD validation report. The errors are visualized to the knowledge experts in the sub process visual SD modeling. They have to change SDs of new modeled services until the validation process is successful.

The input of this sub process is extracted semantics from the semantic analysis specific sub processes and the SD validation report of the sub process governance. The output of this sub process is SDs of new services that can be mapped to different formats, e.g. service ontology [OBB+09] or USDL [CWV09].

Governance

The SDs of new modeled services have to be checked before they can be published on service platforms. This sub process uses SD validation techniques for the validation of new modeled SDs. The SD validation techniques assure the consistency of the SD modeling according to the modeling guidelines.

This sub process synchronizes SD modeling activities and SD validation via feedback loop between them as shown in Figure 4.8. The visualization of validation feedback in the sub process visual SD modeling supports knowledge experts to check consistency of new modeled SDs. In the case of a successful SD validation, the new modeled SDs can be published on service platforms.

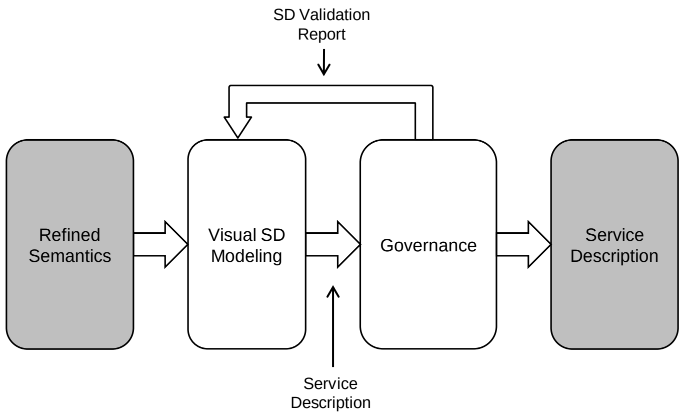


Figure 4.8: The sub process visual SD modeling and governance

4.5 Formal Specification of the VSA Process

A process definition consists of network of activities, relations between activities, process start and termination conditions, and further activity-related information such participants, connected applications, data, etc [WFM99]. Hofmann [Hof10] has introduced a formal specification of multimedia annotation. It includes basic classes of information entities contained in an annotation workflow and serve as scientific base for the formal specification of the VSA process. The formalization of the Formal Concept Analysis [GW99] (see Chapter 2 supports to define context and semantics generated and processed in the VSA process. In this section, first a formal specification of a generic process is presented and then the VSA process is specified on the base of the formal specification of the generic process. The VSA process specification includes the elements: (i) VSA process, (ii) work items, (iii) transition, (iv) technologies, and (iv) data specifications generated and processed. Furthermore, context and semantics are defined in the context of the VSA process.

Definition 4.5.1 (Process). *A process is defined by a tuple* $p = (W, T, C, A, D)$ *where :*

- $W = \{w_1, w_2, w_3, ..., w_j\}$ *is a finite set of work items in* p*.*
- $T = \{t_1, t_2, t_3, ..., t_k\}$ *is a finite set of transition between work items.*
- $C = \{c_1, c_2, c_3, ..., c_l\}$ *is a finite set of termination conditions.*
- $A = \{a_1, a_2, a_3, ..., a_m\}$ *is a finite set of applications available in* p*.*
- $D = \{d_1, d_2, d_3, ..., d_n\}$ *is a finite set of data processed, edited, generated and transferred between work items of* p*.*

Definition 4.5.2 (Work Item)**.** *A work item is defined as a tuple* $w = (C_i, C_o, A_w, D_i, D_o) \in W$ *where* :

- $C_i \subseteq C$ *is a finite set of input conditions for w.*
- $C_o \subseteq C$ *is a finite set of output conditions for w.*
- $A_w \subseteq A$ *is a finite set of applications to perform w.*
- $D_i \subseteq D$ *is a finite set of data transferred to w as input.*
- $D_o \subseteq D$ *is a finite set of data forwarded by w as output.*

Definition 4.5.3 (Transition)**.** *A transition is defined as a tuple* $t = (w_o, w_d) \in T$ *where* :

- w_o is an origin work item.
- w_d is a destination work item.

Visual Semantic Analysis Process. According to Definition 1.5.1 the Visual Semantic Analysis process is defined as a process $VSA = (W_{vsa}, T_{vsa}, C_{vsa}, A_{vsa}, D_{vsa})$ *where* :

- $W_{vsa} = \{Visual\ Context\ Acquisition,\ Visual\ Context\ Analysis,\ Semantic\ Extraction, Visual\ Semantic\ Refinement,\ Visual\ SD\ Modeling,\ Governance\}$ *is a set of work items in vsa.*
- T_{vsa} *is a finite set of transition between work items according to Figure 4.3.*
- $C_{vsa} = \{c_{vsa_1}, c_{vsa_2}, c_{vsa_3}, ..., c_{vsa_o}\}$ *is a finite set of termination conditions in VSA.*
- $A_{vsa} = Natural\ Language\ Processing \cup Visualization \cup Semantic\ Analysis \cup Service\ Validation,$
  is a set of techniques from the field of Natural Language Processing (NLP), visualization, semantic analysis, and service validation available in VSA.
- $D_{vsa} = Service\ Descriptions \cup Context \cup Semantics \cup Service\ Validation\ Reports,$
  is a finite set of data specifications Service Descriptions (SDs), Context (see Definition), Semantics (see Definition), and service validation reports processed, edited, generated and transferred between work items of VSA

Visual Context Acquisition. According to Definition 1.5.2 the Visual Context Acquisition is defined as a work item
$VisualContextAcquisition = (C_{acq_i}, C_{acq_o}, A_{acq}, D_{acq_i}, D_{acq_o})\ where:$

- $C_{acq_i} \subseteq C_{vsa}$ *is a finite set of input conditions for Visual Context Acquisition.*
- $C_{acq_o} \subseteq C_{vsa}$ *is a finite set of output conditions for Visual Context Acquisition.*
- $A_{acq} = Natural\ Language\ Processing\ \cup\ Visualization,$
is a finite set of techniques from the field of NLP and visualization available in Visual Context Acquisition.
- $D_{acq_i} = Service\ Descriptions,$
is a finite set of SDs transferred to Visual Context Acquisition as input.
- $D_{acq_o} = Context,$
is a finite set of context generated, processed and forwarded by Visual Context Acquisition as output.

Visual Context Analysis. According to Definition 1.5.2 the Visual Context Analysis is defined as a work item
$VisualContextAnalysis = (C_{vca_i}, C_{vca_o}, A_{vca}, D_{vca_i}, D_{vca_o})\ where:$

- $C_{vca_i} \subseteq C_{vsa}$ *is a finite set of input conditions for Visual Context Analysis.*
- $C_{vca_o} \subseteq C_{vsa}$ *is a finite set of output conditions for Visual Context Analysis.*
- $A_{vca} = Visualization,$
is a finite set of techniques from the field of visualization available in Visual Context Analysis.
- $D_{vca_i} = Context,$
is a finite set of context transferred to Visual Context Analysis as input.
- $D_{vca_o} = Context$ *is a finite set of context generated, processed and forwarded by Visual Context Analysis as output.*

Semantic Extraction. According to Definition 1.5.2 the Semantic Extraction is defined as a work item
$SemanticExtraction = (C_{se_i}, C_{se_o}, A_{se}, D_{se_i}, D_{se_o})\ where:$

- $C_{se_i} \subseteq C_{vsa}$ *is a finite set of input conditions for Semantic Extraction.*

- $C_{se_o} \subseteq C_{vsa}$ *is a finite set of output conditions for Semantic Extraction.*
- $A_{se} = Semantic\ Analysis$ *is a finite set of techniques from the field of semantic analysis available in Semantic Extraction.*
- $D_{se_i} = Context$ *is a finite set of context transferred to Semantic Extraction as input.*
- $D_{se_o} = Semantics\ \cup Context,$
is a finite set of context and semantics generated, processed and forwarded by Semantic Extraction as output.

Visual Semantic Refinement. According to Definition 1.5.2 the Visual Semantic Refinement is defined as a work item
$VisualSemanticRefinement = (C_{vsr_i}, C_{vsr_o}, A_{vsr}, D_{vsr_i}, D_{vsr_o})\ where:$

- $C_{vsr_i} \subseteq C_{vsa}$ *is a finite set of input conditions for Visual Semantic Refinement.*
- $C_{vsr_o} \subseteq C_{vsa}$ *is a finite set of output conditions for Visual Semantic Refinement.*
- $A_{vsr} = Visualization$ *is a finite set of techniques from the field of visualization available in Visual Semantic Refinement.*
- $D_{vsr_i} = Semantics$ *is a finite set of semantics transferred to Visual Semantic Refinement as input.*
- $D_{vsr_o} = Semantics\ \cup Context,$
is a finite set of context and semantics generated, processed and forwarded by Visual Semantic Refinement as output.

Visual SD Modeling. According to Definition 1.5.2 the Visual SD Modeling is defined as a work item
$VisualSDModeling = (C_{vsdm_i}, C_{vsdm_o}, A_{vsdm}, D_{vsdm_i}, D_{vsdm_o})\ where:$

- $C_{vsdm_i} \subseteq C_{vsa}$ *is a finite set of input conditions for Visual SD Modeling.*
- $C_{vsdm_o} \subseteq C_{vsa}$ *is a finite set of output conditions for Visual SD Modeling.*
- $A_{vsdm} = Visualization$ *is a finite set of techniques from the field of visualization available in Visual SD Modeling.*
- $D_{vsdm_i} = Semantics \cup Service\ Validation\ Reports,$
is a finite set of context and service validation reports transferred to Visual SD Modeling as input.

- $D_{vsdm_o} = Service\ Descriptions(SDs)$ *is a finite set of SDs generated, processed and forwarded by Visual SD Modeling as output.*

Governance. According to Definition 1.5.2 the Governance is defined as $Governance = (C_{gov_i}, C_{gov_o}, A_{gov}, D_{gov_i}, D_{gov_o})\ where:$

- $C_{gov_i} \subseteq C_{vsa}$ *is a finite set of input conditions for Governance.*
- $C_{gov_o} \subseteq C_{vsa}$ *is a finite set of output conditions for Governance.*
- $A_{gov} = Service\ Validation,$ *is a finite set of techniques from the field of service validation available in Governance.*
- $D_{gov_i} = Service\ Descriptions,$ *is a finite set of SSDs transferred to Governance as input.*
- $D_{gov_o} = Service\ Validation\ Reports \cup Service\ Descriptions(SDs),$ *is a finite set of service validation reports and SDs generated, processed and forwarded by Governance as output.*

Definition 4.5.4 (Context)**.** *The context generated and processed in the VSA process is defined as* $Context\,(S, \Phi, I)\ \ where:$

- S *is a finite set of Service Descriptions (SDs).*
- Φ *is a finite set of attributes extracted from SDs.*
- $I \subseteq S \times \Phi$

Definition 4.5.5 (Semantics)**.** *The semantics generated and processed in the VSA process is a set* Σ*. The concept of the context* $Context\,(S, \Phi, I)$ *is* (A, B)*, where* $A \subseteq S\ ,\ B \subseteq \Phi$*,.For the concepts* (A_1, B_1) *and* (A_2, B_2) *from the set* Σ *of all the concepts of* $Context\,(S, \Phi, I)$ *:*

$$\begin{aligned}(A_1, B_1) \leq (A_2, B_2) \ &\Leftrightarrow\ A_1 \subseteq A_2 \\ &\Leftrightarrow\ B_1 \supseteq B_2\end{aligned} \tag{4.1}$$

The relation $\leq$ *is an order on* Σ *e.g. the concept* (A_1, B_1) *is less general as compared to the concept* (A_2, B_2)*.*

4.6 Summary

The Visual Semantic Analysis (VSA) is presented in this chapter to support semi-automatic modeling of the Service Description as formalized representation of non-functional aspects of services. The VSA approach may help to establish service ecosystems in near future. The terms used in this generic process model are defined and the requirements analysis for the VSA is discussed in detail in this chapter. The comparison of VSA approach with the existing related research efforts gives a brief overview of its features. A generic process model of the VSA that describes the sub processes and their task, and the execution order of the sub processes. Finally, a formal specification of the basic elements of the VSA process is presented.

This chapter describes only the process model of the VSA approach. A conceptual framework for the VSA approach is described in the next chapter. It presents in detail the application of VSA approach in the service ecosystems. For example, how the VSA approach can use different platform services of service ecosystems for the semantic analysis to support semi-automatic modeling of SD. Furthermore, the software components are also presented in the next chapter, which allow the realization of different tasks in the sub-processes of the Visual Semantic Analysis process model.

Chapter 5

A Conceptual Framework for the Visual Semantic Analysis

The process model for the Visual Semantic Analysis (VSA) approach is described in the previous chapter. It supports the *semi-automatic* modeling of Service Descriptions (SDs), which include formalized description of non-functional aspects like service annotation, price plans, legal aspects, and SLAs. The conceptual framework is developed on the base of the VSA process model. It describes system behavior and system components of the Visual Semantic Analysis System. The system components manage sub processes and their execution order according to the generic process model of the VSA.

The services of service ecosystems (service platforms) play an important role for the visual semantic refinement. In this chapter, we present the system components of *visual semantic refinement system* and its interaction with service platforms. The VSA system imports available services from *service platforms*, analyzes them, model new services on the base of service analysis results and register new modeled services on the *service platform* by using the *service registry* service.

A conceptual framework for the Visual Semantic Analysis (VSA) to support *semi-automatic* modeling of Service Descriptions is depicted in Figure 5.1. The *visual semantic refinement system* uses different platform services for the analysis, modeling and reanalysis of Service Descriptions (SDs).

The *visual semantic refinement system* collects the Service Descriptions (SDs) and uses *context acquisition* component to extract context by applying Natural Language Processing (NLP) and conceptual scaling techniques. It offers multiple perspectives for the visual semantic refinement, which allow the knowledge experts to analyze, model and reanalyze different aspects

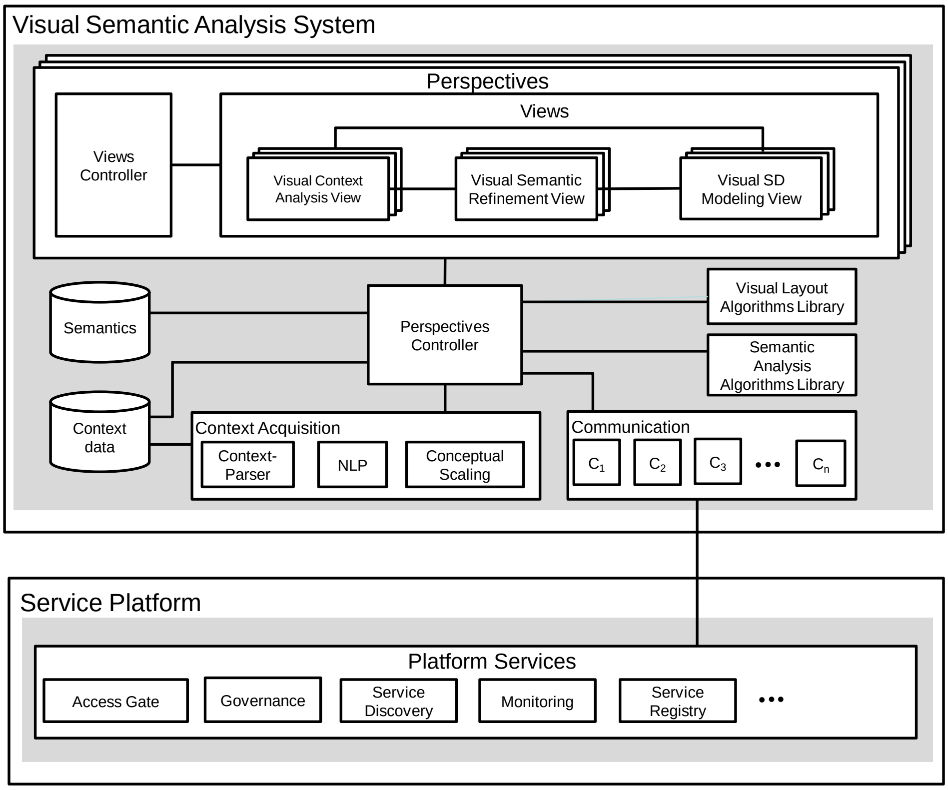


Figure 5.1: A conceptual framework for the Visual Semantic Analysis

of SDs such as price plans and SLAs in different *perspectives* separately.

Each *perspective* offers three different views *visual context analysis view*, *visual semantic refinement view* and *visual SD modeling view*. Each view offers different visualization techniques by using visual layout algorithms from the *visual layout algorithms library*. The views *visual context analysis view* and *visual semantic refinement view* support the knowledge experts to analyze the available services by using different semantic analysis algorithms available in the *semantic analysis algorithms library*. After the semantic analysis of services is completed, the knowledge experts can save the enriched context and extracted semantic information in repositories *context data* and *semantics*.

After the semantic analysis of services is completed, the knowledge experts model their new services in the *visual SD modeling view* by taking into consideration the results of service analysis. The *visual SD modeling view* offers different interactive visualization techniques to visualize the semantic analysis results and allows the knowledge experts to model new SD by reusing and adapting the semantic analysis results. After the SD modeling, the knowledge experts reanalyze the new modeled services with the existing service to compare them with the existing services. They can iterate this process until they are satisfied with the new modeled service. Finally, the *communication* components allow them to check the SD validity of the new modeled services by using the *governance service* of the *service platform*. The results of the SD validation are visualized in the *visual SD modeling view*. After a successful SD validation, the VSA system publish new modeled service by using the *service registry* service of the *service platform*. The *communication* component manages the external communication, e.g. service registry. The components of the conceptual framework for the VSA are described in the following sections.

5.1 Visual Semantic Analysis System

According to Bertin's theory "Semiology of Graphics" [Ber83], the basic visual variables play a key role for human perception. He defined the eight visual variables position, shape, orientation, color, motion, texture, value, and size as depicted in Figure 5.2. The choice of a visual variable for specific information can affect human perception for the information. Bertin's theory serves as a basis for different research work such as [Mac86, Bry10]. Mackinlay identified further visual variables such as length, volume, connection, and containment [Mac86]. He developed an order for the visual variables for the quantitative, ordinal and nominal data as shown in Figure 5.3. For example, the visual variable length (e.g. bar chart) is suited well for the quantitative data, but it is not convenient for ordinal and nominal data.

Different types of data formats are generated within the sub processes of the VSA process model as discussed in Chapter 4. According to Mackinlay, different visualization techniques with specific visual variables may suite for different type of data formats. Different visualization techniques (approximately 300 - 400) are available right now, which can be suitable for specific task or data formats [God09, NBB+10]. Therefore, the VSA conceptual framework offers different visualization techniques for its different *views*.

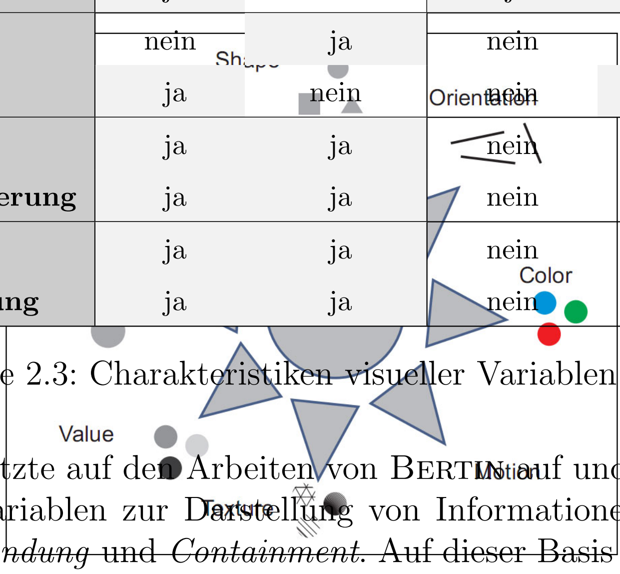
To facilitate the use of different visualization techniques for the same data, the Model-View-Controller (MVC) design pattern is used in the conceptual framework of the VSA. It allows the integration of further visualization techniques later to extend views easily. MVC is a classic design pattern often used by applications that need the ability to maintain multiple *views* of the same data. The MVC pattern divides application into three units model for maintaining data, view for information visualization and controllers for handling events that affect the model or view(s) [BMR+96].

The fundamental of Human Visual Perception [War04] and Visual Informaunterstützen diese Perzeption in unterschiedlichem Maße, dargelegt in Tabelle 2.3. In der Informationsvisualisierung kann durch gezielte Anwendung visueller Variablen eine geeignete Perzeption und Interpretation der Informationen erreicht werden.

ist von der Variable *Farbe* zu trennen (siehe Tabellen 2.3, ???). Für moderne Visua... et. In Abbildu...

	Selektiv	Assoziativ	Geordnet	Quantitativ
Position	ja	ja	ja	ja
Größe	ja	nein	ja	ja
Form	nein	ja	nein	nein
Wert	ja	nein	nein	ja
Farbe	ja	ja	nein	nein
Orientierung	ja	ja	nein	nein
Textur	ja	ja	nein	nein
Bewegung	ja	ja	nein	nein

Tabelle 2.3: Charakteristiken visueller Variablen ([Qel07])



MACKINLAY setzte auf den Arbeiten von BERTIN auf und identifizierte weitere visuelle Variablen zur Darstellung von Informationen: *Länge*, *Winkel*, *Volumen*, *Verbindung* und *Containment*. Auf dieser Basis entwickelte er eine Rangreihenfolge der visuellen Variablen für quantitative (Mengen), ordinale (Reihenfolge) und nominale (keine Reihenfolge) Daten (siehe Abbildung 2.9) [Mac86].

Abbildung 2.8: Variablen der visuellen Wahrnehmung des Menschen ([Qel07]) ??? germanize!!

Figure 5.2: The eight visual variables [Ber83]

Eine geeignete Auswahl visueller Variablen zur Übermittlung bestimmter Infor... zer die Informati... ie Perzeptionsch... itative Perzeptio...

Assoziat... andere überdeckt... Größe eines Obj... ung. So ist die Or...

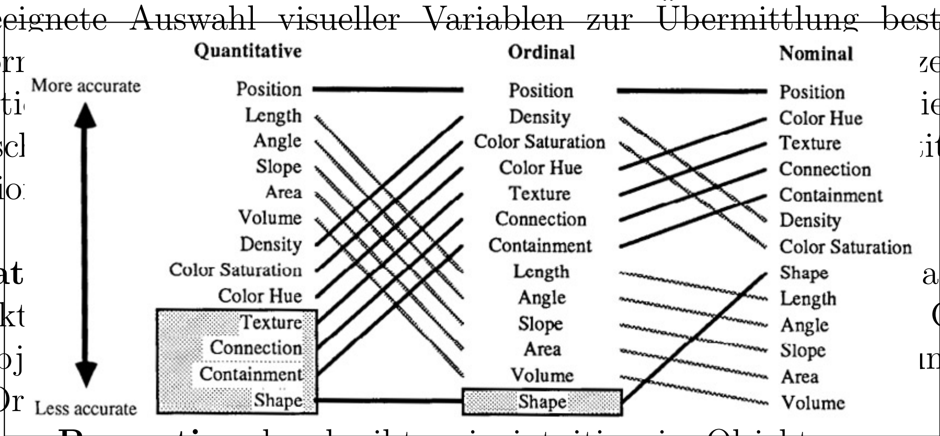


Abbildung 2.9: Rangreihenfolge visueller Variablen für quantitative, ordinale und nominale Daten ([Mac86]) ??? germanize!!

Figure 5.3: The order of the visual variables for the quantitative, ordinal and nominal data [Mac86, Bry10]

Selektive Perzeption beschreibt, wie intuitiv ein Objekt von anderen visuell ...

Geordnete Perzeption einer visuellen Variable entspricht einer wahrgenommenen (nicht mathematischen) Reihenfolge.

Quantitative Perzeption schränkt die geordnete Perzeption weiter ein. Ein Benutzer kann den Unterschied und die Quantität dessen erkennen.

tion Seeking Mantra [Shn96] as discussed in Chapter 2 serve as basis for the interaction design of all available visualization techniques offered in the ... overview, zoom, filter, and details-on-demand to support the visual semantic analysis process. This approach allow the knowledge experts to explore context and semantics in the views *visual context analysis view* and *visual semantic refinement view*.

Die Perzeptionscharakteristiken beschreiben, wie Menschen wahrgenommene Informationen in der Regel interpretieren und welche Gesetzmäßigkeiten den Informationen kognitiv auferlegt werden. Verschiedene visuelle Variablen

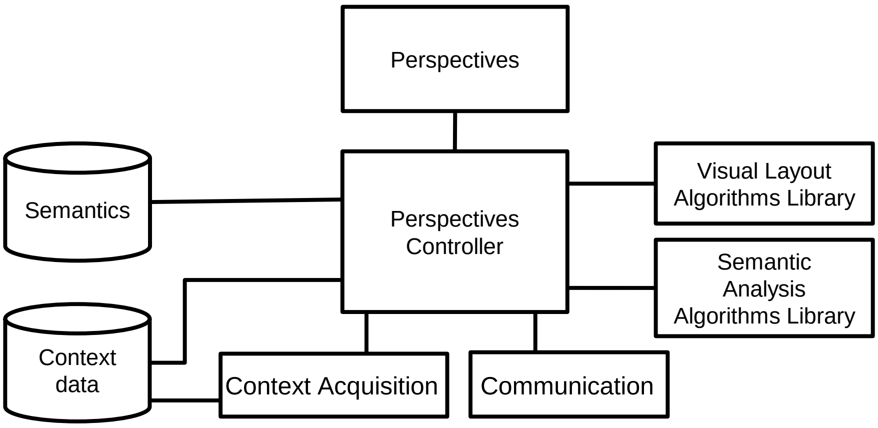


Figure 5.4: The perspectives controller as a bridge between all the components of the visual semantic analysis system

5.1.1 Perspectives Controller

The *perspectives controller* supports the perspectives based VSA by considering sub sets of services and partial context information. The perspectives based VSA enables the knowledge experts to deal with the complexity of SDs and a large number of services. The *perspectives controller* allows the knowledge experts to create new *perspectives* by selecting partial context or sub sets of services.

It also serves as a bridge between all the components of the *visual semantic analysis system* as shown in Figure 5.4. It facilitates the knowledge experts to import services from the *service platform* via *communication* components, extracts context information from them by using the *context acquisition* component and stores context in the repository *context data.* It manages *perspectives* and assigns data models, e.g. context data, to *perspectives.*

It supports the knowledge experts to use different semantic analysis and visualization techniques in the *perspectives.* The VSA approach supports different semantic analysis algorithms and combination of them for the visual semantic analysis process. The combination of different semantic analysis algorithms leads to improve VSA process as it is proposed by Maedche [Mae02]. The usage of different visualization techniques for the VSA makes easier to understand the hidden facts in the extracted semantics for the knowledge experts. Different semantic analysis and visualization techniques are offered in the VSA conceptual framework by the *visual layout* and *semantic analysis algorithms libraries.* After the service analysis is completed, the knowledge experts model new services on the base of semantic analysis results. The *perspectives controller* allows publishing of the new modeled services on the *service platform* by using *communication* components.

According to the MVC-pattern, the *perspectives* and the *perspectives controller* are seen as views and controller. The semantics and context data serves as data model. The *perspectives controller* manages the interaction of the knowledge experts, when they decide to analyze partial context and/or sub sets of services, and switch between different *perspectives* during the VSA process.

The *perspective controller* allows the knowledge experts to save partial context and extracted semantics from different perspectives in the repositories *context data* and *semantics*. The import of *context data* and *semantics* from the repositories facilitate them to integrate context and semantics in the VSA process. In this way, the knowledge experts can analyze context and extract semantics in different sessions or compare the results of different sessions in different perspectives during the VSA process.

5.1.2 Context Acquisition

The *context acquisition* component of the *VSA conceptual framework* allows the knowledge experts to extract context information from available SDs on the *service platform*. The knowledge experts import available services from the *service platform* by using platform service textitservice discovery. The *context acquisition* component executes platform service *service discovery* via *communication* components. The *communication* components receive the semantic service descriptions from the *service platform* and forward them to *context acquisition* component. It parses semantic service descriptions by using the different parsing techniques.

The Knowledge experts use *context parser* component to extract context from the structured data, e.g. non-functional parameters of service descriptions, but they need Natural Language Processing (NLP) techniques to extract context from unstructured data, e.g. textual description of services, of the service description. The *NLP* component allows the knowledge experts to extract context from textual description of service by using NLP-techniques, e.g. POS-tagging, reduction of stop words, stemming, and lexical analysis, as discussed in Chapter 4. The result of *context parser* and *NLP* component is a context matrix as described in Chapter 4. After the context acquisition, the *context acquisition* component merges both context matrixes extracted by components *context parser* and *NLP* and stores resulting context matrix in the repository *context data*.

By the acquisition of context related to non-functional parameters of services, e.g. price plan, legal aspects, and SLAs, the context is very complex. For the visual semantic analysis process, the knowledge experts transform complex context information into suitable context information by using a *conceptual scaling* component, described in Chapter 4. The Knowledge experts define conceptual scaling and perform visual semantic analysis process. They can also redefine conceptual scaling during the visual semantic analysis process, if it leads to improve the results.

5.1.3 Perspectives

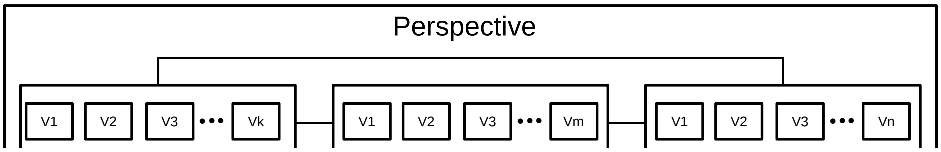
The conceptual framework of VSA offers the knowledge experts multiple *perspectives* to analyze different aspects of service descriptions such as price plan, legal aspects and SLAs. The multiple *perspectives* allow the knowledge experts to extract a sub collection of attributes from service description, e.g. price plan related attributes, to analyze the price plans. The Knowledge experts can create different *perspectives* by choosing sub collection of attributes from the service description. All perspectives offer complete functionality of visual semantic analysis process.

The Knowledge experts load context or/and semantic information to create the initial *perspective*. The context and semantic information serve as data model for the initial *perspective*. Afterwards, the knowledge experts can either perform visual semantic analysis in the initial *perspective* by using the whole context or extract partial context to create new *perspective*. For example, they can extract price plan related context information from the service description to analyze price plans. The extracted partial context serves as data model for the new *perspective*. The knowledge experts can also merge different context within one *perspective*. The *perspective controller* manages the *perspectives* and assigns the data models to different *perspectives*.

Each *perspective* has multiple categories of *views* such as *visual context analysis view*, *visual semantic refinement view* and *visual SD modeling view* as shown in Figure 5.5. They perform the tasks of the sub processes visual context analysis, visual semantic refinement and visual SD modeling of the VSA process model. The knowledge experts can analyze context information and extracted semantics in *visual context analysis view* and *visual semantic refinement view* correspondingly. The *visual SD modeling view* allows them to model new services on the base of the extracted semantics.

Each view of *perspectives* offers multiple *internal views* depending on data formats and visual variables as shown in Figure 5.5. Different *internal views* are available in the *visual context analysis view* for the analysis of the visual context analysis. The knowledge experts can add or remove the *views* in the *perspectives*. The *views controller* manages the *views* and their *internal views*.

The feedback loops between the sub processes visual context analysis, semantic extraction, visual semantic refinement and visual SD modeling in the VSA process model as shown in Figure 5.6 requires synchronization of data.



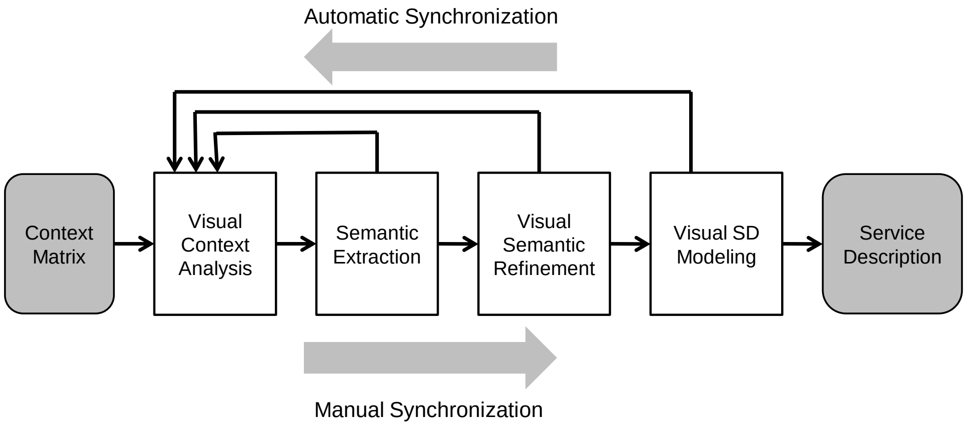


Figure 5.6: The synchronization of view in a perspective

The automatic synchronization takes place between opened *views* in the *perspectives*. For example, the changes in the extracted semantics, expert's knowledge and SD-Modeling induce changes in the context information. The synchronization between opened views occurs automatically. The changes in the context information have impact on the extracted semantics. The semantic extraction algorithms have to be applied to extract semantic after the changes occur in the context. The application of semantic extraction algorithms can be very time intensive depending on the amount of the context. In the case of automatic synchronization, every change in the context could lead to application of the time intensive semantic extraction process. Therefore, the knowledge experts can first perform all changes in the context and then synchronize these changes with semantics by applying semantic extraction techniques from the *semantic analysis algorithms library* manually. The collaborative VSA and synchronization of context, extracted semantics, and modeled SDs between different *perspectives* are not considered in this thesis and presented as future research fields in Chapter 8.

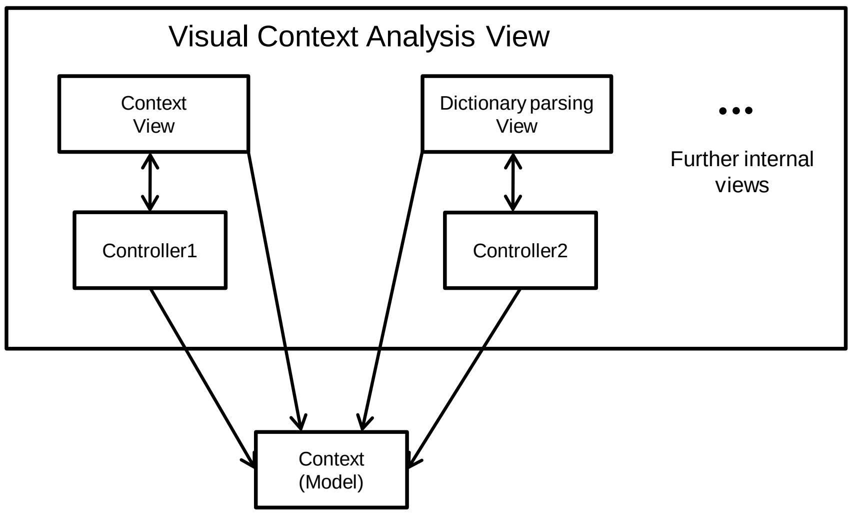


Figure 5.7: The visual context analysis view

5.1.4 Visual Context Analysis View

The *visual context analysis view* allows the knowledge experts to analyze the context information extracted from service descriptions. It contains further *internal views* by using different visualization techniques like *context view* or *dictionary parsing* view as shown in Figure 5.7. Every visualization technique has its own strengths and weaknesses. The choice of appropriate visualization technique depends on the data format and task at hand. The knowledge experts use different visualization techniques for *internal views* depending on data format and task. The *perspective controller* assigns context to *perspectives* as data model. The *internal views* of the *visual context analysis view* use this context as data model as shown in Figure 5.7. The *internal views* have their own *controller* to mange user's interaction.

The interactive visualization techniques support knowledge experts to analyze context information. The visualization techniques in the *semantic context analysis view* follow the basic guidelines of information visual seeking mantra proposed by Shneiderman [Shn96]: Overview, zoom, filter and detail on demand. It allows the knowledge experts to explore context and focus on the important parts of context. The interactive visualization techniques allow them to interact with context information to extract partial context to generate new *perspectives* and analyze context further.

The knowledge experts can enrich the context information assigned to the corresponding *perspective* by editing them. For example, they can add, remove or modify services and attributes of the context information manually.

The modification of the context is considered as expert's knowledge. For this purpose, the context is divided into two categories context acquired automatically by the *context acquisition* component and context generated manually by the knowledge experts. The context enrichment in the *visual context analysis view* changes the context (data model) of the corresponding *perspective*.

The import of external context allows the knowledge experts to merge external context in the context available in the *visual context analysis view*. The *perspective controller* supports the import of external context from the repository *context data* or other file formats. The external context can be merged in the assigned context (data model) of the *perspective* directly. The changes in the data model are propagated to all *internal views* of *visual context analysis view* to visualize merged context to the knowledge experts.

The application of different semantic analysis techniques, e.g. clustering and Dictionary parsing (see Chapter 2), is offered in the *visual context analysis view* to reduce or enrich the context. These semantic analysis techniques allow the knowledge experts to identify similar services or attributes of the services. The *perspective controller* provides different semantic analysis techniques from *semantic analysis algorithms library* in the *visual context analysis view*.

After the context analysis, the knowledge experts can either export the whole refined context or partial context by selecting parts of extracted context to the repository *context data*. The *perspective controller* manages the context export functionality. The knowledge experts can use the exported context again later in another session to analyze it further.

5.1.5 Visual Semantic Refinement View

The knowledge experts use refined context to extract semantics from it. For this purpose, they can use different semantic extraction techniques from the *semantic analysis algorithm library*, e.g. FCA, to extract semantics from the refined context. The *perspective controller* makes possible for the knowledge experts to use semantic analysis algorithms from the *semantic analysis algorithms library* for the semantic extraction.

The *internal views* with different visualization techniques are also offered in the *visual semantic refinement view*, e.g. *semantics view* and *expert's knowledge view*, as shown in Figure 5.8. They facilitate the knowledge experts to explore and interact with semantics to validate and refine it. In the case of large amount of extracted semantic, the usability issues arise. The *internal views* follow the visual information seeking mantra [Shn96] to handle the usability issues. These aspects allow the knowledge experts to navigate through semantics and interact with the semantics to select interesting parts

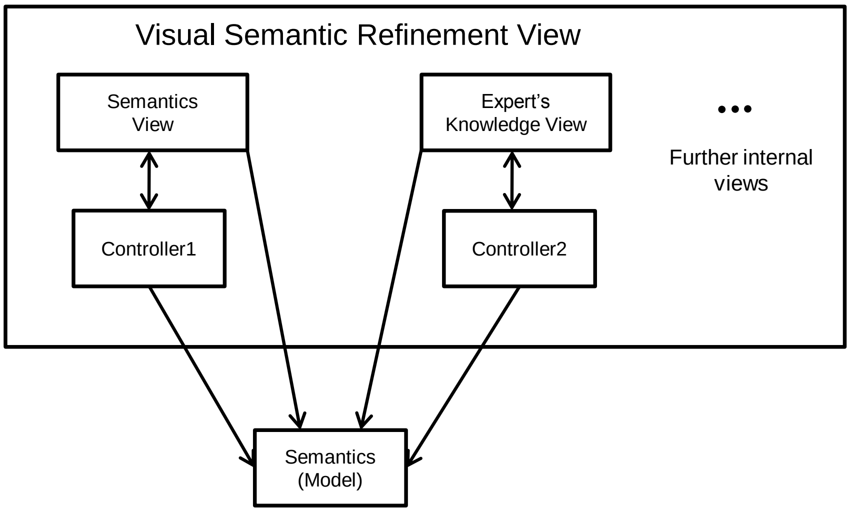


Figure 5.8: The visual semantic refinement view

of semantic according to their interests. The *controller* of the *internal views* takes care of the interaction of the knowledge experts as shown in Figure 5.8.

The navigation through semantics may lead to cognition overload for the knowledge experts [KAF+08, Bha08]. The aspects like keeping track of the interaction history of the knowledge experts may reduce the cognitive load for the knowledge experts, e.g. keeping track of the navigation path of the knowledge experts. The functionalities like navigation, filtering, and history go hand-in-hand with the chosen layout of the visualization, which implements the positioning and visibility of the semantic information within *internal views* of the *visuals semantic analysis view*. The *perspective controller* allows the knowledge experts to use different visualization techniques for the *internal views* of *visual semantic refinement view*.

The validation of the extracted semantics by adding and removing concepts, and semantic relations. The *clipboard* is an *internal view* of the *visual semantic refinement view*, where the validated semantic is collected and visualized. The semantics in the *clipboard* does not affect the data model (extracted semantic) of the *visual semantic refinement view*. The *internal view* of the *visual semantic refinement view knowledge expert's view* allows the knowledge experts to model their own domain knowledge or import external knowledge.

The changes in the extracted semantics and manual modeling of domain knowledge require synchronization of context information and merging of extracted semantic and manually modeled semantics. The synchronization

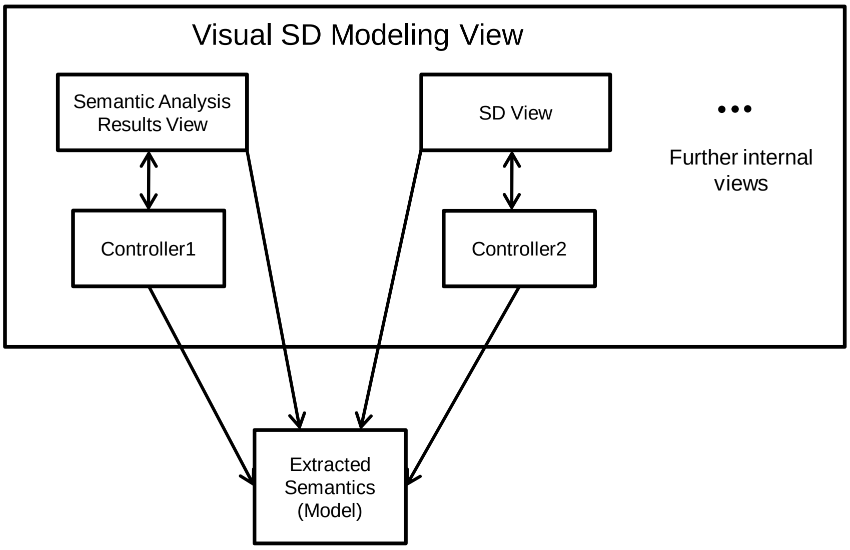


Figure 5.9: The visual SD modeling view

between *visual context analysis view* and *visual semantic refinement view* is done automatically. After synchronization of context information and semantics, the merging manual modeled semantics and extracted semantics is offered by using merging technique, e.g. FCA, from the *semantic analysis algorithms library.*

After the semantic analysis, the *visual context analysis view* offers the export of the extracted semantics to the repository *semantics*. The *perspective controller* manages the export of semantics in the repository *semantic* and supports different formats like OWL or RDF(S) for the export.

5.1.6 Visual SD Modeling View

After the semantic analysis of services, the knowledge experts model the service descriptions on the base of the extracted semantics in the *visual SD modeling view*. It offers different visualization techniques for its internal views, e.g. *semantic analysis results view* and *SD modeling view*, as shown in Figure 5.9 to model services. The combination of different visualization techniques makes it easier for the knowledge experts to identify services in the extracted semantics that are similar to their planned new services. It facilitates the knowledge experts also to model new services by reusing and adapting the extracted semantics.

After the analysis of available services and modeling of new services, the feedback loop between the *visual context analysis view* and *visual SD modeling view* allows the reanalysis of services to compare new modeled services with the available services. In this way, the knowledge experts perform analysis, modeling and reanalysis of services in an interactive manner. They can iterate this process until they are satisfied with the new modeled services.

After the modeling of new services, the knowledge experts have to send the new modeled services for the validation before they can be registered on the *service platform*. The *governance* service of the *service platform* allows the validation of the new modeled services. The results of the validation process are visualized in the *visual SD modeling view*. For example, the errors in the new modeled service description are marked in the *visual SD modeling view*. After the successful validation process, the knowledge experts register the new modeled services on the *service platform* by using the *service registry* service of the *service platform*. The *visual SD modeling view* accesses these services *governance* and *service registry* via *perspectives controller* and *communication* components.

5.1.7 Visual Layout Algorithms Library

According to Mackinlay [Mac86], different visualization techniques with specific visual variables may suite for different type of data formats. Therefore, different visualization techniques are available in the *visual layout algorithms library* of the *conceptual framework of the visual semantic analysis*. The *views* allow the knowledge experts to visualize different data types, e.g. *context data*, *semantics*, and service descriptions, in the *internal views* by using these visualization techniques according to their own needs. All new developed visual layout algorithms should also be a part of the *visual layout algorithms library*.

5.1.8 Semantic Analysis Algorithms Libraries

The *conceptual framework of the visual semantic analysis* offers a bundle of semantic extraction algorithms, e.g. clustering and Formal Concept Analysis (FCA), to the knowledge experts. Different semantic analysis algorithms cannot provide best results depending on the context information, data types, or amount of context information. The combination of different semantic analysis can improve the results of the semantic analysis process. Therefore, the knowledge experts use either one semantic analysis algorithm or combination of them from the *semantic analysis algorithms library* for the visual semantic analysis process.

5.1.9 Context Data

The context in *visual context analysis view* possesses two types of context automatic acquired context from services and manually generated context by the knowledge experts. The knowledge experts employ different semantic analysis algorithms from the *semantic analysis algorithms library* to refine context or extract semantics from the context. In the visual semantic analysis process, the perspectives based VSA allows the knowledge experts to create new *perspectives* by using a partial context. In the new *perpectives*, they continue the visual semantic analysis process for just partial context. After the visual context analysis, they save the context from different *perspectives* in the repository *context data*. The integration of the saved context in new *perspectives* is supported by the import of context. In this way, the VSA is possible in different sessions.

5.1.10 Semantics

After the extraction of semantics from the context, the validation and refinement of extracted semantics takes place in the *visual semantic refinement view*. The knowledge experts add or remove concepts and semantic relations to refine the extracted semantics. The *internal views* of the *visual semantic refinement view* allow the knowledge experts to model their own domain specific semantics. The perspectives based visual semantic analysis generates also different semantic information in the different perspectives. The export of the manual generated semantics and extracted semantics allow knowledge experts to save it in the repository *semantics*. The import of different semantic information enables the knowledge experts to integrate semantics extracted in different *perspective* sessions.

5.1.11 Communication

The *communication* components serve as external interface to communicate with the *service platform* by using the platform services. For the VSA, the *communication* components allow the import of service descriptions from the *service platform* by using platform service *service discovery*.

After the analysis, modeling and reanalysis of service, the knowledge experts have to send the new modeled service to the *service platform*. The *communication* component use the *governance* service to send new modeled services for the validation. It forwards the validation results to *visual SD modeling view* via *perspectives controller*. The *visual SD modeling view* visualize the validation results to the knowledge experts. After the successful validation process, the knowledge experts register new modeled service by using *service registry* service of the *service platform* via *communication* components. In this way, the whole communication between the *VSA system* and *service platforms* takes place via *communication* components.

5.2 Service Platform

The *service platform* provides a bundle of platform services to facilitate the interaction between the Visual Semantic Analysis and *service platform*. The main platform services used by the *visual semantic analysis system* are described in the following section.

5.2.1 Platform Services

The access to *service platform* is controlled and monitored by the *access gate* sevice. It checks the user's authentication and detects the success or failure of each access request. In addition to authentication, it checks also if users have valid contract for the specific services. The VSA system has to login on the *service platform* via *access gate* service before it can import the services from the *service platform*.

The search for services and import of service from the service platform is offered by the platform service *service discovery*. The knowledge experts have to import services for the visual semantic analysis in order. The extracted semantics from the imported service support the knowledge experts to model new services and reanalyze new modeled services in order to compare them with the imported services.

The platform service *monitoring* collects all monitoring information about Quality of Service (QoS) parameters of services, e.g. availability and response time. The VSA system uses the SLAs specific QoS parameters from the SDs for the SLAs analysis. The values of these QoS parameters are specified as nominal values in the SLAs. They don't represent actual value of these QoS parameters. The *monitoring* service of the *service platform* have the actual values of these QoS parameters. The knowledge expert have to import the actual values of these QoS parameters by using *monitoring* service in order to perform the SLAs analysis on the base of the actual values of the QoS parameters.

The *governance* service assures compliance of the new modeled services by validating modeling guidelines. It validates the service schema of the new modeled services. After the modeling of new services with the VSA system, the knowledge experts have to send new modeled services for the validation before they can be published on the service platform. The *governance* service of the platform allow the knowledge experts to validate new modeled services.

After the successful validation process, the knowledge experts register the new modeled services on the *service platform* by using the *service registry* service of the *service platform*. The communication components use the platform services to establish an interaction between the VSA system and the service platform.

5.3 Summary

A *conceptual framework for the visual semantic analysis* is presented in this chapter. The components of the *VSA conceptual framework* are described in detail, which supports the knowledge experts to accomplish different tasks of the visual semantic analysis process model, e.g. context acquisition, context analysis, semantic analysis, and modeling of new services. Furthermore, the interaction between *visual semantic analysis system* and *service platform* by using different platform services, e.g. *service discovery*, *monitoring*, *governance*, and *service registry* is described in detail.

A prototypical implementation of the *conceptual framework for the visual semantic analysis* is the focus of the next chapter. The implementation of different components of the *conceptual framework* will be described in detail. Furthermore, application scenarios, e.g. price plans analysis and Service Level Agreements analysis, will also be presented to demonstrate the application of visual semantic analysis to support *semi-automatic* modeling of the service descriptions.

Chapter 6

A Prototype for the Visual Semantic Analysis

The Visual Semantic Analysis (VSA) process model and a conceptual framework for the VSA approach were presented in the previous chapters. A VSA prototype developed on the base of this VSA process model and conceptual framework will be presented in this chapter. The section "applied technologies" gives an overview of the used technology for the VSA prototype. The service platform, VSA system and Web client present different aspects of the VSA prototype. The service platform provides all services that are necessary for the Visual Semantic Analysis, e.g. service discovery and monitoring. The VSA system "Sophie" presents the aspects such as analysis, modeling and reanalysis of services. The Web client "Service Browser" introduces semantic visualization techniques to offer a user friendly interface for services and their semantic descriptions.

Furthermore, the application of the VSA approach in the service lifecycle is presented. For this purpose, the usage of the VSA prototype for the four application scenarios "service matchmaking and annotation", "price plan analysis", "legal aspects analysis" and "Service Level Agreements (SLAs) analysis" is demonstrated. These application scenarios present how the VSA approach can support knowledge experts during the modeling of the service descriptions.

6.1 Applied technologies

The technologies used for the realization of the Visual Semantic Analysis (VSA) can be divided into three parts: *Service Platform*, VSA system *Sophie*, and Web client *Service Browser*. The *Service Platform* hosts ser-

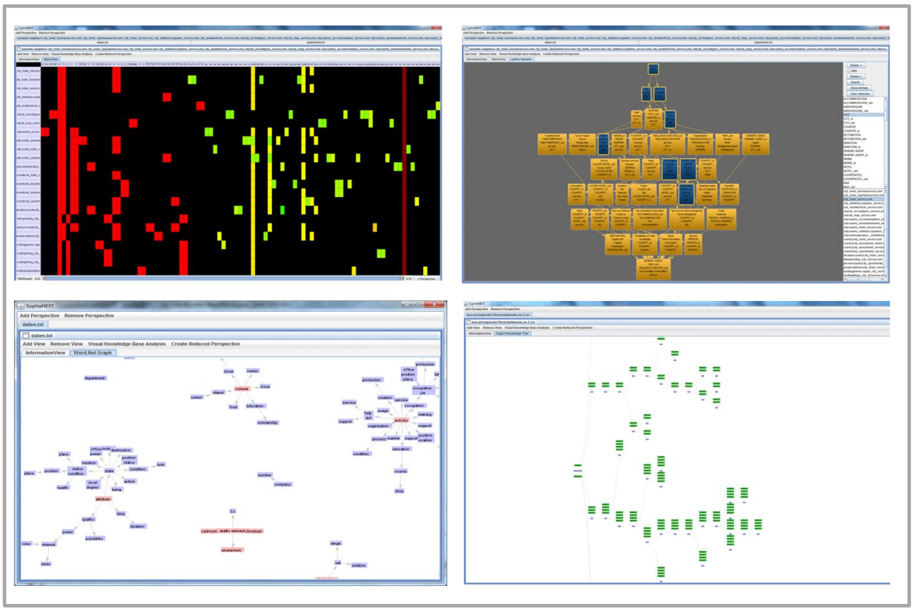


Figure 6.1: The Graphical User Interface of the VSA system *Sophie*

vices and provides service description and monitoring information for the Visual Semantic Analysis. The VSA system *Sophie* allows analysis and semi-automatic modeling of services. The *Service Browser* supports users to explore the results of the semantic analysis process and semantics of services.

The *Service Platform* [TEX10] supports the concept of Web Service Ecosystem and host services, which contains service descriptions with the aspects, e.g. service descriptions, service annotation, price plan, SLAs and legal aspects. Therefore, it suites well for the realization of the VSA prototype. It provides platform services such as service *discovery* and *monitoring* to import service descriptions and monitoring information into the *Sophie*. After the modeling of a service, *Sophie* register new modeled services on the *Service Platform* by using the services *governance* and *service registry*. The *ConWeaver* tool [Con10] is used for the Natural Language Processing (NLP) to extract context information from the service descriptions. It supports also the conceptual scaling of complex context information as it is discussed in section 4.4.1.

The VSA system *Sophie* realizes the VSA process model, which extends the reference process model of information visualization proposed by Card et al. [CMS99] discussed in section 2.2.3. The *Prefuse* visualization toolkit [Pre10] has been developed as a realization of the reference process model proposed by Card et al., and assists software developers to realize interactive infor-

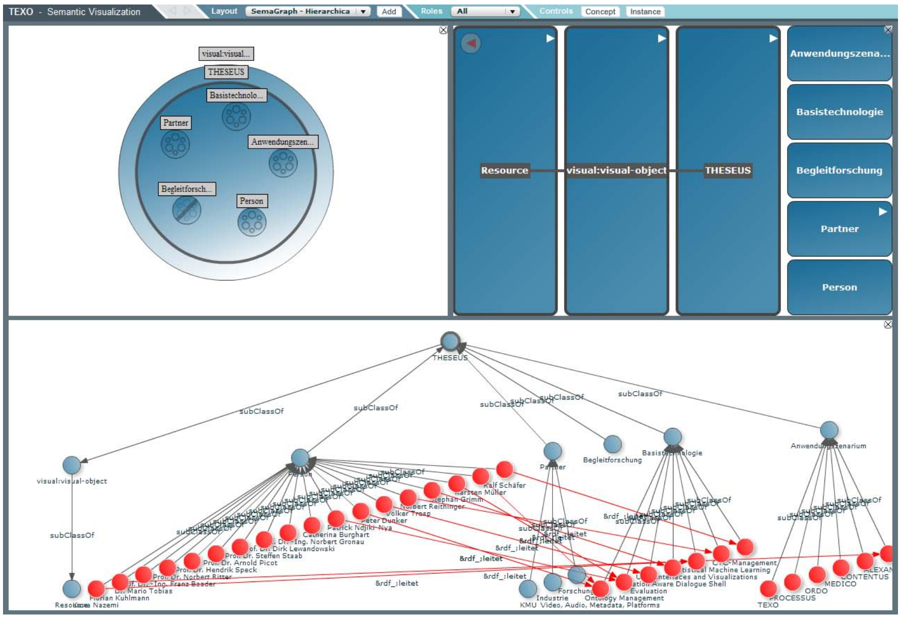


Figure 6.2: The Graphical User Interface of the *Service Browser*

mation visualization applications using the Java programming language. It provides a recommendation and standard implementation of graph and tree data structures, abstract classes for assigning color schemes and rendering of the visualization, layout algorithms and interfaces for graphics-user-interactions [Pre10], which are especially useful for our purposes. That is why the *Prefuse* visualization toolkit is used to develop the VSA system *Sophie*. In addition to the *Prefuse* standard functionality, the complementary functionality for the realization of the VSA process model is realized for *Sophie*. Figure 6.1 shows the graphical user interface of the VSA system *Sophie*.

Users can explore the results of the VSA and semantics of services by using the client *Service Browser* [BW09]. As the development environment for the *Service Browser* Adobe Flex was chosen. Applications that are developed with the Adobe Flex open source framework tool can be executed by the Adobe Flash Player. The *Service Browser* can be executed by all major web browsers with installed Adobe Flash Player. According to a survey conducted on March 2010 [Fla10], flash player is installed on 99% of all computer devices with internet access.

The Service Bowser is developed on the base of the Semantic Visualization Framework (SemaVis) [BW09, NBH09, NBB+10]. It is available as an Adobe Flex library and offers interactive visualization techniques for the semantic visualization. It loads and manages semantic information and provides a standard interface for the visualization modules. The modular design of the SemaVis framework allows an easy integration of additional visualization modules in the framework. The graphical user interface of the *Service Browser* is depicted in Figure 6.2. The VSA system *Sophie* and *Service Browser* are described in the following sections.

6.2 Sophie: VSA system

The VSA system *Sophie* facilitates knowledge experts to perform semantic analysis and modeling of services according to the VSA process model. It allows users to consider different aspects of SDs, e.g. price plans, SLA analysis and functionality, separately or combination of these aspects together. Knowledge experts can analyze, model, and reanalyze these aspects in different perspectives individually or together.

In this way, the VSA system *Sophie* allows to deal with the complexity of SDs and a large number of services. The packages and classes of the perspective based VSA are tare shown in the class diagram (see Figure 6.3). The main packages of the architecture are Perspectives, Views, semantic analysis algorithms, and data model. These packages with their classes are described in the following sections.

6.2.1 Perspectives

The package perspectives contains the classes and interfaces *Perspective Controller*, *Perspective*, *Perspective Interface*, and *View*. *Perspective Manager* and *Perspective* mange the generation and removal of perspectives to support perspectives based VSA as discussed in Chapter 5.

Perspective Controller

The *Perspective Manager* extends the Swing class *JFrame* and represents the main window of the VSA system *Sophie*. It contains 0 to n *Perspective Interfaces* for the main window. It allows knowledge experts to create or delete perspectives as described in section 5.1. They are organized as tabs in the main window of the *Sophie*. It assigns context to different perspectives by creating new perspectives in the main window of *Sophie*. It acts as an interaction bridge for the perspectives in providing them with external libraries like semantic analysis algorithm.

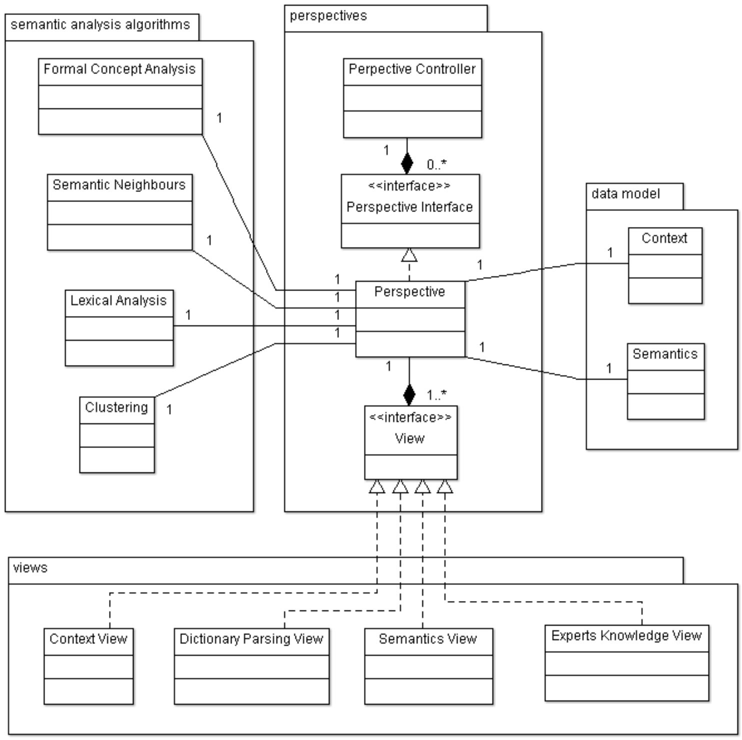


Figure 6.3: The class diagram of the VSA system *Sophie*

Perspective

Each instance of *Perspective* is the one perspective in *Sophie* and represented as a tab. Knowledge experts can create perspective instances by using *context data* or by selecting partial context from one opened perspective. Every perspective instance also serves as a bridge for their views to spawn additional new perspectives. Perspective uses the *Context* and *Semantics* to store context and semantic information and handles the creation and removal of views on the perspective level. As it creates the views, the *Perspective* forwards the context and/or semantics as data objects to the views, so they can use that information to visualize them. It also assigns the views *visual context analysis view*, *visual semantic refinement view* and *visual SD modeling view* as described in section 5.1.

Perspective also handles the synchronization between the context and semantic information. The changes in semantics are always forwarded directly to the context and these changes are also immediately visible in the views. The re-analysis of changed context information by using semantic analysis techniques, e.g. FCA can be re-executed manually. The manual execution ensures that the time consuming semantic analysis algorithms are used only, when it is needed. The manual execution of semantic analysis algorithms is also handled by *Perspectives*.

6.2.2 Views

The internal views of the views *visual context analysis view*, *visual semantic refinement view* and *visual SD modeling view* as decribed in section 5.1 are part of the package Views. These views allow to accomplish the task of sub processes visual semantic context, visual semantic refinement and visual SD modeling view as discussed in Chapter 5.

Context View

Context View implements the *View* interface. Each view has to implement the View interface to make sure that the related perspective can interact properly with it. It organizes the associated *context data* and visualizes the entries in the matrix by utilizing the *Prefuse* visualization toolkit. *Context View* is registered as an observer to the *Context* instance that holds the context information. The synchronization between the context data and the *Prefuse* data tables is also managed here.

Figure 6.1 (upper left view) shows the matrix view of *Sophie*. The rows and columns in the matrix represent services and attributes correspondingly. The relation between services and attributes is represented by matrix entries. The panning and zoom functionalities facilitate users to navigate through the matrix visualization. The axis adjustment and highlighting of rows and columns support the navigation during the usage of these functionalities.

The dialog windows are offered in the *Context View* to specify parameters of semantic analysis algorithms such as clustering and semantic neighbors. User can use these clustering algorithms to cluster similar services or attributes. After the application of these algorithms, the clustering results are visualized in the matrix view. In the same way, users can apply semantic neighbors algorithm to find services related to a specific context. After the application of semantic neighbor algorithms, the results are also visualized in the matrix view. Users can also select the results of clustering or semantic neighbors search to create new perspective in order to continue with the semantic analysis. They can also manually select parts of the matrix to create new perspectives.

Dictionary Parsing View

Dictionary Parsing View is another view that utilizes the context information and uses the WordNet database to enrich or reduce the context information. By using the WordNet functionality, the synonyms, hypernyms and hyponyms of attributes contained in the context information can be identified and visualized in the *Dictionary Parsing View*. Each group in this view represents a word-group according to lexical relations. The words are connected to their hypernyms and hyponyms.

Users can have an overview of lexical relationship between attributes contained in the context information as depicted in Figure 6.1 (bottom left view). They can reduce context information by clustering attributes that are synonyms. They can also replace the attributes with their hypernyms to have just more general attributes in the context information. By using synonyms and hypernyms functionality of the WordNet, users can reduce the dimensionality of the context matrix. Users can also enrich the context information by adding subordinate of attributes to the context information by using hyponyms functionality of the WordNet. These functionalities are available in the context menu of the *Dictionary Parsing View*.

Semantics View

The semantics extracted from context information in the semantic analysis process is visualized by the *Semantics View*. The basic layout functionality is available in the *Semantics View* as shown in Figure 6.1 (upper right view) . The *Semantics View* offers semantic information as lattice visualization, where general concepts are placed higher than the specific one. It allows users to start with a concept as starting point and visualize all concepts, which are related to the selected concepts.

This view provides users the panning, zoom and filter functionality to explore semantics. The interactive navigation with dynamic lattice allows users to have a better overview of semantics. The filter functionality supports users to highlight parts of the lattice to see the parts of lattice according to their needs. For the validation, users can select parts of the lattice and add them to a *clipboard*, which contains the validated semantic information. Additionally, user can prune the extracted semantics by deleting or merging concepts of lattice.

Experts Knowledge View

The *Experts Knowledge View* as shown in Figure 6.1 (bottom right view) allows knowledge experts to model their own knowledge manually. They can model their own knowledge by adding or removing concepts, attributes, relations and instances. It facilitates knowledge experts to model the missing semantic information in the extracted semantics. For example, knowledge experts can model special service features during the price plan analysis, if these features are missing in the results of price plan analysis.

Knowledge experts can also import external semantics, e.g. available semantics from the web, and modify it within the *Experts Knowledge View*. They can merge both the manually generated and the extracted semantic by using the merging techniques such as Formal Concept Analysis (FCA). For this purpose, the manually generated semantics have to be synchronized with the context matrix. The synchronization between both views *Experts Knowledge View* and *Context View* is managed by the corresponding Perspective.

6.2.3 Semantic Analysis Algorithms

The VSA offers a combination of semantic analysis and visualization techniques to support semi-automatic modeling of SDs as described in Chapter 4. The package semantic analysis algorithms contain all semantic analysis techniques offered for the VSA system.

6.2.4 Formal Concept Analysis

The semantic analysis algorithm Formal Concept Analysis (FCA) allows users to extract semantics from the context information. It can also be used to merge different semantic information. For example, FCA allows merging of manual and automatic generated semantic information. The FCA-algorithm Colibri [Lin10] is offered in *Sophie* for the semantic analysis. It is a Java software package and offers the fast Formal Concept Analysis [Lin10].

The Colibri tool supports the import of the XML and CON file formats [Lin10]. For the integration of the Colibri software package in the *Sophie*, it is modified to import the context information directly from the *MartixView*. Knowledge experts can analyze the context information by using the *Context View* and extract semantics from the context information by using Colibri. For the semantic merging of manual and automatic generated semantic, the synchronization of the *Experts Knowledge View* and *Context View* is necessary. After the synchronization, the semantics can be merged by using the Colibri.

6.2.5 Semantic Neighbors

The algorithm semantic neighbors supports users to find documents that have similar attributes as the selected service. The two parameters of the algorithm down-tolerance and up-tolerance allow users to decide how similar the documents should be. The down-tolerance lets users decide how many attributes of the selected document can be missing in the neighboring documents. Users can specify with the up-tolerance how many additional attributes of the targeted neighboring documents should be fulfilled, which are not fulfilled by the selected document.

The semantic analysis of a huge context matrix can lead to vast semantic information and lattice visualization. By using semantic neighbors functionality, users can perform an iterative analysis of the context information. For example, knowledge experts don't have to analyze all the services related to car repair shops at once, when they just want to analyze car rental services. They can first find semantic neighbors of the car rental services and then perform the semantic analysis.

6.2.6 Lexical Analysis

The pattern matching heuristic allows extraction of lexical entries from morphological processing of text to identify semantic relations. For example, a simple heuristic relates first noun phrase "Deutsche Bank" in the sentence "The Deutsche Bank is the biggest bank in Germany". The lexical analysis allows users to define such type of heuristic to extract semantic relation. The n-gram model can be used as discussed in Chapter 2 for the pattern matching heuristic.

The network of semantic relations between English words WordNet allows the identification of synonyms, hyponyms and hypernyms. These semantic relations play an important role for the lexical analysis. It helps users also to reduce the context information by mapping specific terms to more general terms. For example, the terms car and truck can be mapped to the term vehicle to reduce the context information.

6.2.7 Clustering

Clustering algorithm is used to group similar attributes or documents together in order to reduce the dimensionality of the matrix. The clustering algorithm K-Means++ is provided in the *Sophie* prototype to cluster attributes or documents. It is the modification of the K-Means algorithm, which has been already described in Chapter 2. The K-Means algorithm can only find local minimum to assign centroids. The K-Means++ initializes the centroids so that (a) each centroid is placed directly on a coordinate of a document and (b) every additional centroid is placed with a linearly higher probability further away from the already placed centroids. The centroid serves as the center of a cluster. Obviously this algorithm increases the initialization time, but it lets the main algorithm terminate faster.

After the initialization, each service or attribute is assigned to a nearest cluster, then centroid is recalculated by building the central point of all the coordinates of the contained documents or attributes. Then the services or attributes are assigned again to the nearest cluster. These steps are iterated, until no more services or attributes are being moved between clusters.

6.2.8 Data Model

The package data model contains *context* and *semantic* processed and generated in the VSA process. They are assigned to each perspective to visualize context and semantics in views of the perspective.

Context

The class *Context* incorporates the context data. The context data is represented as table, where the rows and columns represent services and attributes correspondingly. The entries represent which service owns the attributes from the columns. This class stores the service- and attribute-name of the entry and associates it with the value. In most cases it is the *tfidf* value (see section 2.2), but it can also be the exact price of a price plan. For efficiency reasons, the entries themselves can be accessed through both service-names and attribute-names, as different algorithms need to access all entries with same service-name or all entries with same attribute-name very often.

Additionally every *Context* class handles clustering data for the rows and columns of matrix and stores it. It allows efficient access to the row or column clusters which can be queried with service- or attribute-name. This clustering information allows user to reduce the dimension of matrix or enrich it by adding new rows and columns. The class *Context* handles all input/output operations in regard to the context data such as *load*, *merge* and *save* context information for the perspectives.

Semantics

The management of semantic information in the VSA system *Sophie* is done by the class *Semantics*. The semantic information is available as a graph, which contains concepts, instances, attributes and relations between instances. It supports the functionality like manipulation, search and comparison of semantic information, which help users to perform semantic analysis. Users can add or delete concepts, instances, properties and relations in the semantics during the validation of automatically extracted semantics.

The changes in the extracted semantics require the synchronization between extracted semantic and context information. The class *Semantics* transforms the semantic information into *context data* in order to make the synchronization possible. It allows also users to load or store automatically extracted or manually generated semantics.

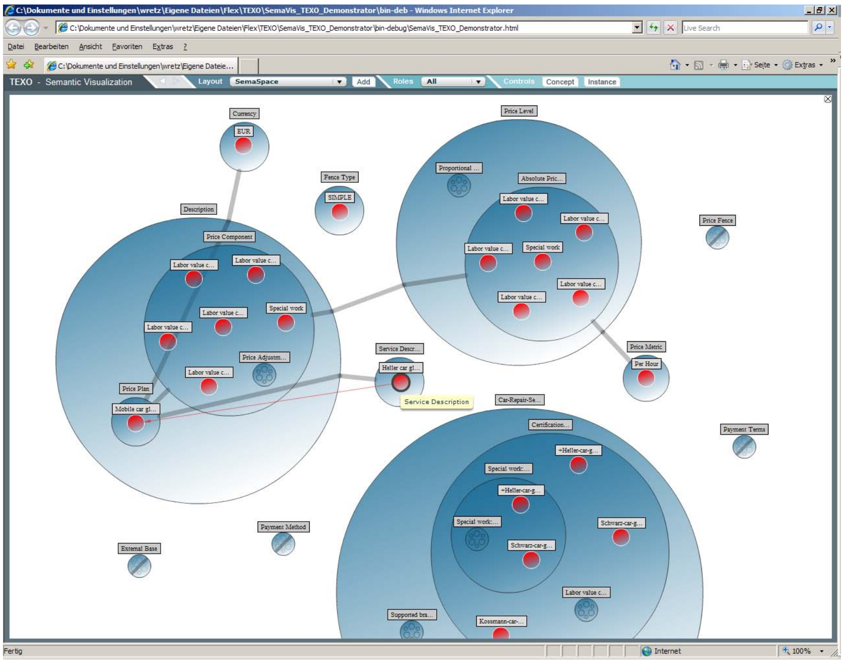


Figure 6.4: *SemaSpace* view of *Service Browser*

6.3 Service Browser

The results of the semantic analysis process and the semantic information of modeled services are visualized with the *Service Browser*. Users can navigate through the semantic information to have an overview of the extracted semantics and the data model of the modeled service, e.g. USDL-schema or Service Ontology. They can also edit the modeled services with the *Service Browser* under the consideration of extracted semantics and the data model of the modeled services.

The *Service Browser* is developed on the base of the *SemaVis framework* [NBB+10]. The functionality such as semantic visualization, editing and annotation are offered by the SemaVis framework. The modular design of the SemaVis framework allows developing interactive semantic visualization applications by using existing functionality or visualization techniques of the SemaVis framework. The layer based concept allows software developers to use different layout techniques for the semantic visualization and adapt the look & feel of the visualization on the presentation layer [NBB+10].

The graphical user interface of the *Service Browser* is depicted in Figure 6.2. It allows users to compose a display by using different views. The changes occurring in one view by user interaction are always synchronized with other views. This approach allows users to combine different visualizations to understand the semantic information better. For example, users can see hierarchical information with a TreeMap and semantic relations with a graph.

The role based visualization provides users semantic information according to their roles, which are specified by the *Service Platform*. For example, the price plan experts can see semantic information related to pricing and functionality of a service. The semantic information related to legal aspects is not visualized to the price plan experts. This approach helps users to focus on their task and reduce cognitive load for them. Users can visualize their role specific semantic information and edit the modeled services.

A view *SemaSpace* [Bha08, BW09] is developed for the *Service Browser* according to standard interface of the SemaVis framework. It is a multilevel and multi branch tree-based semantic browsing view that offers hierarchical information and semantic relations between the individual concepts and instances at the same time as depicted in Figure 6.4.

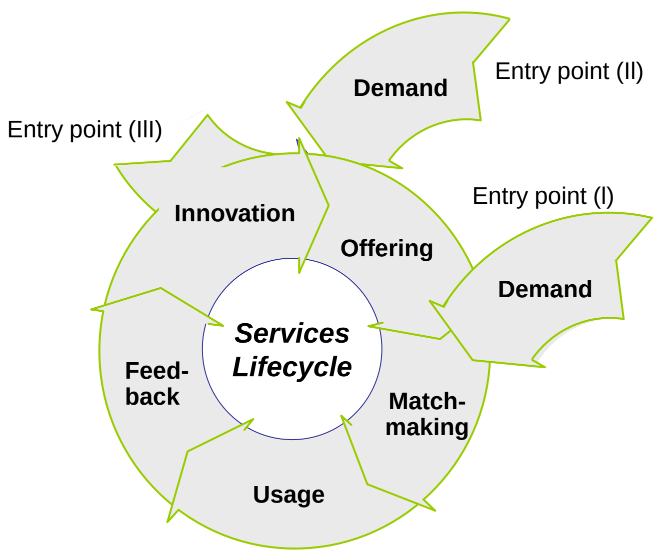


Figure 6.5: Service Life Cycle

Service lifecycle phases	Tasks	Roles of the service lifecycle
Innovation	Service ideas Analysis	Service provider, service broker, service mediator and service consumer
Offering	**(i) Service matchmaking and annotation, (ii) Legal aspects analysis, (iii) price plans analysis and (iv) SLAs analysis**	**Service provider and Service broker**
Matchmaking	Service matchmaking	Service consumer
Usage	Service integration	Service provider, service broker and service consumer
Feedback	Feedback analysis	Service provider, service broker and service consumer

Table 6.1: The application of VSA approach in the service lifecycle. In this thesis we focus on offering phase

Service providers, brokers, mediators and consumers can analyze data available on the Web, e.g. news, blogs or tweets, to generate new service ideas in the innovation phase[OBB+09]. They can also analyze existing ideas to compare the new ideas with existing one. Service providers and brokers can analyze textual service description of services to facilitate service matchmaking and annotation of the new services in the offering phase. Service providers and brokers can also analyze legal aspects, price plans and SLAs of existing services to model these aspects of new services or adapt these aspects of existing services. Service consumers can use VAS approach in the matchmaking phase to find services according to their own needs.

In the usage phase, service providers, brokers and consumers can analyze the technical specifications (functional attributes) of services and customers' applications to support semi-automatic integration of services in the customers' applications. The technical specification should be available to support semi-automatic service integration. Service providers and brokers can analyze implicit and explicit feedback of service consumers to improve the quality of service of their offered services in the feedback phase. Service consumers can analyze implicit and explicit feedback of other service consumers to get information about the quality of services offered by different service providers.

In this thesis we focus on offering phase. The four tasks “service matchmaking and annotation", "legal aspects analysis", "price plans analysis" and "SLAs analysis", which take place in the offering phase, are presented in the following sections briefly.

6.5 Application Scenarios

The usage of VSA system for four application scenarios "service matchmaking and annotation", "price plan analysis", "legal aspects analysis" and "Service Level Agreements (SLAs) analysis" is presented in the following sections as a proof of concept. The basic concepts, like price plans, for these analysis are already discussed in Chapter 4. The project partners SAP AG, Technical University of Darmstadt, Technical University of Karlsruhe and Techasnical University of Dresden have provided the assistance for these application scenarios.

6.5.1 Service Matchmaking and Annotation

In this application scenario, a service provider wants to develop a new service "city information service" and add semantic annotations to the new service. For this purpose, he wants to search for existing services related to city information and uses the existing services to compose a new "city information service". Furthermore, he wants to add semantic annotations to the new service according to the extracted semantics from the existing services.

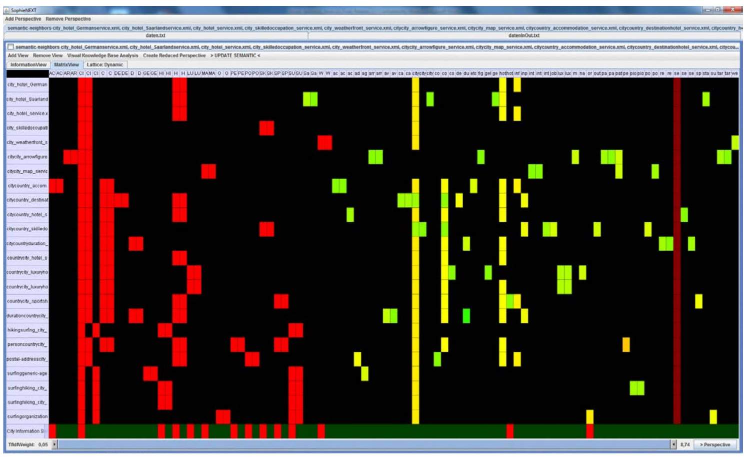


Figure 6.6: The extracted context for service matchmaking and annotation

For this application scenario, the OWL-S MX v2 [OWL10] collection of web services was selected as the data source. It consists of 551 web services

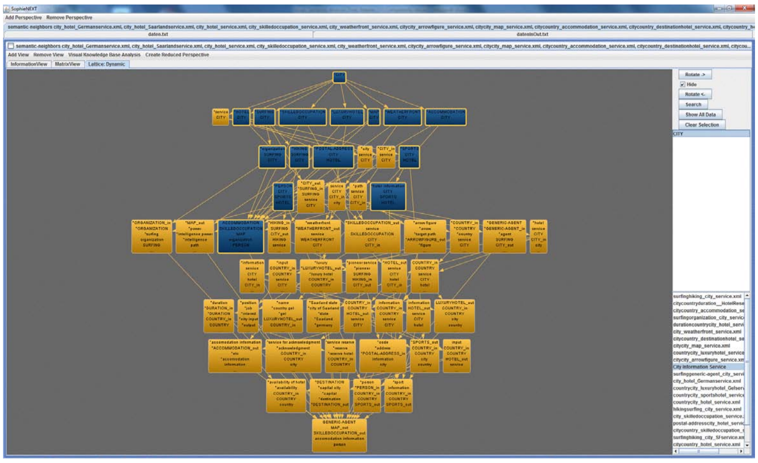


Figure 6.7: The extracted semantics information for the servcie matchmaking and annotation

from 7 domains. The VSA system *Sophie* allows service provider to extract context information from the textual descriptions of services. The visual context analysis view supports the service provider to refine extracted context by using semantic analysis techniques. The visualization of the semantic analysis results in the visual context analysis view allow to identifying services related to “city information service”. The perspective based VSA allows service provider to create different perspectives by using the parts of context in order to analyze and compare the extracted context in different perspectives. It supports him to deal with the complexity of the extracted context.

The application of the FCA allows the service provider to extract semantic information from the context. The visualization of extracted semantic in the visual semantic analysis view facilitate him to have an overview of semantic relations between services. It facilitates him also to identify services, which suite for the composition of “city information service”. Additionally, the VSA system *Sophie* facilitate him to find best fit keywords to annotate his new service. The extracted context information and semantics for service matchmaking and annotation are depicted in Figures 6.6 and 6.7

6.5.2 Price Plan Analysis

In this application scenario, a service provider is the owner of a car repair shop. He wants to offer his car repair service on the service marketplace, which is specially designed for car insurance companies. He wants to model a price plan for his car repair service and compare it with competitors before he publishes his car repair service on the service marketplace.

Figure 6.8: The visualization of extracted context for the price plan analysis (II)

The VSA system *Sophie* allows the service provider to import car repair services published on the service marketplace. The visual context analysis view visualizes automatically extracted context from the imported car repair services and provides an overview of the extracted context to him as depicted in Figure 6.8 . The semantic analysis technique FCA allows him to extract semantics from the extracted context automatically. The visual semantic refinement view presents the extracted semantics by using lattice visualization as shown in Figure 6.9. The lattice visualization facilitates him to compare the price plans of the imported car repair services.

After the analysis of the price plans of already existing services, service provider models his service semi-automatically by reusing and adapting the extracted semantics. Sophie allows him to reanalyze his new modeled service automatically together with the imported services. The visualization of reanalysis results in the visual semantic refinement view facilitate service provider to compare his new modeled service with the imported services. The reanalysis allows him to check the market competiveness of his new modeled service. He can iterate analysis, modeling and reanalysis until he is satisfied with the new modeled service. Afterwards, he can publish new modeled service on the service marketplace.

6.5.3 Legal Aspects Analysis

The service provider from the previous application scenario wants to perform initial legal assessment of his new modeled service according to privacy law issues. He imports the legal aspects specific non-functional parameters

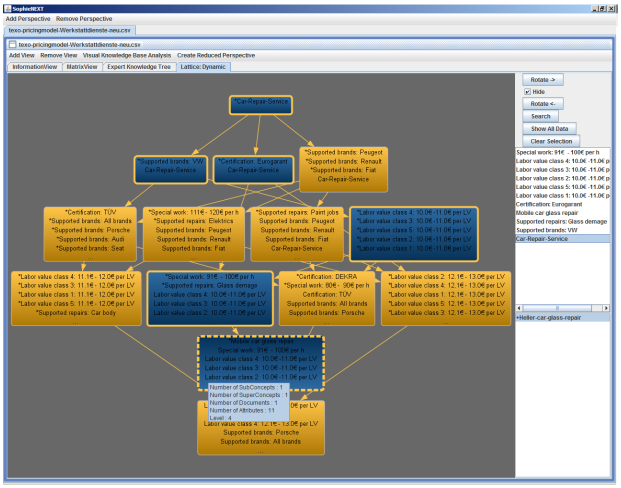


Figure 6.9: The visualization of extracted semantics for the price plan analysis

of the available car repair services from the service marketplace. *Sophie* extracts the context from the legal aspects specific non-functional parameters and presents them to the service provider. The extracted context contains questions about the functionality of services (e.g. “is the service consumer a natural person¿‘). The VSA system performs also automatic semantic analysis by applying FCA to extract semantics from the extracted context. The visualization of semantics provides him an overview of semantic relation between question related to the functionality of services and legal norms. In

Figure 6.10: The extracted context for the legal aspects analysis

order to perform initial legal assessment of his service, he has to answer the questions related to the functionality of his service. The VSA system *Sophie* performs the legal aspects analysis and recommend him if "data ascertainment necessary" required or not. The visualization of legal aspects analysis results allow to gain insight about the functionality of his service and legal issues.

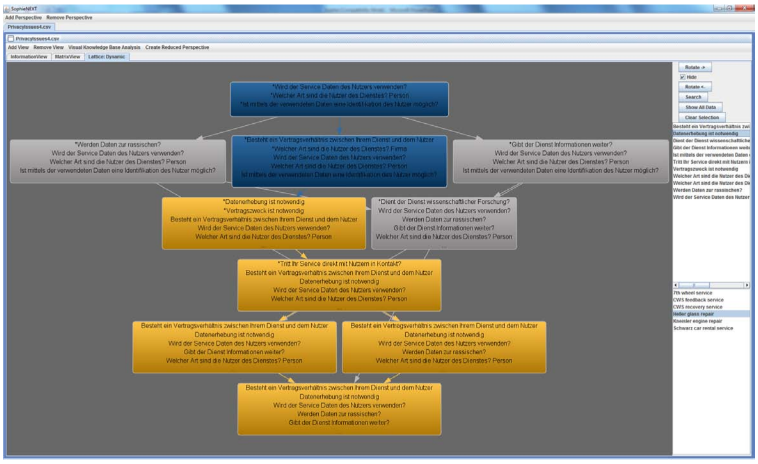


Figure 6.11: The extracted semantics for the legal aspects analysis

6.5.4 SLAs Analysis

Now the car provider wants to offer a "bill verification" service for the car repair shops. This service should check the aspects such as customer's address, customer number, price for spare parts etc. before the bill is finalized and handed over to customers. For this purpose, he wants to analyze the Service Level Agreements (SLAs) of services that are already offered on the service marketplace and then model the SLA of his planed service.

For this purpose, he imports the Quality of Service (QoS) parameters (e.g. response time and availability) and non-QoS parameters (e.g. basic price and price per transaction) of the bill verification service available on the service marketplace. The VSA system *Sophie* allows him to apply semantic analysis techniques to extract context and semantics from the SLAs. The visualization of the extracted context and semantics allows service provider to compare the prices and QoS-parameters of available services and identify competitors of his new planed service.

After the analysis of imported SLAs, the Sophie allows service provide to reuse and adapt the extracted semantics to model the SLA of his new service. The reanalysis of new modeld SLAs with the imported SLAs facilitates the comparison of new modeled SLAs with the imported SLAs. The service

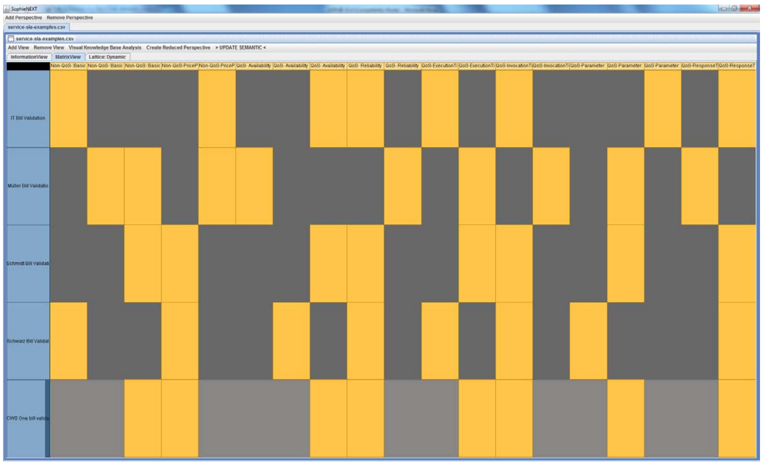


Figure 6.12: The extracted semantics for the SLAs analysis

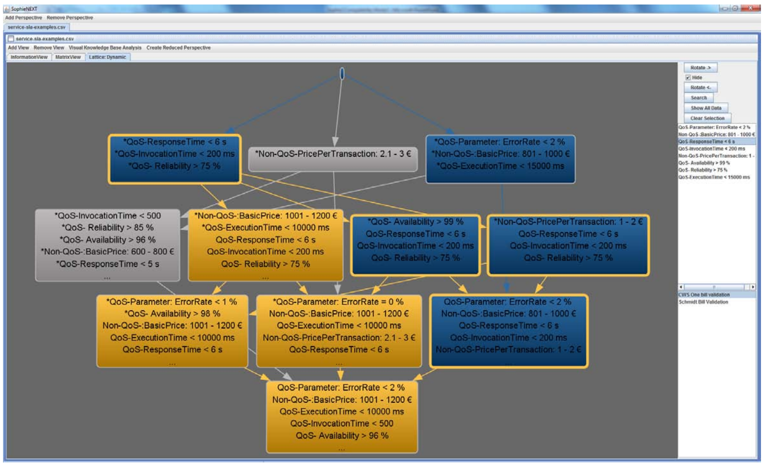


Figure 6.13: The extracted semantics for the SLAs analysis

provider can iterate this process until he is finished with the new modeled SLA of his service. Finally, he can publish the new modeled service on the service marketplace.

6.6 Summary

The VSA prototype is presented in presented in this chapter. The VSA system *Sophie* and Service Browser present the aspects like analysis, modeling and reanalysis of available services, and semantic exploration techniques for the services and their semantic descriptions. The support for semi-automatic modeling of service descriptions by applying the VSA approach in the service lifecycle and four application scenarios "service matchmaking and annotation", "price plan analysis", "legal aspects analysis" and "Service Level Agreements (SLAs) analysis" is also presented briefly in this chapter.

The main advantage of the VSA approach is to improve support for analysis, modeling and reanalysis of services by combining the semantic analysis and visualization techniques. A user study to evaluate the VSA approach will be presented in the next chapter. In this user study, the test subjects model services by using two tools USDL-Editor with a manual approach and the VSA system *Sophie* and provide quantitative and qualitative feedback, e.g. duration for the service modeling and explicit feedback.

Chapter 7

User Study for the Visual Semantic Analysis Approach

A prototype for the Visual Semantic Analysis (VSA) was presented in the previous chapter. This prototype was developed on the base of the VSA process model and the VSA conceptual framework that are presented in the Chapters 4 and 5. The VSA approach combines the semantic analysis and visualization techniques to support the semi-automatic modeling of the Service Descriptions. The expected application benefits as described in Chapter 1. A user study was conducted to verify these expected application benefits. For this purpose, The main theses of the work are that the VSA approach:

1. Improves *time efficiency* regarding time duration required to analyze, model and reanalyze services.

2. facilitates *service analysis* and semantic extraction

3. supports *semi-automatic modeling of new service descriptions* by reusing and adapting the extracted semantics

4. enables *reanalysis of new modeled services* to check their market competitiveness

5. facilitates users to deal with the *complexity of SD modeling*

The evaluation method and study design for the user study are described in the sections 7.1 and 7.2. The results of the user study are discussed briefly in the section 7.3. Finally, the conclusion is presented in the section 7.4.

7.1 Evaluation Method

A user study for the evaluation of the Visual Semantic Analysis approach was conducted with 21 participants. The test subjects are researchers and students who are involved in the research projects of the research institute "Fraunhofer Institute for Computer Graphics" (Fraunhofer IGD). All test subjects are involved in the computer science related research fields. The group of test subjects was composed of 3 females and 18 males with an average age of 27.75 years.

The cross evaluation method is applied for the evaluation of the Visual Semantic Analysis approach. According to this method, the participants were divided into two groups α (10 test subjects) and β (11 test subjects). The group α starts with the Variant A and then uses Variant B to perform the tasks. The second group uses the variants in a reversed order to perform the tasks as depicted in Figure 7.1. This approach helps to exclude factors such as learning process, which can affect the evaluation results. This approach represents a combination of within-groups design and between groups design [God09, Hof10].

According to the cross evaluation method, the two variants Variant A with USDL-Editor and Variant B with the VSA system *Sophie* as shown in Figure 7.1 are prepared for the evaluation. The USDL-Editor [TEX10] is a service

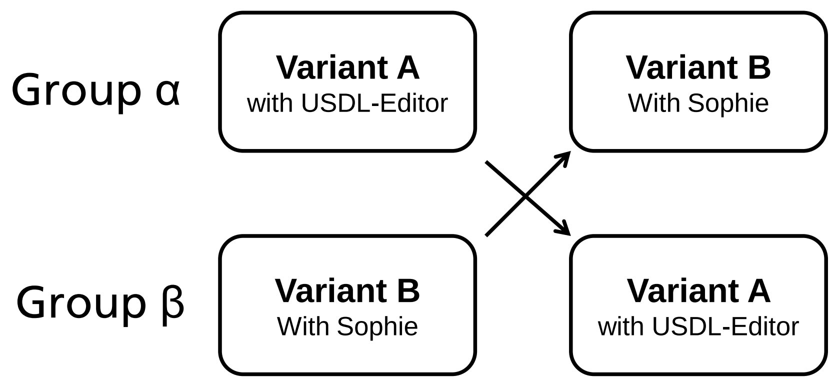


Figure 7.1: The cross evaluation

7.2 Study Design

The scenario and task for the evaluation of the Visual Semantic Analysis approach are discussed here. Furthermore, the quantitative and qualitative parameters for the user study and experiment procedure are also described in this section.

7.2.1 Scenario and Task

The evaluation study is performed within the following scenario [TEX10]: a service market place is developed for the car insurance companies to trade car insurance related services, e.g. car repair, and car insurance. Service providers such as car repair shops can offer their car repair services to the car insurance companies. An owner of a car repair shop has been observing the new trend "service market places for car repair services" in the web. He realized that it could be a new channel for the acquisition of new customers to sell car repair services like car repairing and car rental. The new potential customers can be insurance companies, who are looking for authorized car repair shops. He decides to offer his car repair service on the service market place, which is specially designed for the car insurance companies. For this purpose, he has to model his service according to the guidelines of the service market place. Specifically, he has to model the price plan of his car repair shop according to the following subtasks:

- Subtask 1 - Analyze available services on the service market place: In this subtask, service provider can analyze the price plans of the available car repair services on the service market place to have an overview of existing price plans. He can check the offered services, e.g. car glass repair or paint jobs and their prices. He can also identify the different pricing strategies of the competitors.
- Subtask 2 - Model new car repair service: After the analysis of the available service, service provider can decide for his pricing strategy and model the price plans of his new "car repair service" under the consideration of the results of service analysis. For the modeling of new car repair service, the modeling guidelines of the service market place have to be followed.
- Subtask 3 - Reanalyze new-modeled service together with available services on the service market place: After the modeling of new service, the reanalysis of the new-modeled services together with available services for the comparison. It helps users to check the market competiveness of new-modeled service and have an overview of competitors.

- Additional subtask - Support for iterartive service modeling: The iterative modeling of services allows first the selection of different aspects of SDs, e.g. price plans or SLAs, for the analysis and then performs analysis, modeling, and reanalysis. It facilitates users to deal with the comlexity of SDs. These steps can be iterated until the service provider is satisfied with the new-modeled service and its market competitiveness.

7.2.2 Quantitative and qualitative parameters

The time duration to perform a subtask and sum of the time duration of the entire task are considered as quantitative parameters for the evaluation of the VSA approach. The order of two variants can affect these quantitative parameters. The time duration to perform each subtask of both variants was measured by using a stopwatch. The sum of duration for each subtask was also calculated as time duration for the entire task.

Three questionnaires for each subtask were prepared to collect qualitative data during the evaluation of the VSA approach. The focus of these questionnaires was the functionality of both variants rather that usability issues. For example, whether the analysis and reanalysis techniques support the service provider during the modeling of new services. The questionnaires were composed of multiple choice questions with five alternative answers. The questions were formulated as statements. The alternative answers could range from 1 to 5 (1 = supported not at all; 5 = Supported very much). The test subjects should fill out a questionnaire after the accomplishment of each subtask for both variants.

An additional questionnaire was designed to evaluate the iterative analysis and modeling functionality of the *Sophie*, because this functionality is not directly offered by USDL-Editor. To collect qualitative data for this functionality of the VSA approach, this functionality of *Sophie* was presented to test subjects and they should fill out the questionnaires to provide their feedback.

A final questionnaire was prepared to compare both tools directly. The test subjects should fill out this questionnaire at the end of the user study. They should choose a tool from both USDL-Editor and *Sophie* as their proffered tool for the each subtask and entire task. They could also provide their comments for the evaluation as free-text. They could provide their comments if they like a lot the functionality of the *Sophie* tool or they could provide their suggestions for the improvement of the *Sophie*. The questionnaires are presented in the appendix A.

1 DIAGRAM 1

Version:

Author:

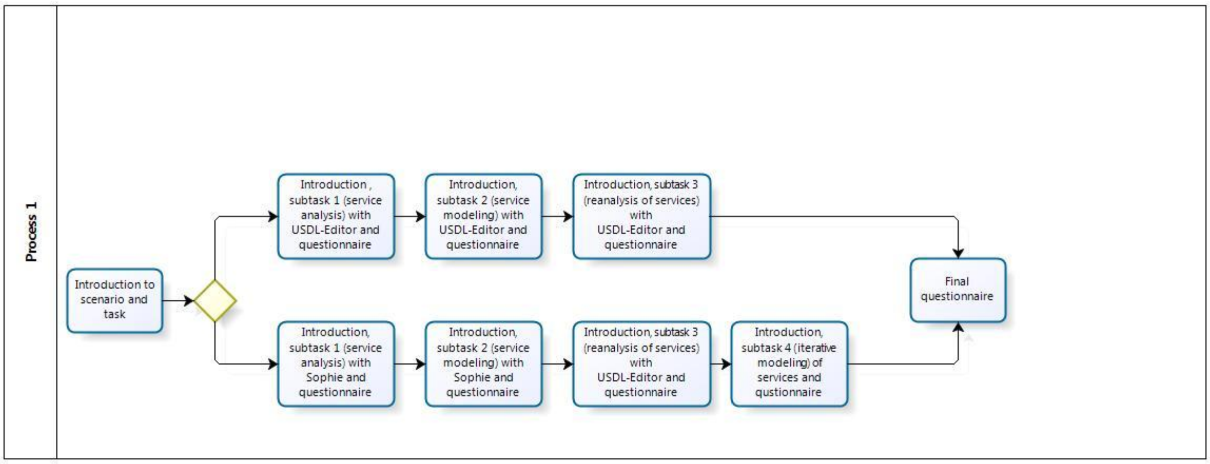


Figure 7.2: The experiment procedure for the user study

1.1 PROCESS 1

7.2.3 Experiment Procedure

The experiment procedure is depicted in Figure 7.2. It consists of three main phases general introduction, experiment and wrap up. The general introduction and wrap up phases are same for both variants. The order of Variants A (with USDL-Editor) and Variants B (with *Sophie*) changes for the test subjects depending on their assignment to group α or β in the experiment phase.

The application scenario and task were introduced to test subjects as a general introduction. Additionally, the experiment procedure and basic concept of services and price plans were also introduced. After the initial introduction, test subjects started either with USDL-Editor or *Sophie* depending on their assignment to the group α or β as proposed by the cross evaluation method. The group α starts always with the USDL-Editor and then perform the subtasks with *Sophie* in the experiment phase as shown in Figure 7.2. The group β used USDL-Editor and *Sophie* in reverse order to perform the subtasks.

At the beginning of each subtask, the domain knowledge, e.g. pricing model and the functionality of the used tool, were introduced briefly to test subjects. They could try the functionality of the used tool before they start performing the subtasks. The introduction sessions to each tool were the same for both groups α or β. This approach should avoid the learn effects, when they were performing subtasks by using both tool in a specific order.

In the wrap up phase, the test persons should fill out the final questionnaire and choose their preferred tool for the individual subtasks and the entire task. Finally, they could also add their comments about the VSA prototype *Sophie*.

1.1.1 PROCESS ELEMENTS

1.1.1.1 Element

Gates

Gate

Gate

1.1.1.2 Final questionnaire

Introduction , subtask 1 (service analysis) with

1.1.1.3 USDL-Editor and questionnaire

7.3 Evaluation Results

An ANalysis Of VAriance (ANOVA) [Fie09] provides a statistical test whether three or more means are the same, so it tests the hypothesis that all group means are equal. In our user study, we have four means according to the cross evaluation, therefore, the ANOVA is suitable for our study. An ANOVA produces an F-statistic or F-ratio. It compares the amount of systematic variance in the data to the amount of unsystematic variance. The F-statistic was conducted to evaluate the results of the user study. It indicates F-value and p-value. One example of F-value and p-value is $F(1; 19) = 48.05$, $p < .001$. Two values in the parentheses of this example the "degrees of freedom of the treatment variance (number of compared groups - 1)" and "the degrees of freedom of the error variance (number of test subjects - number of groups)" are significant parameters for the F-value [Hof10, Fie09].

The significance of the F-Value can be checked by comparing the F-value with a given table of F-distribution. The difference between F-Value and a given F-distribution results in the p-value. The p-value $< .05$ indicates for significant variance and p-value $< .001$ indicates for highly significant variance[Hof10, Fie09].

The results of F-statistics show that only highly significant effects with p-value $< .001$ for the VSA prototype*Sophie* were identified. The order of tools execution did not have any significant effect for the user study. For this reason, the results of F-statistics for all test subjects are presented together without distinguishing between groups.

7.3.1 Time efficiency

The time efficiency was measured for each subtask individually and for the entire task. The results of F-statistic processed by the SPSS tool [SPS10] are listed in Table 7.1 and shown in Figures 7.3 - 7.6.

The usage of the semantic analysis technique FCA and interactive lattice visualization for the VSA approach facilitate users to analyze price plans of services automatically. The reuse and adaption of semantics extracted from the automatic analysis of services support users to accelerate the modeling of new services. After the modeling of new services, automatic reanalysis of the new-modeled service with available services supports users to compare the price plans of new-modeled service with the available services. Therefore, it was assumed that the VSA system *Sophie* leads to the improvement of the time efficiency. The results of the user study show that the time efficiency was improved by using the VSA system *Sophie* for each subtask and the entire task as depicted in the Figures 7.3 - 7.6.

Subtask	F-value and p-value	Mean time in minutes for *Sophie*	Mean time in minutes for USDL-Editor
Subtask 1	Significant main effect for the support of *Sophie*, F(1; 19) = 23.74, p < .001.	M = 1:47 SD = 0:58	M = 2:52 SD = 1:14
Subtask 2	Significant main effect for the support of *Sophie*, F(1; 19) = 28.71, p < .001.	M = 3:45 SD = 1:22	M = 5:53 SD = 2:00
Subtask 3	Significant main effect for the support of *Sophie*, F(1; 19) = 16.27, p < .005.	M = 2:28 SD = 1:14	M = 3:33 SD = 1:43
All subtasks	Significant main effect for the support of *Sophie*, F(1; 19) = 48.05, p < .001.	M = 8:02 SD = 2:25	M = 12:19 SD = 3:17

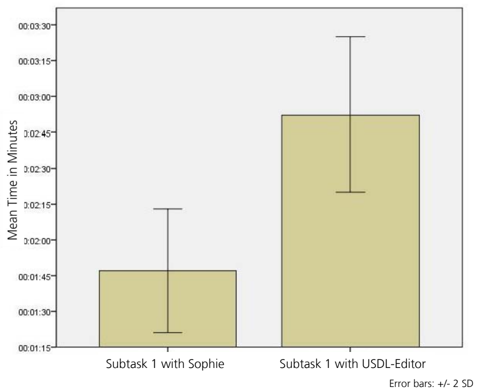
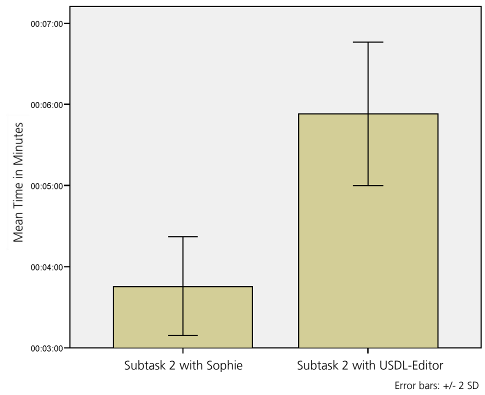


Figure 7.3: The comparison of time required for the subtask 1

7.3.2 Service Analysis

The VSA system *Sophie* offers the semantic analysis technique FCA to analyze services automatically. The interactive lattice visualization of service analysis results facilitates users to compare the price plans of services and find cheap or expensive services easily. Therefore, it was expected that the automatic analysis of services and interactive lattice visualization of analysis



Fi ‹ 2

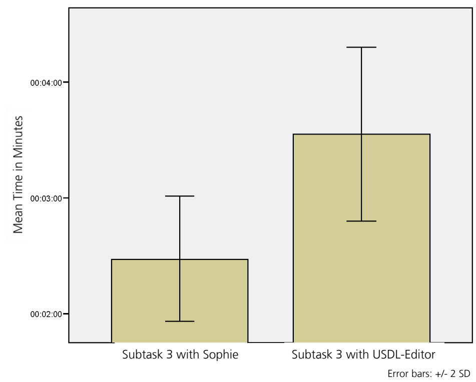


Figure 7.5: The comparison of time required for the subtask 3

results can improve the support for the service analysis.

According to the F-statistics, there was a highly significant main effect for the VSA system *Sophie* to support the service analysis, $F(1; 19) = 25.68$, $p < .001$. The users indicated that the *Sophie* facilitated the service analysis better ($M = 3.79$, $SD = 0.57$) than the USDL-Editor ($M = 2.78$, $SD = 0.81$) as shown in Figure 7.7. It verifies that the VSA approach improves the support for the service analysis.

7.3.3 Semi-Automatic Modeling of New service descriptions

The interactive visualization of service analysis results also allows users to identify services similar or competitor to the new planed services. The reuse or adaption of semantics extracted from the service analysis facilitate them to model new services semi-automatically. Additionally, the interactive lattice visualization supports users to model new services easily. Therefore,

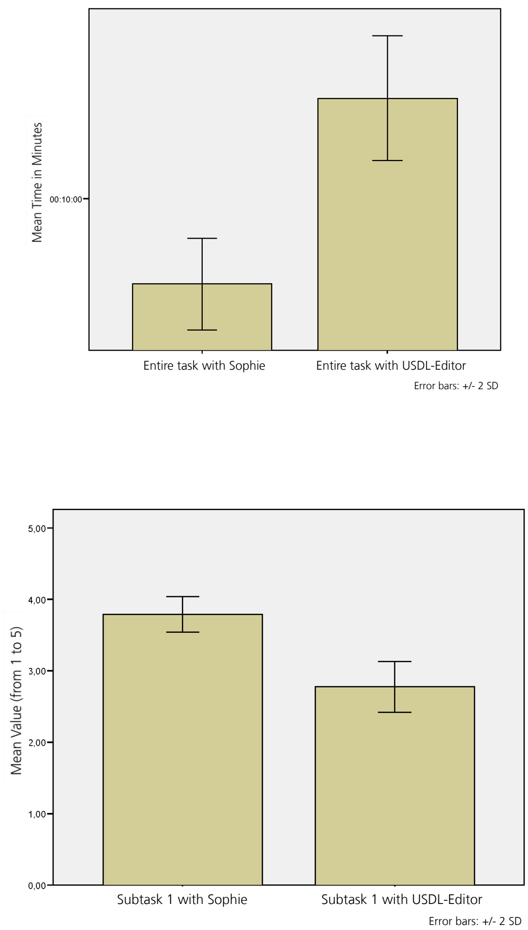


Figure 7.7: The comparison of support for the service analysis

it was assumed that the VSA approach could improve the support for the service modeling.

The variance analysis shows that the VSA system *Sophie* facilitates users during the modeling of services significantly, $F(1; 19) = 212.89$, $p < .001$. The modeling of services is better supported by *Sophie* ($M = 4.23$, $SD = 0.49$) than the USDL-Editor ($M = 2.63$, $SD = 0.66$) as shown in Figure 7.8.

7.3.4 Reanalysis of New Modeled Services

After the analysis of available services and modeling of new services, the automatic reanalysis of new modeled services with the Formal Concept Analysis (FCA) and visualization of reanalysis result with the lattice visualization support users to compare new modeled services with the available services easily. It allows them to check the market competitiveness of their new mod-

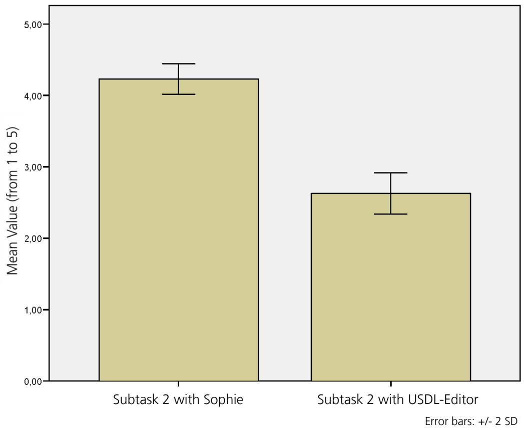


Figure 7.8: The comparison of support for the service modeling

eled services. Therefore, It is expected that the VSA approach improves the support for the reanalysis of new modeled services.

According to the F-statistics, the VSA system *Sophie* offers also a significant better support for users to reanalyze new-modeled services with available services, $F(1; 19) = 75.20$, $p < .001$. The support for the reanalysis of modeled services with available services was better with *Sophie* ($M = 3.92$, $SD = 0.61$) than the USDL-Editor ($M = 2.63$, $SD = 0.84$) as shown in Figure 7.9. It supports the hypothesis that the VSA prototype improves the support for the reanalysis of new modeled services.

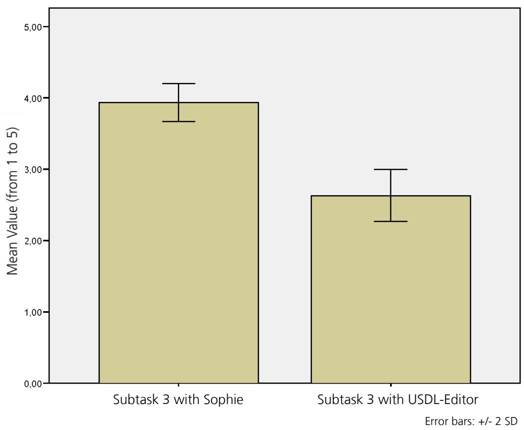


Figure 7.9: The comparison of support for the reanalysis of services

7.3.5 Complexity of SD Modeling

The analysis, modeling, and reanalysis of different aspects, e.g. price plans, SLA analysis and functionality, of services at the same time can be very challenging issue for users. The perspectives in the VSA offers multiple perspectives to deal with the complexity of the SDs. It allows users to consider different aspects of the SDs separately or combination of these aspects together. Users can analyze, model, and reanalyze these aspects in different perspectives individually or together.

The results of the variance analysis are presented in Table 7.2. These results support the hypothesis that the perspectives in the VSA facilitates users to deal with the complexity of SD modeling.

Question nr.	Questions	Mean Value
1	The iterative analysis and modeling of services with *Sophie* was intuitive	4.19 (SD=0,93)
2	The iterative analysis and modeling approach of *Sophie* helped well to get better analysis overview.	4.51 (SD = 0.51)
3	The iterative analysis and modeling approach of *Sophie* helped to improve the modeling of services.	4.33 (SD=0.57)

Table 7.2: The results of the variance analysis for the iterative modeling of services

Task	Preferred Tool *Sophie*	Preferred Tool USDL-Editor
Subtask 1	90.5%	9.5%
Subtask 2	81%	19%
Subtask 3	100%	0%
Entire task	100%	0%

Table 7.3: The results of F-statistic for the general user satisfaction

7.3.6 General user satisfaction

After the user study, the test subjects should fill out a final questionnaire to choose their prefer tool for individual subtasks and for the entire task. The results of their feedback are shown in Table 7.3. These results show that a significant majority of test subjects preferred *Sophie* to analyze, model and reanalyze services. For subtask 2, 19% of test subjects prefers the USDL-Editor because they prefer to model services with forms based editor. The test subjects have also provided their positive or negative feedback about the VSA system *Sophie* as free text. In this way, they could provide their explicit feedback that was covered by the questionnaires. These comments are listed in Appendix A.

7.4 Conclusion

The Visual Semantic Analysis approach combines semantic analysis and interactive visualization techniques to support semi-automatic modeling of SD. The prototype of the VSA system *Sophie* is developed on the VSA conceptual framework. Different application benefits referring to aspects efficiency, service analysis, service modeling, reanalysis of modeled services, iterative modeling and general user satisfaction were expected from this prototype. A user study was conducted to verify its application benefits. Two tools USDL-Editor (without support of the VSA approach) and *Sophie* (with the support of the VSA approach) were compared to verify the application benefits of the VSA system *Sophie*.

The user study supports the hypotheses that the VSA system Sophie improves the time efficiency for the individual subtasks and entire task of the service modeling process. It improves also the support for the analysis, modeling, and reanalysis of services. Additionally, it leads to high degree of user satisfaction for the individual subtasks and entire task. The VSA approach supports the semi-automatic modeling of services and accelerates the service modeling process.

The perspectives based VSA allows users to consider different aspects of SDs, e.g. price plans, SLAs and functionality of services, in different perspectives. The users can analyze, model, and reanalyze these aspects of sevices in different perspectives individually or together with the perspectives in. The results of the user study emphasize the hypothesis that the VSA approach improves support for iterative modeling of services by offering perspectives in VSA.

The user study should not lead to false impression that the VSA approach can replace the USDL-Editor. The basic assumption for the VSA approach is that the services are available, which could be similar or related to new planned services. The VSA approach can analyze these available services to support the semi-automatic modeling of new planned services. If the similar or related services to new planned services are not available and the new planned services have to be modeled from the scratch, then the USDL-Editor is more efficient than the VSA system Sophie. It means that the VSA approach of Sophie and manual modeling approach of USDL-Editor complement each other and a combination of both could lead to better support of the semi-automatic modeling of services.

Chapter 8

Conclusion and Future Work

A new trend *Web service ecosystems* for Service-Oriented Architectures (SOAs) and Web services is emerging [BD06, BDB05]. Service providers and service brokers can trade their own services and combine services to create value added services in these ecosystems. Service brokers offer services of different service providers to service consumers. The *flexible service discovery* to find and combine services is one of the main challenges for Web service ecosystems (e.g. IoS) [BD06, CBMK10, CWV09, OBB^{+}09]. The non-functional parameters of services, e.g. price plans, legal aspects, Service Level Agreements (SLAs), penalties, and delivery modes, have to be described as Service Descriptions (SDs) to offer the *flexible service discovery*. Service descriptions can be very complex, e.g. the price plans of SDs may contain information like pay per use, monthly fees, flat rates, discounts, and special offers. Therefore, the explicit and formalized description of services' non-functional parameters is the bottleneck for Web service ecosystems [CBMK10, OBB^{+}09, CWV09]. The automation of SD modeling task is desirable to as it is mentioned in different research work [WWWC08, SP07, Sab06, SWGS05, WGG^{+}04].

This thesis introduces an innovative approach Visual Semantic Analysis (VSA) to support semi-automatic modeling of services in Web service ecosystems. This approach combines semantic analysis and interactive visualization techniques to support analysis, modeling, and reanalysis of services in an iterative loop. Additionally, it offers perspectives to consider different aspects of SDs individually or simultaneously and thus facilitates users to deal with the complexity of Service Descriptions.

This chapter summarizes the main contributions of this thesis in Section 8.1. Section 8.2 illustrates the application benefits of the VSA approach. Finally, prospects for the directions of future research are provided in Section 8.3.

8.1 Conclusion and Contributions

The contributions of this thesis are as follows:

8.1.1 Study on the Visual Semantic Analysis

The knowledge experts from different domain like service engineers and SSD modeling experts were interviewed in a study to drive the requirements for the VSA as presented in Chapter 4. The requirements were considered form three perspectives. First, users perspective deals with the interplay between the semantic analysis and the SD modeling,and usage of visualization to support semi-automatic modeling of services. According to users perspective the requirements were (I) a process model to support the analysis, modeling, and reanalysis of services, (ii) support for the service analysis, (iii) support for the modeling of services on the base of the results of service analysis, (iv) support for the reanalysis of new modeled services with available services, (v) the iterative analysis, modeling and reanalysis to support the semi-automatic modeling (vi) visual support for the analysis, modeling and reanalysis of services, and (vii) multiple perspective to deal with the complexity of SSDS and a large number of services.

Second, the semantic analysis perspective covers the aspects related to analysis of the available service and usage of visualization techniques for it. The requirements form the semantic analysis perspective are summarized as (i) abstract process model for the analysis of services, (ii) support for different input formats, (iii) visual support for the context and semantic analysis (iv) support for the multiple semantic extraction techniques (v) support for the import and export of context and extracted semantics. Third, the modeling of new services and usage of visualization for the service modeling are addressed by the SD modeling perspective. The requirements (i) an abstract process model for the SSD modeling, (ii) support for the comparison of new modeled services with available services, (iii) Support for the Modularization approach of SSD modeling, (iv) Support to assure the consistency of SSDs, (v) support for different SD modeling languages, and (vi) visual support for the SD modeling are covered in the SD modeling perspective. These requirements serve as basis for the VSA approach.

8.1.2 A Generic Process Model for the VSA

A generic process model for the VSA describes (i) sub processes and tasks, (ii) transitions between sub processes, (iii) technologies used in sub processes, and (iv) data processed and generated in sub processes. The six sub processes “Visual Context Acquisition”, “Visual Context Analysis”, “Semantic Extraction, “Visual Semantic Refinement”, “Visual SD modeling” and ´´Governance" are presented in Chapter 4. The iterative loop illustrates the analysis, modeling, and reanalysis of services in an iterative manner. The *Visual Context Acquisition* allows users to extract context from unstructured and structured information of service descriptions, e.g. textual description of services and non-functional parameters. The *Visual Context Analysis* facilities users to prune or refine context information by using semantic analysis techniques from the sub process *Semantic Extraction* that provides all semantic analysis techniques in the VSA. The *Visual Semantic Analysis* supports users to validate and refine the extracted semantics by offering visualization techniques. The extracted semantics is used in the sub process *Visual SD Modeling* to model new services. The sub process *Governance* validates the new modeled SDs. After a successful validation, the new modeled services can be published on service platforms.

The Natural Langue Processing (NLP) techniques allow users to extract context from the service description in the sub process *Visual Context Acquisition*. The visualization techniques supports users in the sub processes *Visual Context Analysis*, *Visual Semantic Refinement*, and *Visual SD Modeling* to gain insight into context, semantics, and service descriptions. The semantic analysis techniques are offered in the sub process *Semantic Extraction* to analyze context and extract semantics form it. The service validation techniques allow users to check the consistency of new modeled services in the sub process *Governance*.

8.1.3 A Conceptual Framework of the VSA

A conceptual framework describes system components and system behavior of the Visual Semantic Analysis system according the VSA process model. Multiple perspectives for the VSA play an important role for the design decisions of the conceptual framework of the VSA. Multiple perspectives with views are defined in the conceptual Framework of the VSA that allow consideration of different aspects of service descriptions, e.g. price plans and SLAs, individually or simultaneously. The *perspective controller* serves as a bridge between all system components and manages the transitions between the sub processes and their execution order. The views *Visual Context Analysis View*, *Visual Semantic Refinement View*, and *Visual SD Modeling View* represent the sub processes *Visual Context Analysis*, *Visual Semantic Refinement*, and *Visual SD Modeling* of the VSA process model. They use available visualization techniques from the *Visual Layout Algorithms Library* to visualize processed and generated context, semantics and new modeled services. The semantic extraction techniques are provided by the *Semantic Analysis Algorithms Library*. The component *Context Acquisition* allows the context extraction from service descriptions.

The *communication* components serve as an external interface of the conceptual framework of the VSA. They manage communication between the VSA system and service platforms (Web Service Ecosystem), e.g. import of the available services from service platforms and the publishing of the new modeled services on service platforms. Furthermore, the management of the extracted context and semantics is also described in the VSA conceptual framework.

8.1.4 Application of the VSA in the Service Lifecycle

A prototype of the Visual Semantic Analysis System is developed on the base of VSA conceptual framework. It consists of a *service platform*, a VSA system *Sophie* and a Web client *Service Browser* as discussed in Chapter 6. The service platform provides all services needed by the VSA prototype, e.g. service discovery and service registry. The VSA system Sophie presents the aspects such as analysis, modeling and reanalysis of services. The Web client Service Browser allows users to navigate through the semantic relation of services and edit the service descriptions.

The application of the VSA in the service lifecycle is presented in Chater 6. It gives an overview of tasks in the service lifecycle phases, where the VSA approach can be applied. The roles of the service lifecycle, e.g. service providers and service brokers, who can profit from the application of VSA approach, are also presented in Chapter 6.The usage of VSA system for four application scenarios service matchmaking and annotation, price plan analysis, legal aspects analysis and Service Level Agreements (SLAs) analysis is presented as a proof of concept.

8.2 Application Benefits

A user study was conducted to verify the expected application benefits referring to efficiency, service analysis, service modeling, reanalysis of modeled services, and complexity of SDs and user satisfaction as presented in Chapter 7. The cross evaluation method was applied as evaluation method in the user study. The VSA prototype Sophie was compared with a service modeling tool USDL-Editor in this user study. According to the evaluation of the results of user study, the application benefits of the VSA approach are as follows:

The VSA approach improves time efficiency regarding time duration required to analyze, model, and reanalyze services.

The automatic analysis of service descriptions and visualization of the analysis results facilitate users to compare the already existing services easily.

The reuse and adaption of analysis results to model new services allow them to model new services semi-automatically. The reanalysis of new modeled services with the already existing services supports users to compare new modeled services with existing ones. In this way, the VSA approach allow users to analyze, model and reanalyze services in an iterative loop and accelerate modeling of new services. The results of the user study show that the time efficiency was improved significantly by using the VSA system *Sophie* for each subtask and the entire task.

We show that the VSA approach facilitates the service analysis and semantic extraction.

The VSA approach allows users to apply a combination of semantic analysis techniques to analyze services and extract semantic information from the already existing services. The visualization of the service analysis results and extracted semantics allows users to detect patterns, e.g. similar services, in the analysis results and compare different aspects of service descriptions such as price plans. A combination of semantic analysis techniques and visualization techniques improves the support for the service analysis as shown by the results of the user study.

The reuse and adaption of extracted semantics supports semi-automatic modeling of new service descriptions significantly with the VSA approach.

The interactive visualization of the service analysis results and extracted semantics facilitates users not only to gain an overview of analysis results, but also reuse and adapt the extracted semantics to accelerate the modeling of new services. In this way, the VSA provides an integrated approach for analysis and modeling of services to support the semi-automatic modeling of service descriptions. The integrated approach allows users to consider market competitiveness aspects, e.g. price plans of competitor services, during the modeling of new services and improve support for the semi-automatic modeling significantly.

The VSA approach enables reanalysis of new modeled services to check their market competitiveness.

The reanalysis of new modeled services with the already existing services allow users to compare their new modeled services with existing ones. The VSA approach offers them an iterative loop, where they can combine semantic analysis and visualization techniques to analyze, model and reanalyze services. The results of the user study shows that the iterative loop improves the reanalysis of services and overall support for the semi-automatic modeling of service descriptions.

The perspectives in the VSA approach facilitate users to deal with the complexity of SD modeling.

The simultaneously modeling of different aspects of service description, e.g. price plan, SLA and legal aspects, is a complex task. According to the user requirements, users want to model different aspects of SDs either individually or a combination of them. The VSA approach offers them different perspectives to consider different aspects of SDs individually or simultaneously. The results of the user study prove that perspectives in the VSA facilitate users to deal with the complexity of service descriptions.

8.3 Future Work

Several open issues, which were not discussed in depth or emerged in the course of this thesis, lead to the following research directions.

8.3.1 Collaborative Visual Semantic Analysis

According to Oberle et al. [OBB+09], the modeling of service descriptions is a multi disciplinary task. The support for *collaborative* modeling of service descriptions is one of the most import issues to allow usesr to model different aspects of SDs collaboratively in the different phases of a service lifecycle. Different users provide the same kind of information independently and differently during the modeling of different aspects of service description. A sensible information management (change tracking) is needed to avoid *inconsistency*. The information has to be modularized to allow different user roles to maintain and contribute information corresponding to different aspects in different phases of a service lifecycle [OBB+09].

The collaborative VSA approach has to be investigated to support experts from different domains during the modeling of service descriptions. The issues such as task assignment to experts or group of experts, change tracking and versioning have to be investigated particularly. The change tracking to avoid inconsistency and service governance for the validation of services according are the main challenges for the collaborative visual semantic analysis.

8.3.2 Service Monitoring

The monitoring of services based on Service Level Agreements (SLAs) is one of important challenges for the Internet of Services (IoS). The traditional service monitoring approach based on Quality of Service (QoS) parameters is not sufficient for the IoS. The monitoring needs to account not only technical aspects, but also the business aspects of services. These aspects should be comprised of business related aspects such as pricing and legal aspects as well as technical aspects like QoS parameters. The monitoring helps to ensure that the obligation of consumers and providers are met and promised service quality is fulfilled [WCS08].

A clear understanding of monitoring parameters at technical and business level and a mapping between them would be the base for automation the monitoring process by driving business parameters from technical monitoring results. The application of VSA approach in the automatic monitoring process to gain a clear understanding of monitoring related technical and business level parameters and mapping between them can help to automate the monitoring process.

8.3.3 Semantic Annotation of Multimedia Data

A process-based design of multimedia annotation system [Hof10] presents a generic process model of multimedia annotation for the fundamental knowledge about the activities and procedures associated with multimedia annotation. Hofmann [Hof10] has also presented the application of VSA approach as an external service to perform semi-automatic semantic annotation of multimedia data. The specified tags of the multimedia data from the multimedia annotation system serve as context in the VSA to generate semantics. The multimedia annotation system uses the extracted semantics further for the modification of it.

An interesting research area is the extension of generic process model of multimedia annotation with the generic VSA process model to develop a generic process based design for the semantic annotation of multimedia data. For this purpose, sub processes and tasks, execution order of the sub process, technologies used in the sub processes, and data processed in the sub processes have to be defined for the semantic annotation of multimedia data.

8.3.4 Context Based Visualization

The context based automatic choice of visualization techniques [God09] for the VSA approach is very interesting research question. For this purpose, the objectives of users referring to analysis of different aspects e.g. price plan, legal aspects and SLAs, have to be taken into account. Additionally, it has to be investigated which metaphors are appropriate in the VSA process model to visualize context, semantics and service descriptions. The empirical study can be performed to find appropriate metaphors for the VSA approach. Particularly, the influence of the multi disciplinary teams in the service modeling process has to be investigated for the context based automatic choice of visualization techniques in the VSA.

Appendix A

User Study: Questionnaires

Evaluation of the Service Modeling Tools
(Variant B: Service Modeling with SOPHIE)

In this evaluation, I want to evaluate two service modeling tools "USDL-Editor" and "SOPHIE". For this purpose, I will introduce you first the application scenario for the evaluation. Then you will be asked to perform a set of tasks. I will introduce you to the required functionality of each tool for each task before you start to perform the task. Afterwards, you will be asked to perform each task with both tools in order to compare both tools. After the completion of each task, you will be asked to fill out one questionnaire. At the end of the evaluation, you will be asked to fill out a final questionnaire.

Participant's Info

Participant number:
Age:
Gender:

Question 1.1: how good is your knowledge about PC?

No any knowledge □ □ □ □ □ Very good knowledge

Question 2.2: How good is your knowledge about services and service modeling?

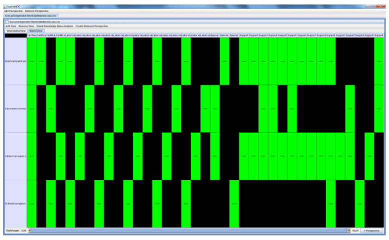
No any knowledge □ □ □ □ □ Very good knowledge

Application Scenario

You are the owner of a car repair shop. You are observing a new trend "service market places for car repair services" in the web. It can be a new channel for the acquisition of new customers to sell car repair services like car repairing and car rental. The new potential customer can be insurance companies, who are looking for authorized car repair shops. You decide to offer you car repair service "MyCarRepair-Service" on the service market place, which is specially designed for car insurance companies.

Task

Your task is to model a price plan for your car repair service "MyCarRepair-Service". You use Unified Service Description Language Editor (USDL-Editor) as shown in the following figure to model a price plan for your car repair service. For this purpose, you have to perform the following sub-tasks.



Sub-task 1: Analyze car repair services available on the service market place

(Short introduction to SOPHIE's functionality "analysis of existing services")

Find a service with the highest price for the "Supported repair: paint jobs" and "Certification:DEKRA" from the available car repair services on the service market place. To perform this sub-task, you have to follow the following steps:

1. Select "Supported repair: paint jobs" and "Certification:DEKRA" in the right upper panel of graph visualization.
2. Select the nodes in the graph to find the most expensive service.
3. Write down the name of most expensive service and its prices (See right bottom panel of the graph visualization)

Questionnaire 1

Question 1.1: It was intuitive to analyze available service on the service market place.						
Not at all	□	□	□	□	□	Very much
Question 2.2: The support for the analysis of available service on the service market place was good.						
Not at all	□	□	□	□	□	Very much
Question 1.3: The navigation between services and their price plans was good.						
Not at all	□	□	□	□	□	Very much
Question 1.4: It was easy to get an overview about prices of services.						
Not at all	□	□	□	□	□	Very much
Question 1.5: It was easy to compare the prices of services.						
Not at all	□	□	□	□	□	Very much
Question 1.6: The support to find most expensive services was good.						
Not at all	□	□	□	□	□	Very much
Question 1.6: The overall support to perform this sub-task was good.						
Not at all	□	□	□	□	□	Very much

Sub-task 2: Model your own car repair service

(Short introduction to SOPHIE's functionality "model a service")

Model your service "MyCarRepair-Service" with the highest price and "Certification:DEKRA" for the "Supported repair: paint jobs" and add new feature to your service "Supported repair: express paint job". To perform this sub-task, you have to follow the following steps:

1. Add new service (Right click -> add document) in the matrix visualization
2. Add new attribute (Right click -> Supported repair: express paint job)
3. Add price plan features (Right click -> add Entries)
   a. Select your new service
   b. Select all related attributes
      i. Car repair Service
      ii. All Labor value classes 1 to 5 (according to the service Schwarz-car-repair-service)
      iii. Special work (according to the service Schwarz-car-repair-service)
      iv. "Supported repair: paint jobs"
      v. Supported repair: express paint jobs
      vi. Certification:DEKRA

Questionnaire 2

Question						
Question 2.1: It was intuitive to model your service.						
Not at all	□	□	□	□	□	Very much
Question 2.2: The support to model your service was good.						
Not at all	□	□	□	□	□	Very much
Question 2.3: The support to find the different parameters for the modeling of your service was good.						
Not at all	□	□	□	□	□	Very much
Question 2.4: The support to see the competitor most expensive service during the modeling of your service was good.						
Not at all	□	□	□	□	□	Very much
Question 2.5: The overall support to perform this sub-task was good.						
Not at all	□	□	□	□	□	Very much

Sub-task 3:Re-analyze (Compare your car repair service with available car repair services on the service platform)

(Short introduction to SOPHIE's functionality "re-analyze services")

Re-analyze your modeled service "MyCarRepair-Service" with competitor car repair services available on the service platform and find all car repair services, which are direct and indirect competitors of your service. To perform this sub-task, you have to follow the following steps:

1. Click the button "show all data" in the right upper panel
2. Search for "Supported repair: paint jobs" and "Certification:DEKRA" in the right upper panel of graph visualization.
3. Find all competitors
   a. Services with attributes "Supported repair: paint jobs" and "Certification:DEKRA", which have same or more price
      i. Write down the name of services
   b. Services with attributes "Supported repair: paint jobs" and "Certification:DEKRA", which are cheaper
      i. Write down the name of services

Questionnaire 3

Question 3.1: It was intuitive to find all competitors of your service?						
Not at all	□	□	□	□	□	Very much
Question 3.2: The support to compare your own service with other available services on the service market place was good.						
Not at all	□	□	□	□	□	Very much
Question 3.3: It helped well to compare the unique selling points of your service with competitors.						
Not at all	□	□	□	□	□	Very much
Question 3.4: It helped well to improve the market competitiveness of your service?						
Not at all	□	□	□	□	□	Very much
Question 3.5: The overall support to perform this sub-task was good						
Not at all	□	□	□	□	□	Very much

Sub-Task 4: Iterative analysis and modeling of services

Demonstration of the iterative analysis and modeling of services with SOPHIE.

Questionnaire 4

Question 4.1: The iterative analysis and modeling of services with SOPHIE was intuitive?						
Not at all	□	□	□	□	□	Very much
Question 4.2: The iterative analysis and modeling approach of SOPHIE helped well to get a better analysis overview?						
Not at all	□	□	□	□	□	Very much
Question 4.3: The iterative analysis and modeling approach of SOPHIE helped to improve the modeling of services?						
Not at all	□	□	□	□	□	Very much

Final Questionnaire:

Question 5.1: Which Software-Tool do you prefer to analyze available services on the service market place?
USDL-Editor □ □ Sophie
Question 5.2: Which Software-Tool do you prefer to model your service?
USDL-Editor □ □ Sophie
Question 5.3: Which Software-Tool do you prefer to compare your service with competitor services available on the service market place?
USDL-Editor □ □ Sophie
Question 5.3: Which Software-Tool do you prefer overall to perform all the tasks?
USDL-Editor □ □ Sophie
I like the features of SOPHIE good:
Your comments:
I don't like specific features of SOPHIE, which can be improved.
Your comments:

Evaluation of the Service Modeling Tools
(Variant A: Service Modeling with USDL-Editor)

In this evaluation, I want to evaluate the two service modeling tools "USDL-Editor" and "SOPHIE". For this purpose, I will introduce you to the application scenario for the evaluation first. Then you will be asked to perform a set of tasks. I will introduce you to the required functionality of each tool for each task before you start to perform the task. Afterwards, you will be asked to perform each task with both tools in order to compare both tools. After the completion of each task, you will be asked to fill out one questionnaire. At the end of the evaluation, you will be asked to fill out a final questionnaire.

Participant's Info

Participant number:
Age:
Gender:

Question 1.1: how good is your knowledge about PC?

No any knowledge □ □ □ □ □ Very good knowledge

Question 2.2: How good is your knowledge about services and service modeling?

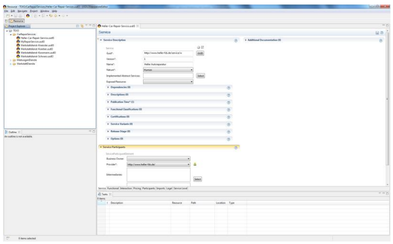
No any knowledge □ □ □ □ □ Very good knowledge

Application Scenario

You are the owner of a car repair shop. You are observing a new trend "service market places for car repair services" in the web. It can be a new channel for the acquisition of new customers to sell car repair services like car repairing and car rental. The new potential customer can be insurance companies, who are looking for authorized car repair shops. You decide to offer your car repair service "MyCarRepair-Service" on the service market place, which is specially designed for car insurance companies.

Task

Your task is to model a price plan for your car repair service “MyCarRepair-Service”. You use Unified Service Description Language Editor (USDL-Editor) as shown in the following figure to model a price plan for your car repair service. For this purpose, you have to perform the following sub-tasks.



Sub-task 1: Analyze car repair services available on the service market place

(Short introduction to USDL-Editor's functionality "analysis of existing services")

Find a service with the lowest price for the "Supported repair: car glass repair" and "Certification: TÜV" from the available car repair services on the service market place. To perform this sub-task, you have to follow the following steps:

1. Find services, which have the capability (Tab: Functional) "Supported repair: car glass repair" and "Certification: TÜV"
2. Compare the price for the found services to find the cheapest one (Tab: Pricing).
3. Write down the name of cheapest service and its prices

Questionnaire 1

Question 1.1: It was intuitive to analyze available service on the service market place.						
Not at all	□	□	□	□	□	Very much
Question 2.2: The support for the analysis of available service on the service market place was good.						
Not at all	□	□	□	□	□	Very much
Question 1.3: The navigation between services and their price plans was good.						
Not at all	□	□	□	□	□	Very much
Question 1.4: It was easy to get an overview about prices of services.						
Not at all	□	□	□	□	□	Very much
Question 1.5: It was easy to compare the prices of services.						
Not at all	□	□	□	□	□	Very much
Question 1.6: The support to find cheapest services was good.						
Not at all	□	□	□	□	□	Very much
Question 1.6: The overall support to perform this sub-task was good.						
Not at all	□	□	□	□	□	Very much

Sub-task 2: Model your own car repair service

(Short introduction to USDL-Editor's functionality "model a service")

Model your service "MyCarRepair-Service" with the lowest price and TÜV-certification for the car glass repair and add a new feature to your service "mobile car glass repair". To perform this sub-task, you should perform following steps.

1. Add new Service (right click on the "CarRepairService" -> New -> Others -> USDL3 -> USDL3 Description)
2. Add new price plan and fill out all required fields, which are marked with "*".
3. Define the price components (Prices according to service Kneissler as you found in the last sub-task)
   a. Labor value class 1 – 5
   b. Special work
4. Add price components to price plan
5. Add capability in the tab "Functional"
   c. Supported repair: Car glass repair
   d. Supported repair: Mobile car glass repair
   e. Certification: TÜV

Questionnaire 2

Question 2.1: It was intuitive to model your service.							
	Not at all	□	□	□	□	□	Very much
Question 2.2: The support to model your service was good.							
	Not at all	□	□	□	□	□	Very much
Question 2.3: The support to find the different parameters for the modeling of your service was good.							
	Not at all	□	□	□	□	□	Very much
Question 2.4: The support to see the competitor cheapest service during the modeling of your service was good.							
	Not at all	□	□	□	□	□	Very much
Question 2.5: The overall support to perform this sub-task was good.							
	Not at all	□	□	□	□	□	Very much

Sub-task 3:Re-analyze (Compare your car repair service with available car repair services on the service platform)

(Short introduction to USDL-Editor's functionality "re-analyze services")

Re-analyze your modeled service "MyCarRepair-Service" with competitor car repair services available on the service platform and find all car repair services, which are direct and indirect competitors of your service. To perform this sub-task, you should perform following steps.

1. Find services, which have the capability (Tab: Functional) "Supported repair: car glass repair"
2. Find all competitors
   a. Services with functionality "Supported repair: car glass repair" which have same or less price
      i. Write down the name of services
   b. Services with Functionality "Supported repair: car glass repair", which are more expensive
      i. Write down the name of services

Questionnaire 3

Question 3.1: It was intuitive to find all competitors of your service?							
	Not at all	□	□	□	□	□	Very much
Question 3.2: The support to compare your own service with other available services on the service market place was good.							
	Not at all	□	□	□	□	□	Very much
Question 3.3: It helped well to compare the unique selling points of your service with competitors.							
	Not at all	□	□	□	□	□	Very much
Question 3.4: It helped well to improve the market competitiveness of your service.							
	Not at all	□	□	□	□	□	Very much
Question 3.5: The overall support to perform this sub-task was good							
	Not at all	□	□	□	□	□	Very much

Final Questionnaire:

Question 5.1: Which Software-Tool do you prefer to analyze available services on the service market place?
USDL-Editor ☐ ☐ Sophie
Question 5.2: Which Software-Tool do you prefer to model your service?
USDL-Editor ☐ ☐ Sophie
Question 5.3: Which Software-Tool do you prefer to compare your service with competitor services available on the service market place?
USDL-Editor ☐ ☐ Sophie
Question 5.3: Which Software-Tool do you prefer overall to perform all the tasks?
USDL-Editor ☐ ☐ Sophie
I like the features of SOPHIE good:
Your comments:
I don't like specific features of Sophie, which can be improved.
Your comments:

Appendix B

CV and Publications

	Mohammad Nadeem Bhatti *January 10, 1974 in Faisalabad - Pakistan
1992	Abitur, Government College in Faisalabad, Pakistan
1992 - 1994	Studies of Electrical Engineering Americanos College in Nicosia, Cyprus
1995 - 1996	Studienkolleg at the Technische Universität Darmstadt, Germany
1996 - 2002	Studies of Electronics at the Technische Universität Darmstadt, Germany
since 2002	Research associate at the Fraunhofer Institute for Computer Graphics Research Darmstadt, Germany. Projects: • THESEUS (BMWi): New Technologies for the Internet of Services (IoS) • THESEUS-TEXO (BMWi): Infrastructure for Web-based services

1. Bhatti, Nadeem; Fellner, Dieter W. (2011): **Visual Semantic Analysis to Support Semi-Automatic Modeling of Service Descriptions**. In *Modern Software Engineering Concepts and Practices: Advanced Approaches.* Hershey, Pennsylvania : IGI Global, 2011

2. Oberle, Daniel; Bhatti, Nadeem; Brockmans, Saartje; Niemann, Michael; Janiesch, Christian (2009):**Countering Service Information Challenges in the Internet of Services.** In *Business & Information Systems Engineering.* 1 (2009), 5, pp. 370-390.

3. Oberle, Daniel; Bhatti, Nadeem; Brockmans, Saartje; Niemann, Michael; Janiesch, Christian (2009):**Effektive Handhabung von Service-Informationen im Internet der Dienste.** In *Wirtschaftsinformatik.* 51 (2009), 5, pp. 429-452.

4. Bhatti, Nadeem; Weber, Stefan Hagen (2009): **Semantic Visualization to Support Knowledge Discovery in Multi-Relational Service Communities.** In *Cruz-Cunha, Maria Manuela (Ed.); Oliveira, Eva Ferreira de (Ed.); Tavares, António José Videira (Ed.); Ferreira, Luis G. (Ed.): Handbook of Research on Social Dimensions of Semantic Technologies and Web Services.* Hershey, Pennsylvania : IGI Global, 2009, 23 p.

5. Burkhardt, Dirk; Nazemi, Kawa; Bhatti, Nadeem; Hornung, Christoph (2009): **Technology Support for Analyzing User Interactions to Create User-Centered Interactions.** In *HCI International 2009. Proceedings and Posters [DVD-ROM] : With 10 further Associated Conferences.* Berlin, Heidelberg, New York : Springer, 2009, LNCS 5614, pp. 3-12. (Lecture Notes in Computer Science (LNCS)).

6. Godehardt, Eicke; Bhatti, Nadeem (2008): **Using Topic Maps for Visually Exploring Various Data Sources in a Web-based Environment.** *In Maicher, Lutz (Ed.); Garshol, Lars Marius (Ed.): Scaling Topic Maps : Third International Conference on Topic Map Research and Applications, TMRA 2007.* Berlin; Heidelberg; New York : Springer, 2008, 6 p. (Lecture Notes in Computer Science (LNCS)).

7. Bhatti, Nadeem (2008): **Web Based Semantic Visualization to Explore Knowledge Spaces - An Approach for Learning by Exploring.** In *Luca, Joseph (Ed.); Weippl, Edgar R. (Ed.); Association for the Advancement of Computing in Education (AACE): Proceedings of ED-Media 2008 : World Conference on Educational Multimedia,* Hypermedia & Telecommunications [CD-ROM]. Chesapeake, 2008, pp. 312-317.

8. Nazemi, Kawa; Bhatti, Nadeem; Godehardt, Eicke; Hornung, Christoph (2007): **Adaptive Tutoring in Virtual Learning Worlds**. In *Montgomerie, Craig (Ed.); Seale, Jane (Ed.); Association for the Advancement of Computing in Education (AACE): Proceedings of ED-Media 2007 : World Conference on Educational Multimedia, Hypermedia & Telecommunications [CD-ROM].* Chesapeake, 2007, pp. 2951-2959.

9. Foroughi, Roya; Bhatti, Nadeem (2005): **Dynamic Test Generation, Visual Assessment and Modular Virtual Learning Worlds.** In *Richards, Griff (Ed.); Association for the Advancement of Computing in Education (AACE): E-Learn 2005. Proceedings : World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education.* Chesapeake, VA, USA : Association for the Advancement of Computing in Education (AACE), 2005, pp. 2923-2928.

10. 2005 Godehardt, Eicke; Bhatti, Nadeem; Hornung, Christoph (2005): **Keeping the Learning History.** In *Kommers, Piet (Ed.); Richards, Griff (Ed.); Association for the Advancement of Computing in Education (AACE): Proceedings of ED-Media 2005 : World Conference on Educational Multimedia, Hypermedia & Telecommunications [CD-ROM].* Norfolk, 2005, pp. 4466-4469.

11. Bhatti, Nadeem; Godehardt, Eicke; Hornung, Christoph (2005): **Modular Virtual Learning World.** In *Kommers, Piet (Ed.); Richards, Griff (Ed.); Association for the Advancement of Computing in Education (AACE): Proceedings of ED-Media 2005 : World Conference on Educational Multimedia, Hypermedia & Telecommunications [CD-ROM].* Norfolk, 2005, pp. 146-153.

12. Bhatti, Nadeem; Hornung, Christoph (2005): **Virtual Learning World.** In *TESI 2005 Conference Proceedings CD-ROM : Training, Education & Simulation International.* [Maastricht], 2005, 9 p.

List of Figures

List of Tables

Bibliography

[AA01] Ankolekar A, Hobbs JR Lassila O Martin DL McIlraith SA Narayanan S Paolucci M Payne TR Sycara KP Zeng H (2001). In: Proceeding of the first semantic web working symposium SWWS Stanford University California pp 4. Burstein MH M. Burstein MH: DAML-S: semantic markup for web services. In: *Proceeding of the first semantic web working symposium SWWS*, 2001

[ABG+04] Akkermans, H. ; Baida, Z. ; Gordijn, J. ; Pena, N. ; Altuna, A. ; Laresgoiti, I.: Using ontologies to bundle real–world services. In: *IEEE Intelligent Systems (EXPERT)* 19 (2004), S. 57–66

[Alt08] Alter, S.: Service system fundamentals: Work system, value chain, and life cycle. In: *IBM Systems Journal* 47 (2008), S. 71–85

[BC05] Becker, Peter ; Correia, Joachim H.: The ToscanaJ Suite for Implementing Conceptual Information Systems. Version: 2005. `http://dx.doi.org/10.1007/11528784_17`. In: *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005. – DOI 10.1007/11528784_17, S. p. 324–348

[BD06] Barros, A. ; Dumas, M.: The rise of Web service ecosystems. In: *IT Prof* 8 (2006), S. 31–37

[BDB05] Barros, A. ; Dumas, M. ; Bruza, P.: The move to Web service ecosystems. In: *BPTrends* (2005)

[BDH+08] Bitsaki, M. ; Danylevych, O. ; Heuvel, W. van d. ; Koutras, G. ; Leymann, F. ; Mancioppi, M. ; Nikolaou, C. ; Papazoglou, M.: An architecture for managing the lifecycle of business goals for partners in a service network. In:

LNCS: an architecture for managing the lifecycle of business goals for partners in a service network, 2008

[BDMDG08] Barrett, M. ; Davidson, E. ; Middleton, C. ; De-Gross, J. I.: *Information Technology in the Service Economy: Challenges and Possibilities for the 21st Century*. Springer Publishing Company, Incorporated, 2008

[BDQN02] Benatallah, Boualem ; Dumas, Marlon ; Quan, Sheng Z. ; Ngu, H. H. A.: Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services. In: *In: Proceedings of 18th international conference on data engineering (ICDE 02)*, 2002

[Ber83] Bertin, Jacques: *Semiology of graphics*. University of Wisconsin Press, 1983. – ISBN 0299090604

[Ber07] Berbner, Rainer: *Dienstgüteunterstützung für Service-orientierte Workflows*, Technischen Universität Darmstadt, Diss., 2007

[Ber08] Berre, A. J.: UPMS - UML profile and metamodel for services - an emerging standard. In: *Proceedings of the 12th international IEEE enterprise distributed object computing conference*, 2008

[BGA01] Baida, Z. ; Gordijn, J. ; Akkermans, H.: Service ontology. OBELIX Deliverable 6.1 / Free University Amsterdam. 2001. – Forschungsbericht

[Bha08] Bhatti, N.: Web Based Semantic Visualization to Explore Knowledge Spaces - An Approach for Learning by Exploring. In: *In Proceedings of ED-Media 2008: World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 2008

[BHL08] Buxmann, P ; Hess, T ; Lehmann, S: Software as a Service. In: *WIRTSCHAFTSINFORMATIK* 50(6) (2008), S. 500 – 503

[BMR+96] Buschmann, F. ; Meunier, R. ; Rohnert, H. ; Sommerlad, P. ; Stal, M: *Pattern-Oriented Software Architecture Volume 1 A System of Patterns.* Wiley, 1996

[BMW93] Biggerstaff, T. ; Mitbander, B. ; Webster, D.: The Concept Assignment Problem in Program Understanding. In: *In Proceedings of the 15th International Conference on Software Engineering*, 1993

[BPWR10] Baumann, C. ; Peitz, P. ; Wacker, R. ; Raabe, O.: Compliance for Service Based Systems through Formalization of Law. In: *Proceedings of the 6th Web Information Systems and Technologies (WEBIST2010)*, 2010

[BR08] Basole, R. C. ; Rouse, W. B.: Complexity of service value networks: Conceptualization and empirical investigation. In: *IBM Systems Journal* 47(1) (2008), S. 31–37

[BRS+08] Braun, Iris ; Reichert, Sandro ; Spillner, Josef ; Strunk, Anja ; Schill, Alexander: Zusicherung nichtfunktionaler Eigenschaften und Dienstgüte im Future Internet of Services. In: *PIK - Praxis der Informationsverarbeitung und Kommunikation.* Volume 31 (2008)

[Bry10] Bryer, Matthias: *Benutzerzentrierte Visualisierung von Informationsempfehlungen auf Basis von Recommender Systemen*, Technische Universität Darmstadt, Diplomarbeit, 2010

[BSW02] Bederson, B.B. ; Shneiderman, B. ; Wattenberg, M.: Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies. In: *ACM Transactions on Graphics (TOG)* 21 (4) (2002), S. 833–854

[BTN08] Bendaoud, Rokia ; Toussaint, Yannick ; Napoli, Amedeo: PACTOLE: A Methodology and a System for Semi-automatically Enriching an Ontology from a Collection of Texts. Version: 2008. `http://dx.doi.org/10.1007/978-3-540-70596-3_14`. In: Eklund, Ollivier Peter; H. Peter; Haemmerlé (Hrsg.): *Conceptual Structures: Knowledge Visualization and Reasoning* Bd. 5113/2008. Springer Berlin Heidelberg, 2008. – DOI 10.1007/978–3–540–70596–3_14, 203–216

[BW09] Bhatti, N. ; Weber, S. H.: Semantic Visualization to Support Knowledge Discovery in Multi-Relational Service Communities. In: *Handbook of Research on Social Dimensions of Semantic Technologies and Web Services.* IGI Global, 2009

[CBMK10] Cardoso, J. ; Barros, A. ; May, N. ; Kylau, U.: Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments. In: *Services Computing (SCC), 2010 IEEE International Conference on*, 2010

[CCMW01] Christensen, E. ; Curbera, F. ; Meredith, G. ; Weerawarana, S.: *Web Services Description Language (WSDL)*. 2001. – Retrieved October, 2009, from http://www.w3.org/TR/wsdl

[CES03] Cole, Richard ; Eklund, Peter ; Stumme, Gerd: DOCUMENT RETRIEVAL FOR E-MAIL SEARCH AND DISCOVERY USING FORMAL CONCEPT ANALYSIS. In: *Applied Artificial Intelligence* Volum 17 (2003), S. pp. 257–280

[Cha93] Chalmers, M: Using a Landscape to represent a corpus of documents. In: *In: Proceeding COSIT*, 1993

[Che06] Chesbrough, H. W.: Open innovation. The new imperative for creating and profiting from technology. In: *Harvard Business School Press, Boston, MA.* (2006)

[CHS05] Cimiano, P. ; Hotho, A. ; Staab, S.: Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis. In: *Journal of Artificial Intelligence research* 24 (2005), S. 305–339

[CHST04] Cimiano, Philipp ; Hotho, Andreas ; Stumme, Gerd ; Tane, Julien: Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies. Version: 2004. http://www.springerlink.com/content/4r62l72l3baayxdu/?p=d23d0d6442a244d8a6872c1f3813e8b2&pi=0. In: Eklund, Peter (Hrsg.): *Concept Lattices* Bd. 2961/2004. Springer Berlin / Heidelberg, 2004, 199–200

[CMS99] Card, S. ; Mackinlay, J. D. ; Shneiderman, B.: *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1999

[Con10] *ConWeaver*. RetrievedApril,2010fromhttp://conweaver.de. Version: April 2010

[CPSK07] Cios, Krzysztof J. ; Pedrycz, Witold ; Swiniarski, Roman W. ; Kurgan, Lukasz A.: *Data Mining: A Knowledge Discovery Approach.* Springer Science+Business Media, New York, USA, 2007

[CWR06] Cui, Q. ; Ward, M.O. ; Rundensteiner, E.A: Enhancing Scatterplot Matrices for Data with Ordering or Spatial Attributes. In: *In Proceedings of SPIE*, 2006

[CWV09] Cardoso, J. ; Winkler, M. ; Voigt, K: A Service Description Language for the Internet of Services. In: *First International Symposium on Services Science*, 2009

[DIN02] *DIN PAS 1 1018 Grundstruktur für die Beschreibung von Dienstleistungen in der Ausschreibungsphase. Berlin.* 2002. – DIN PAS 1 1018

[DKG08a] De Kinderen, S. ; Gordijn, J.: e3Service - a model-based approach for generating needs-driven e-service bundles in a networked enterprise. In: *Proceedings of 16th European conference on information systems*, 2008

[DKG08b] De Kinderen, S. ; Gordijn, J.: e3Service - an ontological approach for deriving multi-supplier IT-service bundles from consumer needs. In: *Proceedings of the 41st annual Hawaii international conference on system sciences*, 2008

[Dod07] Dodge, M.: Information Maps: Tools for Document Exploration. In: *Working Paper Series, CASA Working Paper*, 2007

[Dom00] Domik, G.: *Computergenerated Visualization, The Need for Visualization. Universität Paderborn.* 2000. – Retrieved April, 2010 from http://cose.math.bas.bg/Sci_Visualization/compGenVis/download/chapter1.pdf

[Duc07] Ducrou, Jon: *Design for Conceptual Knowledge Processing: Case Studies in Applied Formal Concept Analysis*, University of Wollongong, Diss., 2007

[DVGT10] De Virgilio, Roberto ; Giunchiglia, Fausto ; Tanca, Letizia: *Semantic Web Information Management.* Springer Berlin / Heidelberg, 2010

[DZ07] Domingue, J. ; Zaremba, M.: *Reference ontology for semantic service oriented architectures. OASIS Working Draft 0.1.* 2007. – Retrieved April, 2010 from http://www.oasis–open.org

[Eba10] *Ebay for Business Software.* SEPTEMBER 2010. – Retrieved August, 2010 from http://www.businessweek.com/magazine/content/05_38/b3951097.htm

[EC02] Eklund, Peter ; Cole, Richard: Structured Ontology and Information Retrieval for Email Search and Discovery. Version: 2002. `http://www.springerlink.com/content/bxx3d4t589ukm6r5`. 2002, 75+

[EGSW00] EKLUND, Peter ; GROH, Bernd ; STUMME, Gerd ; WILLE, Rudolf: *A Contextual-Logic Extension of TOSCANA*. 2000

[Emm05] EMMRICH, A.: *Ein Beitrag zur systematischen Entwicklung produktorientierter Dienstleistungen.* 2005. – University of Paderborn, Paderborn

[Fel98] FELLBAUM, C.: WordNet - An electronic lexical database. In: *MIT Press, Cambridge, Massachusetts and London, England* (1998)

[FG08] FERRARIO, R. ; GUARINO, N.: Towards an ontological foundation for services science. In: *Proceedings of future internet symposium 2008*, 2008

[Fie09] FIELD, Andy: *Discovering statistics using SPSS (3rd ed.)*. Sage Publications Ltd., 2009

[Fla10] *Flash Player*. `RetrievedApril,2010fromhttp://www.adobe.com/products/player_census/flashplayer/version_penetration.html`. Version: April 2010

[FLP+07] FENSEL, Dieter ; LAUSEN, Holger ; POLLERS, Axel ; BRUIJN, Jos d. ; STOLLBERG, Michael ; ROMAN, Dumitru ; DOMINGUE, John: *Enabling Seamntic Web Services: The Web Service Modeling Ontology.* Springer Berlin Hydelberg New York, 2007

[FMCNFV98] FERNANDEZ-MANJON, Baltasar ; CIGARRAN, Juan ; NAVARRO, Antonio ; FERNANDEZ-VALMAYOR, Alfredo: APPLYING FORMAL CONCEPT ANALYSIS TO DOMAIN MODELING IN AN INTELLIGENT HELP SYSTEM. In: CUENA, J. (. (Hrsg.): *Proceedings of Information Technology and Knowledge Systems (IT&KNOWS)*, 1998

[FR94] FOLEY, J. D. ; RIBARSKY, W.: Next-generation Data Visualization tools. In: *Scientific Visualization: Advances and Challenges* (1994), S. 103–126

[GB07] GODEHARDT, E. ; BHATTI, N.: Using Topic Maps for Visually Exploring Various Data Sources in a Web-Based Environment. In: *Lecture Notes in Computer Science: TMRA 2007.* Springer Berlin , Heidelberg, 2007

[GGM+02] GANGEMI, A. ; GUARINO, N ; MASOLO, C. ; OLTRAMARI, A. ; SCHNEIDER, L: Sweetening ontologies with DOLCE. In: *Proceedings of the EKAW 2002*, 2002

[God09] GODEHARDT, E.: *Kontextualisierte Visualisierung am Wissenintensiven Arbeitsplatz*, Technische Universität Darmstadt, Diss., 2009

[Gre00] GREENGRASS, E.: *Information Retrieval: A Survey.* `RetrievedOctober10,2009,http://clgiles.ist.psu.edu/IST441/materials/texts/IR.report.120600.book.pdf`. Version: 2000

[GSW05] GANTER, Bernhard ; STUMME, Gerd ; WILLE, Rudolf ; WILLE, Rudolf (Hrsg.): *Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence*. Bd. 3626: *Formal Concept Analysis - Foundations and Applications*. Springer Verlag / Heidelberg, 2005. – 349 S. `http://www.springerlink.com/content/cfafxdh4rpdy/?p=6664508e3dff4967bc7d63ba79f82fdb&pi=0`

[GW99] GANTER, Bernhard ; WILLE, Rudolf: *Formal Concept Analysis. Mathematical Foundations*. Springer Verlag, Berlin, 1999

[Haa04] HAAV, Hele-Mai: A Semi-automatic Method to Ontology Design by Using FCA. In: *International Workshop on Concept Lattices and their Applications CLA 2004*, 2004, S. p. 23 –24

[Har10] HARDING, C.: *Service-oriented architecture ontology. The Open Group Draft 2.0.* 2010. – Retrieved April, 2010 from http://www.opengroup.org/projects/soa-ontology/. Accessed 2009-06-15

[HF97] HAMP, B. ; FELDWEG, H.: Germanet – a lexical–semantic net for German. In: *ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, 1997

[HF06] HENRY, N. ; FEKETE, F.-D.: MatrixExplorer: a Dual-Representation System to Explore Social Networks. In: *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS* Vol. 12 (2006)

[HHN00] HAVRE, S. ; HETZLER, B. ; NOWELL, L: ThemeRiver: Visualizing Theme Changes Over Time. In: *IEEE Symposium on Information Visualization*, 2000

[HJK04] HESS, A. ; JOHNSTON, E. ; KUSHMERICK, N.: ASSAM: A Tool for Semi-Automatically Annotating Semantic Web Services. In: *In Proceeding third International Semantic Web Conference*, 2004

[HK01] Han, Jiawei ; Kamber, Micheline ; Cerra, Diane D. (Hrsg.): *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers, 2001

[HK03] Hess, A. ; Kushmerick, N.: Learning to Attach Semantic Metadata to Web Services. In: *In Procedding second International Semantic Web Conference*, 2003

[HK04] Hess, A. ; Kushmerick, N.: Machine Learning for Annotating Semantic Web Services. In: *In AAAI Spring Symposium on Semantic Web Services*, 2004

[HKRK07] Hamdy, Mohammed ; Koenig-Ries, Birgitta ; Kuester, Ulrich: Non-functional Parameters as First Class Citizens in Service Description and Matchmaking - An Integrated Approach. In: *In Proceeding International Conference on Service Oriented Computing*, 2007

[HM91] Helm, R. ; Maarek, Y.: Integrating Information Retrieval and Domain Specific Approaches for Browsing and Retrieval in Object-Oriented Class Libraries. In: *In Proceedings of Object-oriented Programming Systems, Languages, and Applications*, 1991

[HMS00] Herman, I. ; Melancon, G. ; S., Marshall M.: Graph Visualization and Navigation in Information Visualization A Survey. In: *IEEE Transactions on Visualzation and Computer Graphics* Vol 6, No. 1 (2000), S. 24–43

[HNP05] Hotho, A. ; Nürnberger, A. ; Paass, G.: A Brief Survey of Text Mining. In: *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology* 20 (2005), S. 19–62

[Hof10] Hofmann, Cristian E.: *PROCESS-BASED DESIGN OF MULTIMEDIA ANNOTATION SYSTEMS*, Technischen Universit¨at Darmstadt, Diss., 2010

[HV07] Heck, E. ; Vervest, P.: Smart business networks: how the network wins. In: *Communications of the ACM* Nr. 6 (2007), S. 28–37

[IoS10] *Internet of Services.* 2010. – Retrieved April, 2010 from http://www.internet-of-services.com

[iVi10] iViews: *iViews: Intelligent Views.* 2010. – Retreived April, 2010 from http://www.i-views.de

[JÖ8] Jöger, T.: Kommerzielle Applikationen für Open Source Software und deutsches Urheberrecht. In: *Softwarepatente – Web 2.0* 16 (2008), S. 61 – 78

[JNT08] Jia, Haibo ; Newman, Julian ; Tianfield, Huaglory: A new Formal Concept Analysis based learning approach to Ontology building. Version: 2008. `http://dx.doi.org/10.1007/978-0-387-77745-0-42`. In: Sicilia, Miltiadis D. Miguel-Angel; Lytras L. Miguel-Angel; Lytras (Hrsg.): *Metadata and Semantics*. Springer US, 2008. – DOI 10.1007/978–0–387–77745–0–42, Kapitel Ontology engineering and related techniques, 433–444

[JNT09] Jia, H. ; Newman, J. ; Tianfield, H.: A new Formal Concept Analysis based learning approach to Ontology building. In: *Metadata and Semantics*. Springer US, 2009

[JS02] J., Jin L. ; Sahai, V. M. A.: Analysis on Service Level Agreement of Web Services. / Software Technology Laboratories, HP Laboratories. 2002. – Forschungsbericht

[KAF+08] Keim, D. ; Andrienko, G. ; Fekete, J. D. ; Görg, C. ; Kohlhammer, J. ; Melancon, G.: Visual Analytics: Definition, Process , and challenges. In Journal,. In: *Information Visualization* (2008)

[Lar02] Larman, Craig: *Applying UML and Patterns: An Introduction to Object-Oriented Analysis And Desgin And the Unified Process*. Prentice HAll PTR, 2002

[LB09] Lehmann, S ; Buxmann, P: Pricing Strategies of Software Vendors. In: *Business & Information Systems Engineering* 6 (2009), S. 452–461

[Lin10] Lindig, Christian: *Colibri-Java*. August 2010. – Retrieved August, 2010 from http://code.google.com/p/colibri-java/

[Liv10] Liveplasma: *Liveplasma*. 2010. – Retrieved April 2010 from http://www.liveplasma.com/

[LKS+10] Landesberger, T. von ; Kuijper, A. ; Schreck, T. ; Kohlhammer, J. ; Wijk, J. J. ; Fekete, J.-D. ; Fellner, D. W.: Visual Analysis of Large Graphs. In: Hauser, Helwig (Ed.) et al.: Eurographics 2. (Hrsg.): *State of the Art Reports (STARs), Eurographics*, 2010

[Mac86] Mackinlay, Jock: Automating the design of graphical presentations of relational information. In: *ACM Trans. Graph.* 5 (1986), Nr. 2, S. 110–141. http://dx.doi.org/http://doi.acm.org/10.1145/22949.22950. – DOI http://doi.acm.org/10.1145/22949.22950. – ISSN 0730–0301

[Mae02] Maedche, Alexander: *Ontolgoy learning for the Semantic Web. Bosten.* Kluwer Academic Publischers., 2002

[MAKGM97] Mili, H. ; Ah-Ki, E. ; Godin, R. ; Mcheick, H.: Another nail to the coffin of faceted controlled-vocabulary component classification and retrieval. In: *SIGSOFT Software Engineering Notes* 22 (1997), S. 89 – 98

[MBK91] Maarek, Y. ; Berry, D. ; Kaiser., G.: An Information Retrieval Approach for Automatically Constructing Software Libraries. In: *IEEE Transactions on Software Engineering* 17(8) (1991), S. 800 – 813

[MKJ08] May, Thorsten ; Kohlhammer ; Jörn:: Towards Closing the Analysis Gap: Visual Generation of Decision Supporting Schemes from Raw Data. In: *Computer Graphics Forum* 27 (2008), S. 911 – 918

[MMM98] Mili, A. ; Mili, R. ; Mittermeir, R.: A survey of software reuse libraries. In: *Annals of Software Engineering* 5 (1998), S. :349 – 414

[MS01] Maedche, Alexander ; Staab, Steffen: Ontology Learning for the Semantic Web. In: *IEEE Intelligent Systems* 16 (2001), S. 1541–1672

[Mun97] Munzner, T.: Laying Out Large Directed Graphs in 3D Hyperbolic Space. In: *In Proceedings of the 1997 IEEE Symposium on Information Visualization*, 1997

[Nad04] Nadhan, E. G.: Service-Oriented Architecture: Implementation Challenges. In: Microsoft Architect Journal (2004), Nr. 2. In: *Microsoft Architect Journal* Nr. 2 (2004)

[NBB+10] Nazemi, K. ; Burkhardt, D. ; Bryer, M. ; Stab, C. ; Hofmann, C. ; Hornung, C.: *D.CTC.5.12.1: Design and conceptualization of Semantic Visualization Framework*. 2010

[NBH09] Nazemi, K. ; Breyer, M. ; Hornung, C.: SeMap: A Concept for the Visualization of Semantics as Maps In: HCI International 2009. Proceedings and Posters [DVD-ROM] :

With 10 further Associated Conferences. Berlin, Heidelberg, New York : Springer, 2009, LNCS 5616, pp. 83-91. (Lecture Notes in Computer Science (LNCS)). In: *LNCS: HCI International 2009*. Springer Berlin , Heidelberg, 2009

[Nor00] North, C.L.: *A User Interface for Coordinating Visualizations based on Relational Schemata: Snap-Together Visualization.. Doctoral Dissertation*, University of Maryland Computer Science Dept, Diss., 2000

[NR04] Noy, N. ; Rector, A.: *Defining N-ary Relations on the Semantic Web: Use With Individuals.* ,. W3C Technical Report, 2004. – Retrieved Oktober 2010 from http://www.w3.org/TR/2004/WD-swbp-naryRelations-20040721/

[Ò06] ÒSullivan, J: *Towards a precise understanding of service properties.* 2006. – Faculty of Information Technology. Queensland University of Technology

[OBB+09] Oberle, D. ; Bhatti, N. ; Brockmans, S. ; Niemann, M. ; Janiesch, C.: Countering Service Information Challenges in the Internet of Services. In: *Journal Business & Information Systems Engineering – Special issue Internet of services* (2009)

[OLG+06] Oberle, D. ; Lamparter, S. ; Grimm, S. ; Vrandecic, D. ; Staab, S. ; Gangemi, A.: Towards ontologies for formalizing modularization and communication in large software systems. In: *Applied Ontology* 2 (2006), S. 163–202

[OSEH05] Ó Sullivan, J. ; Edmond, D. ; Hofstede: Formal description of non–functional service properties. In: *Technical FIT–TR* 2005–01 (2005)

[OWL10] *OWL-S MX v2.* `RetrievedApril,2010fromhttp://www-ags.dfki.uni-sb.de/~klusch/owls-mx/`. Version: 2010

[PF09a] Pan, T. ; Fang, K.: Ontology-Based Formal Concept Differences Analysis in Radiology Report Impact by the Adoption of PACS. In: *Proceedings of the 7th International Conference on Formal Concept Analysis*, 2009

[PF09b] Pan, Telung ; Fang, Kwoting: Ontology-Based Formal Concept Differences Analysis in Radiology Report Impact by the Adoption of PACS. Version: 2009. `http://dx.doi.org/`

10.1007/978-3-642-01815-2-20. In: FERRÉ, Sebastian Sébastien; R. Sébastien; Rudolph (Hrsg.): *Formal Concept Analysis* Bd. 5548. Springer Berlin / Heidelberg, 2009. – DOI 10.1007/978–3–642–01815–2–20, 267–283

[PMLZ08] PETRIE, C. ; MARGARIA, T. ; LAUSEN, H. ; ZAREMBA, M.: *Semantic Web Services Challenge: Results from the First Year*. Springer Publishing Company, 2008

[POSV04] PATIL, A. ; OUNDHAKAR, S. ; SHETH, A. ; VERMA, K.: METEOR-S Web service Annotation Framework. In: *In Proceedings of the 13th International World Wide Web Conference (WWW 04)*, 2004

[PPS10] PANDIT, B. ; POPESCU, V. ; SMITH, V.: *SML service modeling language*. 2010. – Retrieved April, 2010 from http://www.w3.org/TR/sml/. Accessed 2009-06-15

[Pre10] *Prefuse visualization toolkit*. RetrievedApril, 2010fromhttp://prefuse.org/. Version: April 2010

[Pri06] PRISS, Uta: Formal Concept Analysis in Information Science. Version: 2006. http://www.upriss.org.uk/top/research.html. In: CRONIN, Blaise (Hrsg.): *Annual Review of Information Science and Technology* Bd. 40. Cronin, Blaise, 2006, Kapitel Formal Concept Analysis in Information Science., 521–543

[Pri09] PRISS, Uta: *Formal Concept Analysis Software. http://www.upriss.org.uk/fca/fcasoftware.html. Accessed 2009-09-13*. 2009

[RBM+06] ROMAN, D. ; BRUIJN, J. de ; MOCAN, A. ; LAUSEN, H. ; DOMINGUE, J. ; BUSSLER, C. ; FENSEL, D.: WWW: WSMO, WSML, and WSMX in a nutshell. In: *Proceedings of the XIX. international conference of RESER*, 2006

[RBRK09] RIEDL, Christoph ; BOEHMANN, Tilo ; ROSEMANN, Michael ; KRCMAR, Helmut: Quality management in service ecosystems. In: *Inf Syst E-Bus Manage*, 2009

[RMF+09] RIEDL, Christoph ; MAY, Norman ; FINZEN, Jan ; STATHEL, Stephan ; LEIDIG, Torsten ; KAUFMAN, Viktor ; BELECHEANU, Roxana ; KRCMAR, Helmut: Managing Service Innovations with an Idea Ontology. In: *In proceedings of XIX International RESER Conference, Budapest, Hungary.*, 2009

[RT81] REINGOLD, E.M. ; TILFORD, J.S.: Tilford, Tidier Drawing of Trees. In: *IEEE Trans. Software Eng.* vol. 7, no. 2 (1981), S. 223–228

[Rul08] RULLKÖTTER, L: Preismanagement – Ein Sorgenkind? In: *Zeitschrift für Controlling und Management* 52(2) (2008), S. 92 – 98

[RWB+10] RAABE, O. ; WACKER, R. ; BAUMANN, C. ; FUNK, C. ; OBERLE, D.: Lawful Service Engineering – Formalisierung des Rechts im Internet der Dienste. In: *In: Proceedings of Internationales Rechtsinformatik Symposium (IRIS2010). Salzburg Austria*, 2010

[Sab05] SABOU, M.: Visual Support for Ontology Learning: an Experience Report. In: *Proceeding of the ninth International Conference on Information Visualization*, 2005

[Sab06] SABOU, Marta: *Building Web Service Ontologies*, Dutch Graduate School for Information and Knowledge Systems, Diss., 2006

[SBN+10] STAB, C. ; BREYER, M. ; NAZEMI, K. ; BURKHARDT, B ; HOFMANN, C. ; FELLNER, D.: SemaSun: Visualization of Semantic Knowledge based on an improved Sunburst Visualization Metaphor. In: *In Proceedings of ED-Media 2010: World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 2010

[SD08] SIMOVICI, Dan A. ; DJERABA, Chabane ; JIAN, Lakhmi (Hrsg.): *Mathematical Tools for Data Mining*. Springer Verlag London, UK, 2008

[Shn96] SHNEIDERMAN, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: *In proceedings of the 1996 IEEE Symposium on Visual Languages*, 1996

[Sim92] SIMON, H: Preismanagement: Analyse, Strategie, Umsetzung. In: *2nd edn. Gabler, Wiesbaden* (1992)

[SM07] SHEN, Z. ; MA, K. L.: Matrix Visualization. In: *Eurographics/ IEEE-VGTC Symposium on Visualization*, 2007

[Som04] SOMMERVILLE, I.: *Software Egineering*. Pearson/Addison-Wesley, Boston, 2004

[SP07] SABOU, M. ; PAN, J.: Towards semantically enhanced Web service repositories. Web Semantics. In: *Services Agents World Wide Web* (2007)

[SPS10] *SPSS*. RetrievedApril,2010fromhttp://www.spss.com/. Version: 2010

[Sun10] SunBurst: *SunBurst*. 2010. – Retrieved April, 2010 from http://www.cc.gatech.edu/gvu/ii/sunburst/

[SWGS05] Sabou, M. ; Wroe, C. ; Goble, C. ; Stuckenschmidt, H.: Learning domain ontologies for Semantic Web service descriptions. In: *J. Web Semantics* 3 (4) (2005)

[TEX10] *THESEUS-TEXO*. RetrievedApril,2010fromhttp://www.theseus-programm.de/en-us/theseus-application-scenarios/texo/default.aspx. Version: 2010

[Tou10] TouchGraph: *TouchGraph Google–Browser*. 2010. – Retrieved April, 2010 from http://www.touchgraph.com/TGGoogleBrowser.html

[UML10] UML: *Unified Modeling Language (UML)*. 2010. – Retrieved April, 2010 from http://www.uml.org

[War04] Ware, C: *Information Visualization*. Morgan Kaufmann San Francisco, 2004

[WCS08] Winkler, Matthias ; Cardoso, Jorge ; Scheithauer, Gregor: Challenges of business service monitoring in the internet of services. In: *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*. New York, NY, USA : ACM, 2008 (iiWAS '08). – ISBN 978–1–60558–349–5, 613–616

[Wer85] Wersig, G.: *Thesaurus-Leitfaden*. K.G. Saur Verlag KG, 1985

[WFM99] *Workflow Management Coalition: Terminology & Glossary*. 1999. – Retrieved December, 2010 from http://www.internet-of-services.com

[WGG+04] Wroe, C. ; Goble, C. ; Greenwood, M. ; Lord, P. ; Miles, S. ; Papay, J. ; Payne, T. ; Moreau, L.: Automating Experiments Using Semantic Data on a Bioinformatics Grid. In: *IEEE Intelligent Systems* 19(1) (2004), S. 48 – 55

[WH02] Wermter, S. ; Hung, C.: Selforganizing classification on the Reuters news corpus. In: *In proceedings of the 19th International Conference on Computational Linguistics*, 2002

[Wij05] Wijk, J.J. van: The value of visualization. In: *IEEE Visualization* (2005), S. 11

[Wil82] Wille, Rudolf: Restructuring lattice theory: an approach based on hierarchies of concepts. In: *Ordered sets*. Reidel, Dordrecht-Boston, 1982, S. 445 – 470

[Wol94] Wolff, Karl E.: A first Course in Formal Concept Analysis. In: *Proceedings SoftStat*, 1994

[WWWC08] Wei, D. ; Wang, T. ; Wang, J. ; Chen, Y: Extracting Semantic Constraint from Description Text for Semantic Web Services Discovery. In: *In proceedings 7th International Semantic Web Conference*, 2008

[ZBZ02] Zhou, L. ; Booker, Q. ; Zhang, D.: ROD - towards rapid ontology development for underdeveloped domains. In: *Proceeding 35 Hawaii International Conference on System Sciences*, 2002

[Zho07] Zhou, Lina: Ontology learning: state of the art and open issues. In: *Information Technology and Management* Volume 8, Number 3 / September 2007 (2007), 241–252. http://dx.doi.org/10.1007/s10799-007-0019-5. – DOI 10.1007/s10799–007–0019–5

[ZMC05] Zhao, S. ; McGuffin, M.J. ; Chignell, M.H.: Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams. In: *Proceedings of the IEEE Symposium on Information Visualization*, 2005

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind als soche kenntlich gemacht.

Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Nadeem Bhatti