FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Moving e-Business to the Next Level:

Regulated and Informed Trading via Virtual Organizations and Hybrid Semantic Search

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der technischen Wissenschaften

by

Markus Gärtner
Registration Number 0226171

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: ao. Univ. Prof. Dipl.-Ing. Dr. techn. Andreas Rauber

The dissertation has been reviewed by:

(ao. Univ. Prof. Dipl.-Ing. Dr. techn. Andreas Rauber - Vienna University of Technology)	(ao. Univ. Prof. Dipl.-Ing. Dr. techn. Josef Küng - Johannes Kepler University Linz)

Wien, 20.02.2012

(Markus Gärtner)

Erklärung zur Verfassung der Arbeit

Markus Gärtner
Kienmayergasse 4/8, 1140 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser)

Acknowledgements

I would like to thank everyone who supported me during my doctoral study. Special thanks goes to my parents, Helga and Hans, my fiancé, Christine, my supervisor, Andreas, my friend and colleague, Ingo and our project leader, Helmut.

Kurzfassung

Immer mehr Handel findet über das Internet statt. In den letzten Jahren wurden zahlreiche Plattformen die den Austausch von Gütern und Leistungen über das Internet ermöglichen eingeführt. Eine der am weitesten verbreiteten Plattformen ist der elektronische Markt. Das Hauptmerkmal eines elektronischen Marktes ist, dass im Gegensatz zum elektronischen Shop eine Vielzahl von Verkäufer ihre Produkte einer Vielzahl von Kunden anbieten. Des Weiteren können Nutzer eines elektronischen Marktes sowohl als Verkäufer als auch als Kunden auftreten. Diese Plattform findet weite Verwendung in der B2B (Unternehmen zu Unternehmen) und C2C (Privatkunden zu Privatkunden) Domäne. Beispiele für elektronische Marktplätze sind eBay[1], Alibaba[2] und EC21[3].

Um an einem elektronischen Markt teilzunehmen müssen die internen und externen Geschäftsprozesse einer Organisation angepasst oder zusätzliche erstellt werden, damit ein elektronsicherer Austausch von Gütern und Leistungen erfolgen kann. Das Problem ist jedoch, dass der Aufwand für die Regulierung und Anpassung dieser Geschäftsprozesse mit steigender Komplexität tendenziell ansteigt. Dies ist speziell bei jenen Organisationen der Fall, welche auf einem elektronischen Markt mit einer Vielzahl anderer Organisationen konkurrieren. Diese Arbeit konzentriert sich unter anderem auf die Regulierung von elektronischen Märkten mittels virtueller Organisationen. Diese virtuellen Organisationen basieren auf elektronischen Institutionen - einer Multi-Agent System Methodologie. Die Hauptprobleme von elektronischen Märkten sind meist, dass diese aufgrund ihrer Komplexität nicht ausreichend reguliert sind, was zu Betrug und einseitigem Handel führen kann. Oder dass diese zu restriktiv sind, wodurch kein effizienter Austausch von Gütern und Leistungen zustande kommt. Ein weiteres Problem von elektronischen Märkten ist, dass die Art der Interaktion mit einem elektronischen Markt für einige Benutzer eine Hemmschwelle darstellt, da jene häufig Interaktionsmuster aus der realen Welt gewohnt sind. Für diese Benutzer sollte die Interaktion mit dem elektronischen Markt so einfach wie möglich gehalten und gewohnte Interaktionsmuster aus der realen Welt imitiert werden.

[1] http://www.ebay.com/
[2] http://www.alibaba.com/
[3] http://www.ec21.com/

Teilnehmer des Marktes benötigen relevante Informationen um entscheiden zu können welche Güter und Leistungen sie zu welchem Zeitpunkt von wem beziehen. Kurz gesagt, der Benutzer, die Benutzerin benötigt relevante und akkurate Informationen um informierte Entscheidungen zu treffen. In den letzten Jahren ist das Internet zu einer der Haupt-Informationsquellen avanciert. Die Verwendung des World Wide Webs, ein über Hypertext Dokumente verbundenes Sub-Netz des Internets, als Informationsquelle stellt jedoch einige Probleme dar. Hauptprobleme sind die immer weiter wachsende Datenflut, welche es erschwert relevante Information von irrelevanter Information zu unterscheiden, sowie die Zerstreuung von Information über eine Vielzahl unterschiedlicher Web Sites. Im Gegensatz dazu stellt das Semantic Web, eine Erweiterung des World Wide Web, strukturierte Information in maschinen-verarbeitbarer Form zu Verfügung. Der Vorteil des Semantic Webs gegenüber dem World Wide Web ist, dass aufgrund der strukturierteren Information zielgenauer zugegriffen werden kann, wodurch genauere Suchresultate ermöglicht werden. Nichtsdestotrotz kann das Semantic Web hinsichtlich der verfügbaren Datenmenge nicht mit dem World Wide Web konkurrieren.

Um die genannten Probleme zu adressieren, konzentriert sich diese Arbeit (1) auf die Erarbeitung und Implementierung von Lösungen zur Formierung und Regulierung von elektronischen Märkten, sowie (2) die Entwicklung und Implementierung von Lösungen, um relevante Informationen mittels hybriden Suchverfahren und intuitiven Abfragemechanismen Benutzern zugänglich zu machen.

Hinsichtlich der Formierung und Regulierung elektronischer Märkte wurden folgende Lösungen erarbeitet: 1.a) ein Prozess mit Hilfe dessen Organisationen komplexe kundenrelevante Prozesse unkompliziert mittels Software Agenten implementieren können und 1.b) die Implementierung eines Frameworks mittels dessen virtuelle Organisationen, welche aus mehreren elektronischen Institutionen bestehen, von Organisationen erstellt werden können. Diese virtuellen Organisationen stellen Leistungen mehrerer Organisationen in gebündelter Form auf einem regulierten elektronischen Markt zu Verfügung. Des Weiteren bietet das Framework die Möglichkeit, den elektronischen Markt und dessen virtuelle Organisationen innerhalb einer 3D Welt zu visualisieren.

Hinsichtlich der Nutzung hybrider Verfahren zur Informationsgewinnung und benutzerfreundlicher Abfragemechanismen wurden folgende Lösungen erarbeitet: 2.a) ein hybrider semantischer Suchansatz und dessen Referenz-Implementierung in Form des Hybrid Semantic Search Systems (HS3), welches über ein kombiniertes Suchverfahren auf Informationen des World Wide Webs und Semantic Webs zurückgreifen kann und 2.b) eine neuartige interaktive Eingabemethode, welche die Schlichtheit von textbasierten Suchmaschinen wie Google[4], Yahoo[5] und Bing[6] mit der Ausdrucksstärke von semantischen Suchmaschinen kombiniert. Unter Zuhilfenahme dieser interaktiven Ein-

[4]http://www.google.com/
[5]http://www.yahoo.com/
[6]http://www.bing.com/

gabemethode ist es Nutzern möglich ohne Kenntnis der zugrundeliegenden Ontologie oder Wissensbasis komplexe semantische Suchanfragen zu formulieren. Des Weiteren wird im Rahmen der Arbeit die Implementierung der interaktiven Eingabemethode in HS3 präsentiert. Um die Anwendbarkeit der Eingabemethode zu prüfen wurde eine Benutzerevaluierung durchgeführt, deren Ergebnisse in dieser Arbeit präsentiert werden. Abschließend werden die Ergebnisse einer Precision, Recall und Performance Evaluierung des Systems präsentiert und mit den Ergebnissen eines ähnlichen Systems verglichen.

Abstract

Business is increasingly conducted over the Internet. In recent years various platforms have been established that support the exchange of goods and services over the Internet. One of the most widespread platforms is the Electronic Marketplace. An Electronic Marketplace, in contrast to an electronic shop, is a platform where multiple sellers offer products and services to multiple customers. Electronic Marketplaces are very common in the Business to Business (B2B) domain and Customer to Customer (C2C) domain. Examples for Electronic Marketplaces are eBay[7], Alibaba[8] and EC21[9].
To participate on an Electronic Marketplace the internal and external business processes of an organization need to be adapted or new ones created for doing business electronically (e-Business), but the more complex processes become, the more alignment and regulation effort is needed. This holds especially true when an organization is competing on an Electronic Marketplace with multiple other organizations. The main drawbacks of Electronic Marketplaces are that they are either insufficiently regulated, resulting in fraud or biased trading, or that they are too restrictive which prohibits participants from efficiently exchanging goods or services. Another issue with Electronic Marketplaces is that the non-standard interaction with the market and its participants might represent a barrier to people who are used to interaction patterns from the real world. For these people the interaction with the market should be kept as simple as possible and resemble the interaction patterns people are used to from real life.
Users need relevant information to decide what to buy, when to buy and from whom to buy to benefit from their participation in the market. In a nutshell, users need relevant and accurate information to base their decisions on. In recent years the Internet has become one of the main sources for information. However, information on the World Wide Web, a subnet of the Internet, suffers from issues such as the sheer amount of ever expanding unstructured data, the difficulty to differentiate between relevant and irrelevant information and that information is scattered over multiples Web Sites. In contrast, the Semantic Web, which is an extension of the World Wide Web, offers structured information that is interpretable by machines. The advantage of the Semantic Web

[7]http://www.ebay.com/
[8]http://www.alibaba.com/
[9]http://www.ec21.com/

over the World Wide Web is that it can be accessed more precisely and more accurate searches can be conducted due to its structured nature. However, the Semantic Web cannot compete with the World Wide Web in terms of pure data amount.
To address the identified issues this work concentrates on (1) the development and implementation of solutions for the formation and regulation of Electronic Marketplaces and (2) the formulation and implementation of approaches to acquire and search relevant information by means of a hybrid search mechanism and a user-friendly interactive ontology-aware user interface.
Regarding the issues that arise when forming and regulating an Electronic Marketplace, the following solutions have been realized : 1.a) an implementation process that helps organizations to implement their customer centric business processes via software agents in a straight forward manner and 1.b) the implementation of a general framework that enables organizations to build so-called Virtual Organizations which consist of several interacting Electronic Institutions that coherently provide services to customers in a regulated Electronic Marketplace. The presented approach provides the possibility to visualize the Electronic Marketplace comprising the Virtual Organizations within a 3D Virtual World. Every Virtual Organization can be visualized as building in the 3D Virtual World. By visualizing a Virtual Organization as building and customer centric business processes as agents within a 3D Virtual World, people can use interaction patterns they are used to from the real world to interact with the Electronic Marketplace.
Regarding the issues that arise when relevant information needs to be acquired, the following solutions have been realized : 2.a) a hybrid semantic search approach has been developed and implemented in form of the Hybrid Semantic Search System (HS^3), that leverages information from the World Wide Web and the Semantic Web and 2.b) a novel interactive input mechanism that combines the clean and concise input mechanisms of keyword-based search engines such as the ones used by Google[10], Yahoo[11] and Bing[12] with the expressiveness of the input mechanisms provided by Semantic Search Engines has been developed and implemented in HS^3. In addition, the presented interface can be used to interactively formulate queries without prior knowledge of the underlying ontology. To assess the applicability of the novel interface for the average Internet user we conducted a usability study and present the results in this work. Finally, we present the results of a precision, recall and performance evaluation of HS^3 and compare them with a similar system.

[10]http://www.google.com/
[11]http://www.yahoo.com/
[12]http://www.bing.com/

Contents

Contents **1**

1 Introduction **5**
1.1 Regulating trade: Creating an environment for e-Business . . . 7
1.1.1 Research questions and contributions . . . 9
1.2 Informed trading: Acquiring and searching relevant information . . . 11
1.2.1 Research questions and contributions . . . 12

2 Regulating trade
Creating complex regulated e-Markets by means of Virtual Organizations **17**
2.1 Introduction . . . 17
2.2 Related Work . . . 22
2.3 Electronic Institutions . . . 28
2.4 The 3D Electronic Institution Implementation Process . . . 30
2.5 Framework Description . . . 31
2.5.1 Enabling the formation of VOs . . . 32
2.5.2 Connecting VOs with the 3D Virtual World . . . 33
2.6 Going 3-dimensional in e-Business . . . 35
2.6.1 Creating a 3D Electronic Institution . . . 35
2.6.2 The Itchy Feet EM: Putting VOs to practice . . . 43
2.6.3 Auctions in the 3D Virtual World . . . 53
2.7 Discussion . . . 57
2.7.1 Enabling inter-EI communication . . . 57
2.7.2 Pitfalls . . . 59
2.7.3 Mapping buildings to EIs . . . 60
2.7.4 Usability . . . 62
2.7.5 Suitability of VOs to regulate EMs . . . 63
2.8 Summary . . . 64

3 Acquiring relevant information
Acquiring relevant information by means of Hybrid Semantic Search 67
3.1 Introduction . . . 67
3.1.1 Our contribution . . . 70
3.2 Related Work . . . 71
3.2.1 Semantic Search Systems and Hybrid Search Systems . . . 71
3.2.2 Hybrid search approaches . . . 75
3.2.3 Interaction with Semantic Search Systems . . . 77
3.3 Why a Hybrid Semantic Search System? . . . 78
3.3.1 Characteristics that hinder the use case realization . . . 80
3.4 Bridging structured and unstructured data . . . 81
3.4.1 The Hybrid Semantic Search System HS^3 . . . 88
3.5 Interactive ontology-aware Query Formulation . . . 108
3.6 Putting hybrid semantic search to practice . . . 113
3.7 Evaluation & Deployment . . . 116
3.8 Summary . . . 134

4 Conclusion & Outlook 137

A Details on the EIs forming the Itchy Feet EM 143
A.1 The Auction Electronic Institution . . . 143
A.1.1 Clearing Scene . . . 143
A.1.2 Auction Scene . . . 144
A.1.3 Offering Scene . . . 145
A.1.4 Information Scene . . . 146
A.2 The Travel Agency Electronic Institution . . . 146
A.2.1 Booking Scene . . . 147
A.2.2 Clearing Scene . . . 148
A.2.3 Information Scene . . . 149
A.2.4 Sale Scene . . . 150
A.3 The Ether Electronic Institution . . . 150
A.3.1 Ether Scene . . . 151
A.3.2 Registration Scene . . . 152
A.3.3 Validation Scene . . . 153
A.4 The Forum Electronic Institution . . . 153
A.4.1 Forum Scene . . . 154

B SerQL Queries for the Performance Evaluation 155

Bibliography 163

CHAPTER 1

Introduction

Many different interpretations of e-Business, an abbreviation for Electronic Business, exist. Two common interpretations of e-Business are that it centralizes all computerized and semi-computerized inner processes of a company that are needed to do business and that e-Business is the electronic trade between two or more business partners. In this work we will cover both interpretations, because both definitions express the main intent behind e-Business - “doing business electronically”. The former interpretation concentrates on the inner processes of a company that form the foundation to do business electronically and the latter contemplates the interactions with other companies or consumers to do business. Both aspects are important for a company to generate revenue and to be competitive on the market. The form of inner processes depends to a great extent on the domain a company operates in. For example, a telecommunication company employs different inner processes to generate revenue than a banking company does. A telecommunication company employs processes that deal with contract prolongations and tariff changes, whereas a bank employs processes that deal with lending money and private equities.

However, there are processes that are similar in most companies such as supply chain management processes, human resource management processes and sales processes. This holds also true for external processes, which are needed to interact with suppliers and customers. To exchange goods and services efficiently, companies need suitable platforms that offer the functional and technical capabilities to facilitate trade between a multitude of participants. Electronic Marketplaces have emerged as one of the dominant platforms to facilitate trade between multiple parties.

Various interpretations of an Electronic Marketplace (EM) exist as well, of which two have been formulated by Grieger [54], which we will use in this thesis. He defines an

EM as an institutional medium that assigns different roles in a community, facilitates the exchange of information such as goods, services and payment and provides an infrastructure with protocols and processes to regulate the interaction in this community. The second definition describes an EM as a social community consisting of buyers and sellers which can be described by a certain condition that can be changed through market transactions according to the intentions of the participants. Possible conditions include the participants' knowledge, intention, contracts or goods at a certain time. The most salient characteristic of an EM is that multiple buyers and sellers conduct business, whereas on other platforms only one seller and multiple buyers or one buyer and multiple sellers conduct business. One of the most prominent EMs is eBay[1] - an auction based marketplace where multiple buyers and sellers trade their products. In contrast, the Apple Store[2] is not an EM, because customers may only buy products offered by a single seller.

Wang et al. [95] did a comprehensive literature review of EM research and ascertained that most marketing and economic researchers studied decentralized Electronic Markets, but paid less attention to central platforms. Hence, the usage of central platforms to support the exchange of products and services among multiple participants is still unexplored to a great extent. EMs are not only distinguished by their structure, but they can also be distinguished by the participating parties - whether both emerge from the same domain such as the consumer domain (C2C) or the business domain (B2B) or from different domains such as business to consumer (B2C) or government to business (G2B). Companies that do business electronically by using computerized or semi-computerized processes, need to take care that internal and external processes are harmonized. Internal processes create products or services or support their creation. External processes ensure that goods and services are distributed to consumers or that products and services needed by the company can be acquired from suppliers. Internal and external processes need to be bridged efficiently for a company to be successful and generate revenue. However, the more complex processes become, the more alignment and regulation effort is needed. This holds especially true when a company is competing on an EM with multiple other companies. The main drawbacks of EMs are that they are either insufficiently regulated, resulting in fraud or biased trading, or that they are too restrictive which prohibits participants from efficiently exchanging goods and services. In addition, some of the regulation mechanisms are too complex, resulting in error-prone markets or an inefficient exchange of goods due to performance issues. Another issue of EMs is that the non-standard interaction with the market and its participants represents a handicap to people who are used to interaction patterns from the real world. For these people the interaction with the

[1] http://www.ebay.com

[2] http://store.apple.com/

market should be kept as simple as possible and resemble the interaction patterns people are used to from their daily life.

To fully leverage the possibilities offered by an EM and to get the biggest benefit out of it, customers need to make informed decisions. Hence, customers need to know the answer to questions such as “what to buy”, “when to buy”, “from whom to buy” and “at which price to buy”. On the other hand sellers need to know the answer to question such as “what to sell”, “when to sell”, “to whom to sell” and “at which price to sell”. In a nutshell users of the Electronic Marketplace need relevant information to base their decisions on. Relevant information is crucial to make informed decisions. The Internet has become one of the most prominent information sources in recent years. However, information on the Internet suffers from several problems such as that most information is only available in the form of unstructured data and that the information is unfiltered and spread over multiple sites. To make use of this information it needs to be gathered, parsed, structured to a certain degree, consolidated and made searchable in an efficient way.

Therefore, this work focuses on the topics *Regulating trade* and *Informed trading*. Both topics will be presented in more detail in the following sections followed by the definition of research questions, methods and the presentation of our contributions.

1.1 Regulating trade: Creating an environment for e-Business

Multiple approaches to model and regulate the internal and external processes of an organization exist. One of the more recent forms makes use of Multi-Agent Systems (MAS). Several MAS methodologies such as GAIA [58, 98] and Electronic Institutions [30] have been proposed that facilitate the creation of complex Multi-Agent Systems. Electronic Institutions share similarities to real world institutions in that certain rules and norms exist that participants of the institution have to adhere to. In this work we concentrate on the regulation of an organization’s internal and external processes by means of a Multi-Agent System, because Multi-Agent Systems have proven to be a suitable paradigm for modeling environments that are composed of many autonomous individuals. Furthermore, a comprehensive MAS methodology and framework enable organizations to precisely model their internal and external processes, avoiding modeling errors that would eventually lead to fraud or inefficient exchange of goods on the EM.

However, when modeling Electronic Institutions, constituting the internal processes of a company, from scratch, their definition gets complex and error-prone as they grow in size. Therefore, a straight forward modeling approach that resembles the modeling approach of a real world institution would be beneficial. Furthermore, to form an EM, multiple companies, which have defined their internal processes by means of Electronic Institutions, need to interact. The external processes of these companies are the interfaces between the company's representation as Electronic Institution and the EM. Currently there exists no out-of-the-box solution to connect multiple Electronic Institutions to form an EM. Furthermore, to make use of synergies some organizations form so-called Virtual Organizations. A Virtual Organizations can be regarded as a group of organizations that conjointly conduct business as one organization on a market. The construct of a Virtual Organization is also desirable for EMs. However, for the formation of Virtual Organizations the communication between member organizations is vital. Therefore, a communication facility between Electronic Institutions is needed to enable the formation of Virtual Organizations on an EM.

To reduce the gap between business taking place in the real world and business happening online several approaches have been proposed. Most approaches seek to create Web User Interfaces that help people who are not familiar with online trading to exchange goods or services. However, these approaches require people to get comfortable with an interface that does not reflect their usual interaction pattern of doing business in the real world. Recently, approaches that use virtual environments to simulate the real world to overcome the barrier between real world and online trading have been proposed. One of the most promising approaches are 3D Electronic Institutions [14] which combine the advantages of 3D Virtual Worlds and Electronic Institutions.

However, several issues need to be dealt with when a virtual environment is used for doing business. A major challenge is to keep the underlying system (e.g. a MAS) synchronous with the state of the virtual environment and vice versa. The representation of an EM as virtual environment is a promising approach to lower the barrier between people who are used to do business in the real world and online trading [8]. In addition, the visualization of an EM in a 3D Virtual World entails several unique advantages such as that it facilitates sophisticated visualization of products, it provides transparent and unified access to disparate multi-media information sources and it can act as a community facilitator to create and establish a lively and sustainable community involving both providers and consumers. Providing participants of the Electronic Market with such features can lead to an improved user acceptance of the market. The usage of 3D Virtual Worlds to provide users with new possibilities and to demonstrate their advantages has been researched in [47].

1.1.1 Research questions and contributions

Based on the previously identified issues the following research questions have been defined :

Research Question 1 *Which communication mechanism is suitable for information exchange in Virtual Organizations that are based on Electronic Institutions?*

The Electronic Institution methodology does not specify a communication mechanism for inter-Electronic Institution communication. Inter-Electronic Institution communication is needed to exchange information in Virtual Organizations that consist of multiple Electronic Institutions. As part of this work we propose communication mechanisms for information exchange in Virtual Organizations and ascertain the benefits and drawback of each mechanism.

Research Question 2 *What needs to be considered when connecting a 3D Virtual World with an Electronic Marketplace realized via a Multi-Agent System?*

When connecting a 3D Virtual World with an Electronic Marketplace several design decisions need to be made and certain restrictions, imposed either by the Multi-Agent System or the 3D Virtual World, need to be considered. As part of this work these design decisions and restrictions are discussed and best practices to overcome them are presented.

Research Question 3 *To which extent can buildings in the 3D Virtual World be mapped onto Electronic Institutions?*

In the 3D Electronic Institution methodology [14] scenes of an Electronic Institution are mapped to rooms in the 3D Electronic Institution and transitions between scenes are mapped to rooms or doors in the 3D Electronic Institution. As part of this work we discuss to which extent it is possible to do a straight forward mapping of buildings, representing organizations in the 3D Virtual World, to Electronic Institutions.

Research Question 4 *Does a Multi-Agent System that is used to regulate an Electronic Market, visualized in a 3D Virtual World, affect the Electronic Marketplace's usability?*

Since all actions that are performed in the Electronic Marketplace and are visualized in the 3D Virtual World need to be approved by the Multi-Agent System, an impact on the performance and usability of the overall system might be the result. Therefore, it shall be tested whether the Multi-Agent System that is used to regulate the Electronic

Marketplace affects the usability of the 3D Virtual World and the Electronic Marketplace itself.

Research Question 5 *Are Electronic Institutions suitable for forming Virtual Organizations to regulate an Electronic Marketplace?*

Electronic Institutions have been used in various settings to regulate software agents. However, the combination of multiple Electronic Institutions to form Virtual Organizations and regulate an Electronic Marketplace is still an unexplored area of research. As part of this work we seek to test whether Electronic Institutions are suitable for forming Virtual Organizations, which are used to regulate the actions of all participants, software agents as well as human users, in an Electronic Marketplace.

As part of our research work we developed a framework to create EMs that are regulated via Virtual Organizations. We implemented a reference implementation of a 3D e-Tourism EM by using our framework to obtain answers to our research questions. Our solutions for the identified issues and findings to the research questions are presented in Chapter 2 :

A.1 We present an approach to reduce the complexity of defining internal processes of an organization implemented in a MAS by creating Electronic Institutions based on real world institutions. This approach can be use for the implementation of 3D Electronic Institutions, which are visual representations of Electronic Institutions in a 3D Virtual World, that enable users to interact with organizations in a familiar manner. We showcase our approach by presenting how to create the electronic representation of an auction house by means of an Electronic Institution. This approach has been published in [50].

A.2 Based on our Electronic Institution implementation approach, we address the problem of efficiently connecting multiple organizations to form an EM. Our solution makes use of Virtual Organizations, which are based on Electronic Institutions and can be connected via a novel communication approach called the "Ether" to establish an EM. This approach enables us to build EMs that are composed of B2C- and B2B-Virtual Organizations which are regulated by means of Electronic Institutions. In addition, we present the implementation of the middleware layer that enables the visualization of Virtual Organizations as 3D Electronic Institutions in a 3D Virtual World. For demonstration purpose we create a complex e-Tourism EM with this approach. Furthermore, we present techniques to bridge the underlying MAS with a virtual environment to provide users with the possibility to participate in the EM with interaction patterns they are used to from the real world. Finally, we discuss the issues that arise when connecting a MAS with a virtual environment and present our solutions. This work has been published in [51, 87, 86]. The usage of 3D Virtual Worlds as visualization layer for MAS is discussed in detail in Ingo Seidel's PhD thesis [84] and will be touched only briefly in this thesis.

1.2 Informed trading: Acquiring and searching relevant information

Information is one of the most important factors in making informed decisions, which is crucial in e-Business to be ahead of the competition. Since the early nineties the Internet has become a valuable information source to the general public. People gather information on the Internet before they do business. Unfortunately, they are overwhelmed by the ever expanding amount of information on the Internet and it is hard for them to differentiate between relevant and irrelevant information. Another drawback of information gathering on the Internet is that relevant information can be scattered across multiple sites. Users may gather information from various information sources on the Internet. The World Wide Web is a system of HTML-based documents which are connected via Hyperlinks. Search engines such as Google, Yahoo and Bing can be used to search for documents of the World Wide Web. The scatteredness of information on the World Wide Web results from its unstructured nature, which makes it even harder to pinpoint relevant information that assists users during their decision process.

The Semantic Web is an extension of the traditional World Wide Web. In the Semantic Web common data formats are used and documents are enriched with semantic content that is interpretable by machines. Search engines that make use of the semantic content of the Semantic Web are for example Hakia[3] or Swoogle[33]. The advantage of the Semantic Web over the World Wide Web is that it is structured and can be searched in a concise manner making it easier to pinpoint relevant information. However, the Semantic Web cannot compete with the World Wide Web in terms of pure amount of data. Hence, it would be desirable to bridge these information sources and make them searchable in a combined way. However, bridging the unstructured information of the World Wide Web with the structured information of the Semantic Web and making it efficiently searchable is not a trivial task.

[3]http://www.hakia.com/

There are currently several approaches that either leverage information of the World Wide Web or information of the Semantic Web, but there are only a limited number that actually combine the complementary information of both sources effectively. Approaches that concentrate on the information on the World Wide Web use established Information Retrieval techniques to extract and index information, which can be searched via traditional keyword-based search engines. Approaches that leverage information from the Semantic Web use semantic search techniques upon specifically structured data storages such as Triple Stores to perform searches. Triple Stores have the advantage over traditional indexes that complex queries can be answered due to their structured nature. A big concern of Triple Stores is that query response times get higher as they grow in size. Therefore, the best solution would be to combine the speediness of traditional search engines with the expressiveness and flexibility of Semantic Search Engines. However, most search engines that seek to combine unstructured and structured information do not use a combined index structure but rather two indexes and merge the results afterwards. Furthermore, they either suffer from performance issues or a lack of usability.

Another problem of Semantic Search Engines in general is that they tend to be complex and not applicable for the average Internet user. Internet users are used to clean and concise search interfaces, such as the ones of Google[4], Yahoo[5] and Bing[6], but the typical Semantic Search Engine's interface is overcrowded with multiple non-self-explanatory fields. The average Internet user does not want to spend a considerable amount of time to get used to an interface before she is able to formulate a search query. Furthermore, Semantic Search Engines, which incorporate unstructured information, e.g. from the World Wide Web, suffer from the problem that their interfaces strictly separate the search parameters for structured information and unstructured information, making it even harder for the average Internet user to get used to this type of interface.

1.2.1 Research questions and contributions

Based on the previously identified issues the following research questions have been defined :

Research Question 6 *How scalable is a system that uses a hybrid search approach with respect to the number of triples and the count of documents it can handle?*

[4]http://www.google.com
[5]http://www.yahoo.com
[6]http://www.bing.com

When using a hybrid search approach the system needs to be capable of storing and accessing structured and unstructured data efficiently to scale well when the amount of data grows. Therefore, an important aspect of a hybrid system which we covered in this work is its scalability with respect to the number of triples and documents it can handle.

Research Question 7 *How does the query-performance of a system that indexes semi-structured data relate to the one of a pure Triple Store such as SESAME [17] holding similar data?*

The query-performance of Triple Stores is very good when they operate on structured data, but poor when they operate on unstructured data. As part of this work we will compare the query-performance of a system that operates on an index that was created from semi-structured data to the one of a Triple Store holding similar structured data.

Research Question 8 *How does the interactive ontology-aware keyword-based input mechanism hold up against the traditional keyword-based input mechanism in terms of usability for the average Internet user?*

For the average Internet user it is easy to make use of traditional keyword-based input mechanisms but rather difficult to use the more advanced input mechanisms provided by Semantic Search Engines. Therefore we implemented a novel interactive ontology-aware keyword-based input mechanism that combines the input mechanisms of pure keyword-based search engines and those of Semantic Search Engines. As part of this work we will assess whether this input mechanism is usable for average Internet users who are used to keyword-based interfaces such as the ones of Google[7], Yahoo[8] or Bing[9]. To answer this research question a user evaluation was conducted.

Research Question 9 *Is the precision and recall of the hybrid semantic search approach superior to a keyword-only search?*

Since semantic search accomplishes high precision and recall results in case of a complete and comprehensive KB, it is assumed that the combination of semantic search and keyword-based search will outperform the keyword-only search approach. Therefore, another important aspect that will be covered in this work is the comparison of the hybrid semantic search approach to the keyword-only search approach in terms of precision and recall.

[7]http://www.google.com/
[8]http://www.yahoo.com
[9]http://www.bing.com

Research Question 10 *What trade-offs need to be made when a hybrid search system is implemented that should be applicable for several domains (ontologies)?*

When developing a general approach that can be applied to several domains (ontologies) trade-offs need to be made. In this work we address these trade-offs and assess how they affect the flexibility of the system.

As part of our research we created a reference implementation, namely the Hybrid Semantic Search System (HS^3), which amalgamates keyword-based and concept-based search, to address the identified issues and get answers to the research questions. Our solutions and findings to the research questions are presented in Chapter 3 :

B.1 To combine unstructured and structured information efficiently we propose a combined index structure instead of two indexes. With this approach we seek to combine the speediness of traditional search engines with the expressiveness of Semantic Search Engines to maintain flexibility and speed. A description of this approach is provided in [52].

B.2 Furthermore, we address the problem of acquiring suitable complementing unstructured information based on core knowledge that is stored as structured information in a Triple Store. We propose a system architecture that can be used to automatically enrich a given structured information storage with relevant unstructured information from the World Wide Web. The proposed system architecture is detailed in [52].

B.3 Another issue we address is the lack of search result ranking by Semantic Search Engines. Traditional search engines rank results from the most relevant to the least relevant, so users get the most relevant information presented first. In contrast, Semantic Search Engines hardly use ranking mechanisms to rank their results. Ranking becomes even more relevant when unstructured and structured information are combined. We use a ranking approach that is based on the user's query formulation and relevant data in the KB and combined index. A description of our ranking approach is provided in [49].

B.4 Finally, we address the problem of complex and overcrowded user interfaces of Semantic Search Systems that prevent the average Internet user from efficiently using them. To overcome this issue we introduce a novel input mechanism for hybrid semantic search that combines the clean and concise input mechanisms of keyword-based search engines with the expressiveness of the input mechanisms provided by Semantic Search Engines. In addition, this interface can be used to interactively formulate queries without prior knowledge of the underlying ontology. To the best of our knowledge no other Semantic Search Engine offers this possibility. A detailed description of the novel input mechanism and evaluation results is summarized in [49].

CHAPTER 2

Regulating trade

Creating complex regulated e-Markets by means of Virtual Organizations

Parts of the following chapter have been published in [50, 51, 87, 86].

2.1 Introduction

For the development of complex Multi-Agent Systems comprehensive methodologies, that support the entire development life cycle, are needed [9, 57]. The most prominent methodologies are Gaia [98] and Electronic Institutions [35]. Electronic Institutions are regulated environments, resembling real-world institutions, where agents need to adhere to predefined rules and conventions. In this work we will leverage the Electronic Institution methodology to create a regulated environment for agent mediated trading.

However, when creating sophisticated Electronic Institutions that represent the complete internal and external process portfolio of an organization, their definition and implementation tends to get complicated. This might lead to error-prone, incomplete or inefficient representations of an organization's process portfolio. To overcome this issue it would be beneficial to make use of a straight-forward design approach that facilitates the creation of Electronic Institutions by relating the processes of the corresponding real world organization to the components of the Electronic Institution. Another challenge is the interaction of humans with the Multi-Agent System. In general, software agents are configured by humans to interact on behalf of them with autonomous software agents and other user-configured software agents in the Multi-Agents System. However, a direct interaction between a human user and software agents in the Multi-Agent System does not take place. Therefore, the interaction possibilities of users are limited. Especially when participating in an Electronic Marketplace it would be preferable to have instant control over the actions of the agent. Therefore, it would be desirable to provide users with an interface that allows them to directly participate in the Multi-Agent System. This interface should be easy to use and resemble interaction patterns that people are used to from the real world.

One approach, namely the 3D Electronic Institutions methodology, has been proposed by Bogdanovych et al. [14, 13]. The 3D Electronic Institutions methodology provides a step by step guideline on how to design a 3D Electronic Institution. 3D Electronic Institutions combine the advantages of 3D Virtual Worlds and Electronic Institutions. They can be used to create regulated environments that are visualized in a 3D Virtual World. Agents and users are visualized as avatars and Electronic Institutions are visualized as buildings in the 3D Virtual World. User can interact with other users, agents and Electronic Institutions by using interaction patterns they are used to from the real world. Another distinctive advantage of 3D Virtual Worlds is that they can be used to provide users with access to disparate multi-media information sources. Consider tourism products for example, the product quality of tourism products cannot be assessed prior to purchase. However, by using a 3D Virtual World, tourist attractions, hotels or beaches can be modeled to provide prospective customers with a priori information about them.

Even though the 3D Electronic Institutions methodology provides a very good guideline on how to design a 3D Electronic Institution, it does not explicitly provide a straight forward approach on how to transform the processes of a real world organization to the specification and creation of a (3D) Electronic Institution. Furthermore, the first step in 3D Electronic Institutions methodology, the specification step, is very specific and deals with the actual low level specification of the Electronic Institution. It would be beneficial to start with a more general step that identifies and captures the internal structure and roles of the organization that should be transformed to a 3D Electronic Institution. Therefore, we created the 3D Electronic Institution Implementation Process, which makes use of the 3D Electronic Institutions methodology and provides an intuitive design approach to create a 3D Electronic Institution based on a real world organization. In case no corresponding real world organization exists, a fictional organization can be sketched and used for the 3D Electronic Institution Implementation Process.

The 3D Electronic Institution Implementation Process is an intuitive approach to design complex Electronic Institutions by using a simplified blueprint and role duty/process description of the real world organization and map them to corresponding Electronic Institution components. The blueprint and role duty/process descriptions suit several purposes. First, they act as reference to ensure that all relevant roles/processes of the real world organization are present in the Electronic Institution. Second, they provide a clear and easy to understand overview of the organization's main units. Finally, the blueprint provides guidance to model the 3D representation of the company in a virtual environment. In a nutshell, the 3D Electronic Institution Implementation Process provides a straight forward design approach that is guided by a blueprint and role/process description of the real world organization for modeling (3D) Electronic Institution.

However, a monolithic (3D) Electronic Institution can only provide limited functionality. To form complex virtual trading environments such as Electronic Marketplaces multiple interacting Electronic Institutions need to be integrated. Electronic Marketplaces (EMs) that enable buyers and sellers to conduct business are complex constructions where several characteristics need to be considered to ensure an efficient and secure exchange of goods. Unfortunately the Electronic Institution methodology does not provide an interaction-mechanism for inter-Electronic Institution communication out of the box. Another issue with (3D) Electronic Institutions is the lack of a control mechanism and functionality when agents (users) move outside of Electronic Institutions or between them. Therefore, to form complex trading environments composed of multiple interacting Electronic Institutions, which can form so-called Virtual Organizations, a supporting framework is required.

Not only the technical and functional characteristics of an Electronic Marketplace are crucial for its acceptance among participating parties, social aspects and the interaction with the market need to be considered as well. Fisher and Craig [44] discovered that the lack of social interaction in online channels and the conflict between online and traditional channels are crucial issues that hinder the adoption of EMs. Furthermore, it is important to be aware of the fit between services that are provided by the EM and those actually needed by users [74]. Especially in the B2C domain it is important to bind customers to the platform by providing additional services. The existence of trust and security mechanisms further increases the confidence of users in the EM.

To address the lack of social interaction in online channels, to ensure efficient exchange of goods and to maintain trust and security within EMs our work concentrates on the development of an extensible 3D Electronic Institution Framework supporting the creation of a regulated B2C and B2B Electronic Marketplace in a 3D Virtual World. The proposed framework makes use of the 3D Electronic Institution Implementation Process, to model internal and external processes of an organization. The 3D Electronic Institution Framework consists of a Multi-Agent System to define and regulate the marketplace, a 3D Virtual World as user interface and communication facilities to support community interaction. An Electronic Marketplace, based on our Framework, can comprise two different types of organizations i) B2C organizations and ii) B2B organizations. B2C organizations can be further classified into autonomous organizations that are independent of other organizations and those relying on the services of B2B organizations. A simplified illustration of an Electronic Marketplace with two sample organizations which can be created with our framework is depicted in Figure 2.1. The 3D Electronic Institution Framework provides a communication facility which is used by organizations to form Virtual Organizations. The term "Virtual Organization" has been defined differently in the literature. Oliveira and Rocha [73] define a Virtual Organization as a cooperation of legally independent enterprises, institutions or individuals. This cooperation provides services on the basis of a common understanding of business and appears as a single corporation to externals. A comprehensive description of Virtual Organizations and their role in e-Commerce is given in [93].

A Virtual Organization (VO), created with the proposed framework, is defined as a federation of autonomous organizations forming a single (virtual) organization that jointly conduct business and appear as single organization to all members of the Electronic Marketplace as depicted in Figure 2.1. A Virtual Organization may contain an arbitrary number of B2C and B2B organizations and every organization can be a member of several Virtual Organizations. All members of a Virtual Organization define the inputs they need and the outputs they generate, but only B2C organizations may receive input data from customers of the Electronic Marketplace. Therefore, if a Virtual Organization wants to offer services or products to customers of the Electronic Marketplace, at least one B2C organization needs to be a member of that Virtual Organization. Otherwise, if a Virtual Organization only comprises B2B organizations, it can act as a supplier to other organizations of the Electronic Marketplace. We use the metaphor of a building to visualize an organization in the 3D Virtual World. Since customers only interact with B2C organizations, only these need to be visualized in the 3D Virtual World as depicted in Figure 2.1. Consequently, if a Virtual Organization comprises B2B and B2C organizations the B2C organizations are visualized in the 3D Virtual World and represent the Virtual Organization. In other words, the visualization of the B2C organizations resemble business office(s) of the Virtual Organization to provide their services and products to customers. It is completely transparent to the customer whether he deals with a single organization or a federation of organizations that appear as a single Virtual Organization. In its simplest form a Virtual Organization can consist of just one Electronic Institution. Customers interact with software agents that implement the business processes of an organization. Software agents that interact with customers are visualized as avatars in the 3D Virtual World.

Based on the 3D Electronic Institution Framework we created Itchy Feet, a 3D e-

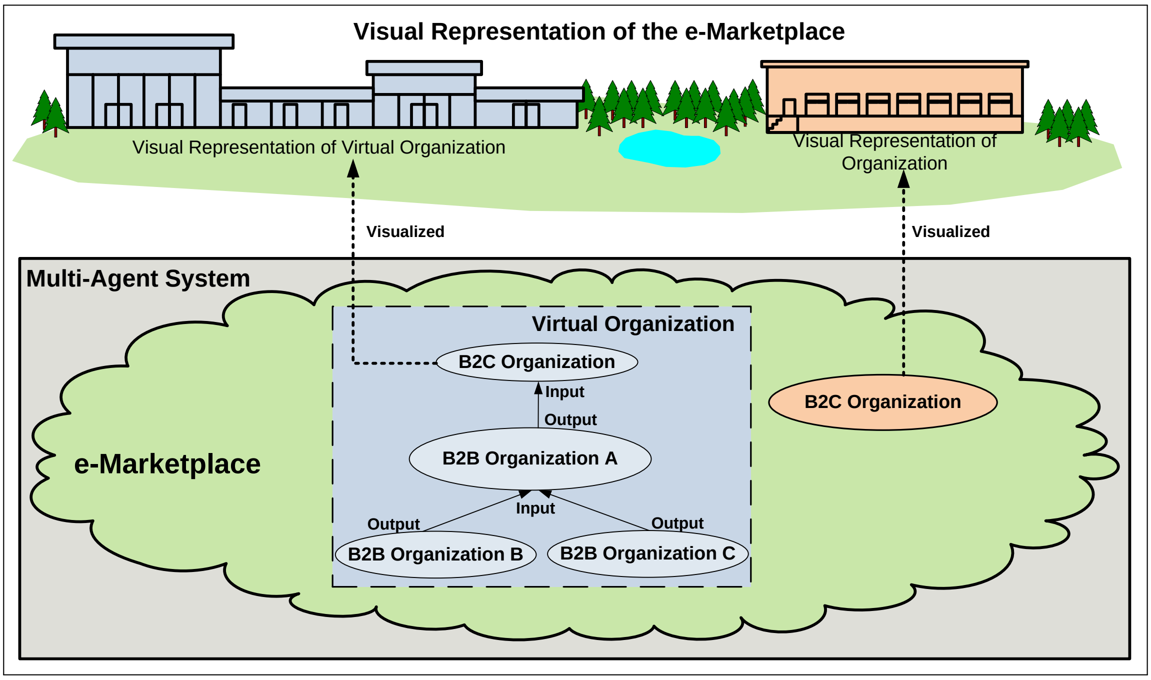

Figure 2.1: Organizations in the Electronic Marketplace

Tourism environment that supports the complex interaction patterns of providers and consumers in an e-Tourism setting. Autonomous software agents are used to render the environment information rich and Virtual Organizations and Electronic Institutions are used to regulate the actions of all participants, software agents as well as human users. Furthermore, the environment was extended with an additional 3D Virtual World application that was used for an adventure game where users learned about the Japanese Culture in a fun way [48]. One of the intentions was to leverage the unique possibilities of a 3D Virtual World and bind users to the marketplace by providing additional incentives to stay in the 3D Virtual World besides doing business.

In this chapter the 3D Electronic Institution Implementation Process, the 3D Electronic Institution Framework and the implementation of Itchy Feet are presented. The focus is laid on four major parts: i) the 3D Electronic Institution Implementation Process, ii) the framework and the creation of the Electronic Marketplace Itchy Feet, iii) the interaction between users and software agents on the Electronic Marketplace, and iv) the challenges of creating the 3D Electronic Institution Framework that connects Virtual Organizations with a 3D Virtual World and the presentation of findings to research questions defined in Chapter 1. The remainder of this chapter is structured as follows. In Section 2.2 related work is presented. In Section 2.3 we provide an overview of Electronic Institutions. The 3D Electronic Institution Implementation Process is presented in Section 2.4. The framework is presented in Section 2.5. The application of the 3D

Electronic Institution Implementation Process and the Itchy Feet environment and its key features are presented in Section 2.6. In Section 2.7 we discuss the restrictions and issues that arise when connecting Virtual Organizations with a 3D Virtual World and provide findings to the research questions.

2.2 Related Work

Research regarding modeling of agent societies has been conducted by various researchers. A formal methodology for the development of agent societies has been introduced by Dignum et al. [31]. They divide domain requirements into functional and interaction requirements. Functional requirements define what a system is supposed to do and interaction requirements define how the system is supposed to do it. The authors introduce a methodology that can be used for the design of agent societies based on the type of coordination structure [32]. Coordination in agent societies can be divided into markets, networks and hierarchies. Societies differ in the type of society, their member's "values", the society's purpose and the interaction of their participants. In a market society where members are self-interested, the purpose of the society is to support the exchange of goods. Interactions are based on standards and resemble an open society. In a network society agents have mutual interests, the society's purpose is to fasten collaboration and is based on trust. The last type of society, the hierarchical society, is closed and populated with agents that depend on each other for production purpose. According to those coordination structures institutions can be created to specify and enforce social norms and conventions. In this work we concentrate on market societies, because they support the exchange of goods and services.

A methodology for agent-oriented analysis and design was introduced by Zamboenelli et al. [98]. GAIA was the first methodology that has been specifically developed for agent-based systems. It is based on the approach of designing a Multi-Agent System as a computational organization. Furthermore, the methodology has been extended for the analysis and design of Multi-Agent Systems. Clear guidelines for analysis and design are provided by GAIA. Juan et al. [58] introduced a methodology based on the GAIA methodology to improve software engineering of agent-based open systems. They extended GAIA with four improvements: i) formal models of knowledge and environment, ii) role hierarchies, iii) explicit representation of social structures and relationships, and iv) the incorporation of dynamic changes. This methodology was used in an intelligent home network example. We decided to use the Electronic Institution methodology instead of the GAIA methodology, because the Electronic Institution methodology already offers a guidance to visualize agents and their environment, namely the 3D Electronic Institution methodology, which we used in our 3D Electronic Institution Implementation Process.

An important aspect of Multi-Agents Systems and an area of active research are electronic negotiations [11]. They promise a higher degree of efficiency and effectiveness, and, most importantly, a higher quality and faster emergence of agreements. The potential economic impact leads to an increased demand of appropriate electronic negotiations for specific negotiation situations. Current electronic negotiations, however, mainly focus on the trade of standardized, mostly inexpensive products. These are often traded in auctions, whereof English auctions are the most common. The goal is to enable the trade of complex and valuable goods which entails a negotiation process consisting of a number of communicative exchanges. Some strategies strive for that goal by relying on game theory [41] others draw ideas from information theory [27, 28]. An interesting approach to separate the agent framework and the negotiation strategy was introduced by Badica et al. [6]. They created a negotiation box, which is a component running the negotiation processes that is separated from other parts of the system. Rule based negotiation mechanisms are used to realize the negotiation box. In our implementation pre-defined interfaces can be used to implement the negotiation strategy of an agent.

Mavetera & Kadyamatimba created a conceptual framework for agent-mediated trading in Electronic Markets [69]. They identified several stages and components such as product brokering, negotiation and contract formation that are needed to implement a full Electronic Market system using agent technology. The authors disclose that stages such as product brokering, partnership formation, contract formation and purchase & delivery have not received enough attention from the agent technology research community yet. Wang et al. [95] did a comprehensive literature review on the current EM research. They identified eight major research themes and showed that most studies address only the potential but not the real impact of EMs. Furthermore, the results indicated that most of the research methodologies are “descriptive” and “qualitative” rather than “quantitative” or based on lab experiments and software prototyping. To overcome this we created an actual reference implementation of an e-Tourism EM with our framework.

Alt et al. [2] did a review of 31 consumer product Electronic Marketplaces. They discovered that most of these marketplaces only provide electronic catalogs, simple auction mechanisms and functions for the exchange of goods. Just a few of the marketplaces provided content beyond the commercial process and encouraged the development of a community. In our reference implementation of an e-Tourism EM we adapted and integrated an open source Web forum[1] that can be accesses from the World Wide Web and from within the EM's representation in the 3D Virtual World, to encourage the development of a community.

Karacapilidis & Moraïtis developed an agent-based artificial market system and introduced *artificial employees* [60]. Such *artificial employees* perform tasks that have been delegated to them by users. These agents are not instantiated for a specific task, but reside in the artificial market system and execute a series of tasks which have been delegated to them. This is similar to our approach where every user is the principal of a software agent that can either be controlled by the user or perform delegated tasks autonomously.

Fasli and Michalakopoulos developed the e-Game platform which can be used as a test-bed for trading agents [40]. Their main intent was to provide a platform where developers are able to run experiments with different bidding strategies. The platform offers a Java Application Programming Interface (API) for developers to create their own agent implementations. Agents that participate in our e-Tourism EM offer an Java API which can be used to extend their application logic.

Zeng and Zhang proposed an integrated approach for developing e-Commerce systems by using intelligent agents, multi-attribute decisions, web technology and a common ontology [99]. Intelligent agents either act as buyers or sellers and carry out tasks such as finding appropriate business partners and performing automated negotiation according to different negotiation strategies. Buyer and seller agents are itself composed of multiple agents such as interface agents, buyer manager agents, evaluation agents and mobile agents. They all work together to achieve the overall goal. To facilitate the data exchange between seller and buyer agent implementations of different partners, a common ontology is defined. The agents use a multi-attribute decision method to decide which product fits the needs of the buyer best. The Java Agent Development Environment (JADE) is used to realize buyer and seller agents which are embedded in the Struts Framework to interact with users via a web-based interface. In contrast, we use a 3D Virtual World where agents are represented as avatars to interact with users.

Pao-Hua et al. proposed an approach to use an intelligent agent for an e-Commerce bargaining system [75]. They use a Multi-Agent Framework to create bidding agents that assist the user in selling or buying items and browsing the transaction information on the Electronic Marketplace platform. Every agent uses components such as a mailbox, a message handler, an execution monitor, a co-ordination engine, a planner and scheduler and an ontology database. The agent uses the mailbox and the message handler to communicate with other agents in the environment. The execution monitor, coordination engine

[1] http://jforum.net/

as well as the planner and scheduler are used to coordinate the internal tasks according to the priority determined by the agent's current goal. The ontology database stores the logical definitions of each fact, i.e. the vocabulary used and its legal attributes, the range of the legal values and any constraints and relations between them. Agents may collaborate either directly or by using a broker agent that acts as middleman between seller and buyer agents. Since our EM uses Electronic Institutions, the message exchange is handled by Ameli [39], the Electronic Institution execution environment, which eliminates the need for a custom implementation.

Chiu and Leung designed a virtual enterprise of independent tourism service providers as a Multi-Agent System [23]. The authors were motivated by the lack of tourist portals that pro-actively assist tourists or adequately retrieve disparate touristic information and services. The developed environment addresses these issues and is a ubiquitous tourist assistance system. Agents make use of Semantic Web concepts to improve the planning stage and help tourists in understanding and specifying their requirements and preferences. We provided agents of the e-Tourism EM with a Web Service interface to access the Hybrid Semantic Search System (HS^3). HS^3 was developed as part of this work and can be used to provide users of the e-Tourism EM with relevant information to their trip.

Argente et al. did a survey of organization-oriented Multi-Agent System methodologies [4]. The authors state that it would be interesting to have human organizational designs, such as virtual organizations, matrix organizations, teams, hierarchy and bureaucracy taken into account in organization-oriented Multi-Agent System methodologies. Furthermore, the authors conclude that the similarities between the topological structures of human organizations and agent systems should be used for the analysis and design of Multi-Agent Systems. Our 3D Electronic Institution Framework is based on Virtual Organizations, which may consist of multiple Electronic Institutions.

Cardoso and Oliveira developed a normative framework based on Electronic Institutions for contract validation and enforcement in Virtual Enterprises [19]. These contracts are the result of the agreement of agents to jointly perform a specific business activity. The authors leveraged Electronic Institutions to regulate interactions between parties engaged in business transactions and to maintain trust among participating parties. In our framework Electronic Institutions are used to form Virtual Organizations that regulate the EM.

3D visualization as instrument to comprehend complex information and its relations more easily as well as the combination of 3D Virtual Worlds and complex systems, such as Multi-Agent Systems, to facilitate more intuitive interaction has become an active research area in recent years. Several successful approaches to visualize systems or processes have been created in the past years. Chao introduced a user interface for the Unix top command by using the game Doom[2] to visualize and literally “kill” Unix processes [22]. Every process is represented as monster in a room. The more important a process is, the more powerful the monster is and, therefore, harder to kill. The author of the interface incorporated the possibility to use different weapons where novice users get less powerful weapons than professionals, in order to avoid an accidental kill of an important process by an inexperienced user. Another successful approach has been proposed by Manojlovich et al. [67]. The framework makes use of Game Bots that are realized via the Unreal Game Engine[3]. In this framework an agent system is used as the mediator between a military simulation and a 3D Virtual World. The 3D Virtual World and the military system are monitored by agents and the current status is used to update the simulation’s visual representation. Kot et al. created a source code comprehension tool by utilizing a 3D computer game engine [66]. The authors stated that a computer game engine can be used for source code comprehension as long as the visualization can be represented in terms of a limited number of discrete, interactive and physical entities placed in a static 3D world of limited size. We use the 3D computer game engine Torque[4] to visualize the EM.

A 3D environment which incorporates agent logics to make the environment more dynamic was introduced by Smith et al. [88]. Most environments are rather static and follow pre-defined routines to generate a dynamic effect. The proposed framework consists of a society of agents in which each agent controls a 3D object. In our e-Tourism EM implementation, avatars in the 3D Virtual World are either controlled by autonomous agents or users. Users and autonomous agents move in the 3D Virtual World via their avatars, therefore the 3D environment is dynamic.

The interaction between agents and humans was studied by Traum and Rickel [92] by analyzing dialog models. Their main interest was directed towards the interplay between speech and nonverbal signals, proximity and attentional focus of others and the ability to maintain multi-part conversations.

The different models of user interaction with agents have been analyzed by Payne et al. [76]. A case study was conducted to ascertain whether different types of user interaction modalities directed toward agents with similar functional capabilities have an effect on the performance of the agent community. The result was that the performance of the agent community was related to the type of user interaction that was used.

So-called liaison agents that facilitate the interaction between humans and autonomous

[2]http://www.idsoftware.com/
[3]http://www.unrealengine.com/
[4]http://www.garagegames.com/

agents have been introduced by Martin et al. [68]. By using these agents users are able to act as an integrated part of a Multi-Agent System. In our framework so-called controlled agents that are visualized as avatars in the 3D Virtual World facilitate the interaction between humans and autonomous agents.

Bromuri et al. developed a virtual e-Retailing environment in GOLEM (Generalized Onto-Logical Environments of MAS) [18]. e-Retailing is a synonym for B2C where a company conducts business with a private customer. GOLEM is an agent environment that shares similarities with Electronic Institutions. Instead of Electronic Institutions to regulate agents behavior, GOLEM uses so-called containers which provide support for agents, objects, processes and environmental services inside a declarative context that constraints the interaction of all contained entities according to a set of declarative rules. Multiple containers may be connected to a single construct similar to the Federation construct of the Electronic Institution Methodology. The virtual e-Retailing environment uses a semantic registry which can be used by agents to query for entities of interest. The user is represented as an avatar within GOLEM and can interact with other entities via triggers and emitters. Products and services of the e-Retailing environment are browsed via the semantic registry. All agents store product preference profiles reflecting user searches and purchases that help to provide suggestions of interest to users.

Aranda et al. created a MMOG (Massively Multiplayer Online Game) architecture based on a Multi-Agent System [3]. The architecture comprises three layers: the Human Computer Interaction layer, the Intelligent Virtual Environment layer and the MMOG layer. The game mechanics are modeled in the MMOG layer and Electronic Institutions are used to run the game. In contrast to our framework, their architecture does not have a dedicated communication layer which causally connects the 3D Virtual World and the Multi-Agent System. The communication facility is included in one of the other layers. Furthermore, the authors introduce a common ontology based on the Web Ontology Language that is used to exchange semantic content among agents and which should aid in the standardization of these kind of systems. Debenham and Simoff have introduced the term "Virtual Institution" [26]. They define a Virtual Institution as a virtual place where agents and humans can trade securely by interacting naturally via a 3D Virtual World. According to the authors Electronic Institutions and 3D Virtual Worlds complement each other by facilitating a high level of interaction but still retaining a certain level of regulation. A prototype of such a Virtual Institution has been realized by Bogdanovych et al. [15]. In contrast to our framework, that supports connecting multiple Electronic Institutions to form complex Virtual Organization which can be represented as single building in the 3D Virtual World, Virtual Institutions associate every building in the 3D Virtual

World with exactly one Electronic Institution.

2.3 Electronic Institutions

An Electronic Institution is the computational realization of a real-world institution and enables agents to interact with each other according to predefined conventions and rules [37]. Electronic Institutions enable the specification of regulated and well structured environments. In particular, Electronic Institutions are taking control over security aspects to ensure participants adhere to the institutional rules and fulfill their obligations. Literally speaking, an Electronic Institution is the guard to ensure that all actions performed by agents are in line with the rules and regulations that apply in the institution. One of the first implementations of an Electronic Institution was an auction house that was developed as part of the Fish Trading (MASFIT) project [24]. The Electronic Institution methodology used in this project emerged from the FishMarket project. The FishMarket is an electronic realization of a real fish market created as test-bed for trading agents in electronic auction markets [81].

The basic components of Electronic Institutions are i) the Dialogical Framework, ii) the Performative Structure, and iii) the Norms and behavioral rules. The Dialogical Framework defines the ontology that holds messages that can be uttered by agents and illocutionary particles that are used by the agents. It defines the organizational structure of the society of agents, comprising the agent roles as well as the relationships between them. The Performative Structure determines the type of dialogues agents can engage in. These dialogues are named scenes. They are connected with each other to create sequences of activities or dependencies among them. Norms and behavioral rules regulate the interaction and communication in a group of agents. They ensure the safety and stability of the system by prohibiting agents to behave in non-expected or non-accepted manners [1]. Norms are used to define the legality and illegality of actions, in a vague and abstract way to ensure they apply in different circumstances [30]. The movement between scenes is realized via so-called transitions, which are responsible for routing agents. Agents leave scenes where they have been playing a given role and enter other scenes to play the same or a different role. A scene instance is generated by the owner of a scene. After a scene has been instantiated, agents may join this scene. For every scene a subset of the overall ontology as well as the scene's structure, by means of a Finite State Machine, are defined. Furthermore, certain states of the Finite State Machine have a list of agent roles, defining whether agents playing in these roles are eligible to join or leave. The arcs of the Finite State Machine are labeled with messages (illocutions) originating from the ontology. If a corresponding message is uttered, a state transition is performed by the Finite State Machine. The Norms and behavioral rules are expressed as pre- and post-conditions of the illocutions that are admissible in the Performative Structure of the Electronic Institution.

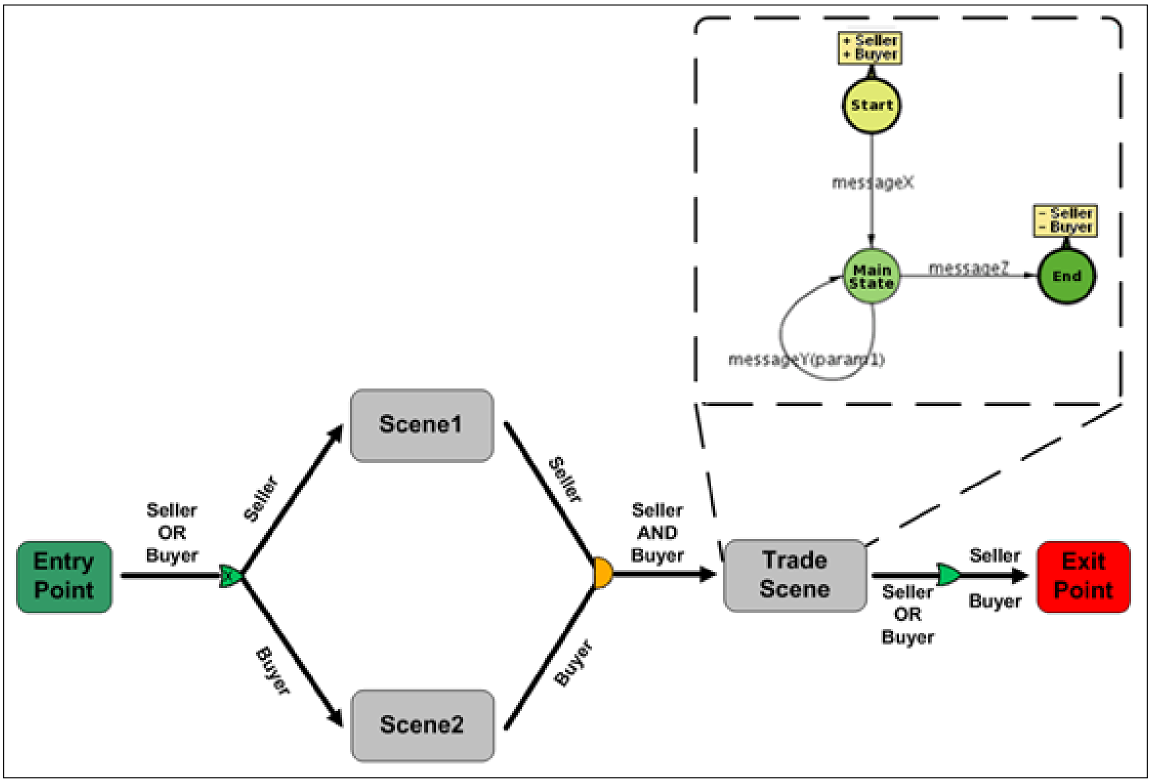

Figure 2.2: Performative Structure and Scene Protocol of an Electronic Institution

Figure 2.2 depicts an example of a Performative Structure and scene in an auction setting. Every Performative Structure has an `Entry` and an `Exit Point`. Agents can move from one scene to another scene by using `Xor`, `Or` and `And` transitions. Consider two agents, one playing in the `Seller` role and the other one in the `Buyer` role. Both agents start from the `Entry Point`. The first transition, depicted as an arrow with an `X` inside, is a `Xor` transition. It restrains an agent from moving to more than one subsequent scene. In our example, the `Seller` has to move to `Scene1` and the `Buyer` to `Scene2`. The second transition, depicted as a semicircle, is an `And` transition. An `And` transition defines a synchronization point which has to be passed by multiple agents simultaneously. An agent has to wait in such a transition until all required other agents have moved to that transition. Then they are able to move to the next scene simultaneously. In our case the `Seller` and `Buyer` agents have to synchronize after leaving `Scene1` and `Scene2`. The `And` transition can only be passed if an agent playing in the

`Seller` and an agent playing in the `Buyer` role are present in the transition. If only the `Seller` is present in the `And` transition it has to wait until the `Buyer` joins, before it can proceed. The third transition, depicted as an arrow without `X`, is an `Or` transition. An `Or` transition is less restrictive than a `Xor` transition and allows agents to join more than one subsequent scene. In our example only the `Exit Point` is reachable from the `Or` transition, therefore the `Buyer` and `Seller` can only move to the `Exit Point`.

Figure 2.2 also shows the scene protocol of the `Trade Scene`. Scene protocols are represented as Finite State Machines where state changes are triggered by uttering messages. All agents in a certain scene share the same scene protocol and state. A state defines the possible messages that can be uttered and which roles are eligible to join or leave. In the `Trade Scene`, the `Seller` and `Buyer` agents are eligible to join the scene in the `Start` state and to leave it in the `End` state. In the `Start` state, `messageX` can be uttered, while in the main state `messageY` or `messageZ` can be said. For a detailed description of Electronic Institutions we refer to [35].

2.4 The 3D Electronic Institution Implementation Process

The 3D Electronic Institution Implementation Process comprises three main steps: the *Conceptual Design* step, the *Specification* step and the *Modeling & Mapping* step. It has been published in [50].

1. ***Conceptual Design***
   In the Conceptual Design step, initial information such as role duty descriptions and a blueprint of the organization, which should be modeled as 3D Electronic Institution, are identified and created. Role duty descriptions are created by identifying processes and tasks of people that work in the organization. These descriptions are used in the subsequent *Specification* step to specify agents that implement these tasks and processes. Based on the role duty descriptions a supporting environment is drafted, namely the blueprint of the organization. This blueprint is not the actual blueprint of the real organization, but rather a simplified version that depicts the supporting environment for the identified roles. It suits the purpose of getting an overview of the organization and is used in the *Modeling & Mapping* step to build the 3D representation of the Electronic Institution that is specified in step *Specification*.

2. ***Specification***
   In the Specification step, the role duty descriptions and the blueprint are used to specify the Electronic Institution. Role duty descriptions are used to define agents, the Dialogical Framework and the scenes of the Electronic Institution and the blueprint is used to define the Performantive Structure and the corresponding transitions between scenes. For the Specification step, the Specification and Verification step of the 3D Electronic Institution methodology are used [14]. The 3D Electronic Institution methodology provides ISLANDER [36], a tool that can be used to specify and verify an Electronic Institution's specification.

3. ***Modeling & Mapping***
   In the Modeling & Mapping step, the 3D representation of the Electronic Institution is created based on the blueprint that was defined in the *Conceptual Design* step. Subsequently, the Electronic Institution's components are mapped to their 3D representations. In a straight forward manner scenes are mapped to rooms and transitions are mapped to doors in the 3D representation.

Even tough the 3D Electronic Institution methodology provides steps that are similar to the Modeling & Mapping step, namely the Automatic Generation, Annotation and Integration step, the 3D Electronic Institution Implementation Process does not make use of them. The reason is that the created blueprint of the building provides more detail on the actual layout of rooms and doors than the Performative Structure does, which is used by the Automatic Generation step to create the 3D representation of the Electronic Institution. Even though the Automatic Generation step of the 3D Electronic Institution methodology creates a building automatically, manual intervention in form of the Annotation step is needed to adapt the building to the needs of the organization. In Section 2.6.1 we will showcase the usage of the 3D Electronic Institution Implementation Process to create the 3D Electronic Institution of an auction house that is part of the implemented e-Tourism EM Itchy Feet.

2.5 Framework Description

The 3D Electronic Institution Framework was developed to support the creation of easily extendable Electronic Marketplaces that can be extended with an arbitrary number of organizations. This framework offers the possibility to visualize any organization in the Electronic Marketplace. To realize the Electronic Marketplace and its organizations we use Electronic Institutions, a Multi-Agent System Methodology, and the Torque Game

Engine[5] to visualize the organizations in a 3D Virtual World. Similar to organizations, Electronic Institutions require their members to obey the (business) rules and use the (business) processes defined by the institution. Therefore, we decided to use Electronic Institutions to implement the organizations of the Electronic Marketplace. We apply Electronic Institutions to regulate the interactions between software agents, implementing the business processes of an organization, and the interaction between users of the Electronic Marketplace and software agents. The 3D Electronic Institution Framework enables the connection of Electronic Institutions to a 3D Virtual World in order to allow human users to participate in the agent system and to facilitate the visualization of software agents in the 3D Virtual World [85].

2.5.1 Enabling the formation of VOs

The Electronic Institution methodology defines a hierarchical structure for the organization of Electronic Institutions. The top-level organizational construct is called Platform. Within the Platform construct multiple so-called Federations can be defined that aggregate multiple Electronic Institutions to a conjoint construct with a local name space. Two Electronic Institutions may share the same name if they belong to different Federations. Before an agent may enter an Electronic Institution that is part of a Federation, it needs to enter the Federation. An agent is able to access any Electronic Institution within a Federation. We leveraged the Federation construct of the Electronic Institution methodology to support the formation of Virtual Organizations, which comprise multiple Electronic Institutions, in the Electronic Marketplace. Unfortunately, the Federation and the Platform construct do not offer a mechanism for inter-Electronic Institution communication. Inter-Electronic Institution communication in our work refers to the ability of software agents that reside in different Electronic Institutions to exchange data with each other.

In the original Electronic Institution methodology it is not possible for two agents in different institutions to communicate with each other out-of-the-box. However, the support of inter-Electronic Institution communication is crucial for the efficient realization of Virtual Organizations. In a Virtual Organization data needs to be exchanged continuously between their member organizations. Furthermore, it should be possible to define global functions that can be used by all members of the marketplace and to regulate inter-Electronic Institution communication. Therefore, we enhanced the Federation construct by introducing the so-called `Ether`. The `Ether` is a dedicated Electronic Institution offering inter-Electronic Institution communication functions as well as global functions such as a shopping cart, an inventory and a chat that are available to all user of the Elec-

[5] http://www.garagegames.com/

tronic Marketplace. Since the `Ether` itself is an Electronic Institution, it is possible to use all features specific to Electronic Institutions for regulating inter-Electronic Institution communication. The `Ether` is not mapped to a building in the 3D Virtual World and its functions are accessible at every location in the 3D Virtual World.

2.5.2 Connecting VOs with the 3D Virtual World

The framework supports agents that are either autonomous or controlled by a human user. Human users delegate actions to a so-called controlled agent which is represented by an avatar in the 3D Virtual World. Actions that are delegated to the controlled agent are executed in the Virtual Organization where the agent currently resides. More precisely the actions are executed in the B2C organization of the Virtual Organization, because user controlled agents may only interact with customer centric processes. The B2C organization is realized as Electronic Institution and connected via the `Ether` with all the member organizations of the corresponding Virtual Organization. Therefore, by observing the direct feedback of the controlled agent in the 3D Virtual World the user becomes familiar with the restrictions that prevail in the Virtual Organization. Any user action is verified by the agent in the corresponding Electronic Institution of the Virtual Organization to ensure that the user acts according to the rules of the institution. However, in contrast to controlled agents, which are always visualized via avatars in the 3D Virtual World, the visualization of autonomous agents depends on their main purpose. Autonomous agents which offer services directly to users are visualized, but those which only interact with other autonomous agents and have no direct contact to users are not visualized. In a nutshell, only agents that implement customer centric business processes need to be visualized in the 3D Virtual World.

Since users only interact with B2C organizations directly, only these are causally connected to the 3D Virtual World. Therefore, if a Virtual Organization needs to be visualized in the 3D Virtual World, at least one of its member organizations needs to be a B2C organization. The visual representation of a B2C organization then represents the entire Virtual Organization in the 3D Virtual World. The architecture used by the 3D Electronic Institution Framework is depicted in Figure 2.3.

The Electronic Institution Layer comprises Ameli [39] and the Remote Server [84]. Ameli is the Electronic Institution runtime environment. All events that occur within an Electronic Institution are propagated to the middleware. The Remote Server provides a communication facility to external agents. External agents use this facility to exchange messages with the Ameli system. Messages can trigger actions that change the internal state of an Electronic Institution. The corresponding Electronic Institution specification

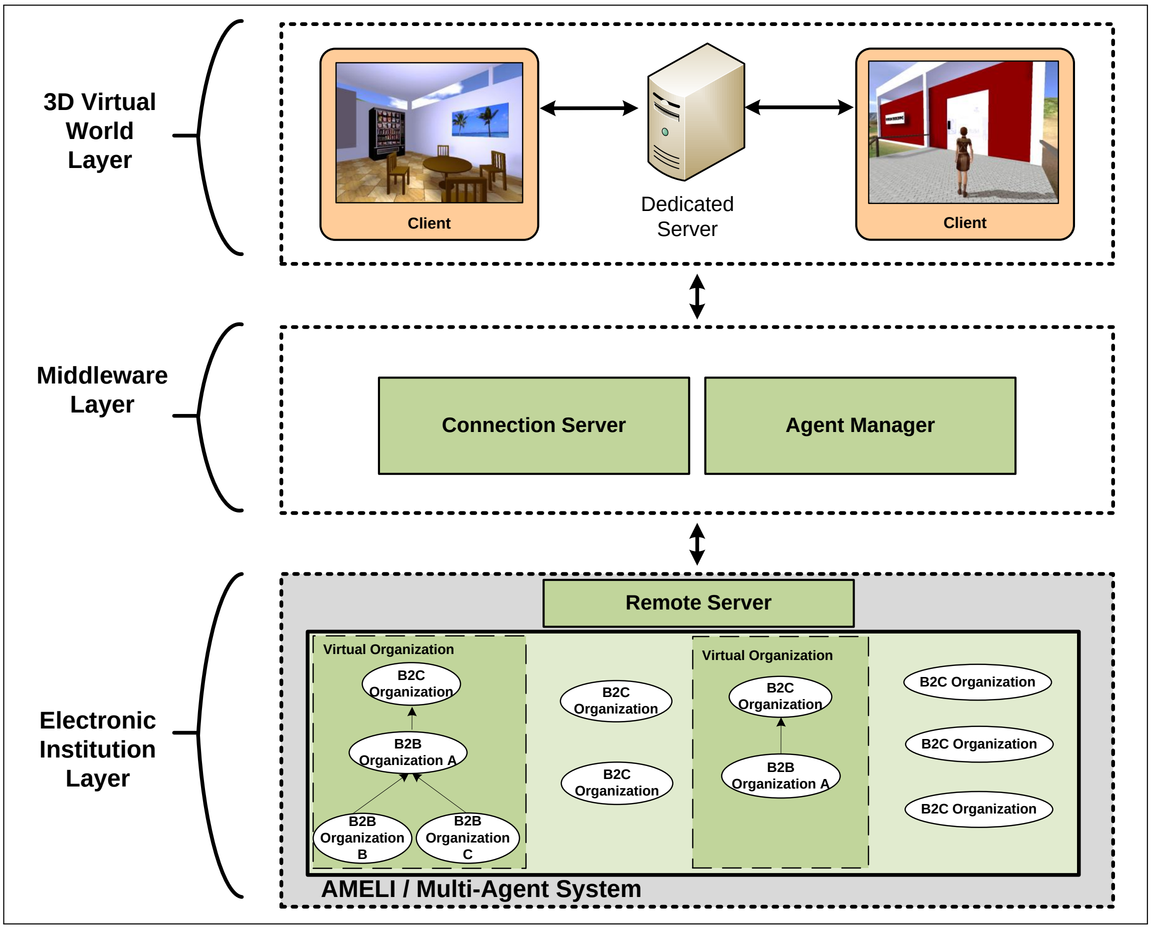

Figure 2.3: The 3D Electronic Institution Framework Architecture

is used by Ameli to validate messages. The external agents get informed of state changes via reply messages.

The Middleware Layer comprises the Connection Server and the Agent Manager. The Connection Server connects the server that is hosting the 3D Virtual World with the Ameli system. The Agent Manager holds instances of external agents. These agents use the communication facility of the Remote Server in the Electronic Institution Layer. Every agent in the Agent Manager is controlled by a user via the 3D Virtual World in the 3D Virtual World Layer.

The 3D Virtual World Layer comprises the dedicated server that is hosting the 3D Virtual World and the client. The dedicated server manages the state of the 3D Virtual World and reacts to actions performed by users or state changes in the Multi-Agent System and the client visualizes the 3D Virtual World on the user's computer. For a detailed description of the Middleware Layer and 3D Virtual World Layer we refer to [85].

2.6 Going 3-dimensional in e-Business

Electronic Institutions are useful for the creation of an agent-based Electronic Marketplace and the formation of Virtual Organizations. They enable the participation of heterogeneous autonomous software agents and define a regulatory environment that governs their actions. However, the ability for human users to take part in Electronic Institutions is rather limited. The Electronic Institution Development Environment (EIDE) [38], which offers Java APIs and tools to create Electronic Institutions, contains a graphical tool which allows users to slip into the role of a software agent and participate in Electronic Institutions. This tool requires a thorough understanding of Electronic Institution concepts and is only valuable for developers, but not usable for end users. In order to overcome this limitation the 3D Virtual World is used as an alternative user interface for end users in our work. Furthermore, the 3D Virtual World is used to provide the functions of the Electronic Marketplace to the user. The fact that the user interacts with a Multi-Agent System is hidden by the framework and the user is only presented with those interface controls that are necessary to complete her goals. In the following subsections we will present how to use the 3D Electronic Institution Implementation Process and the Itchy Feet Electronic Marketplace which was created using the 3D Electronic Institution Framework described in Section 2.5.

2.6.1 Creating a 3D Electronic Institution

In the this section we demonstrate the application of the 3D Electronic Institution Implementation Process by creating a 3D Electronic Institution of an auction house for the Electronic Marketplace Itchy Feet. Every subsection showcases one step of the 3D Electronic Institution Implementation Process to create the auction house's representation as 3D Electronic Institution. Finally, we showcase the 3D Electronic Institution's functionality and the interplay of the Multi-Agent System Layer with the other architectural layers by means of a scenario.

Conceptual Design

In this section we demonstrate how to derive initial information such as role duty descriptions and a blueprint of the organization which are needed for the 3D Electronic Institution Implementation Process, by using a fictional auction house as example. The first step towards realizing an organization as 3D Electronic Institution is to identify the integral roles of the organization and their involved duties (processes). We will start from scratch by identifying roles and corresponding duties needed in an auction house.

1. To keep customers informed about available products, staff is needed that prepares auction catalogues. Therefore the duty of the information staff is to provide information to customers by means of advice and auction catalogues.

2. Furthermore, someone is needed who keeps track of all auctions, decides when and which kinds of goods are auctioned and oversees demand and supply. This task is usually conducted by an offer manager. The offer manager decides which products are auctioned at which times and provides the information to the information staff. She advices the transport staff to get more items of a specific product, if she notices a lack of supply. Furthermore she decides whether a product group is eligible to be auctioned in the auction house.

3. The transport staff is responsible for bringing products into the institution and delivering purchased products to customers. The transport staff keeps track of all products that have been brought into the auction house and informs the auction manager about newly arrived products.

4. In any auction house several auctioneers are required to lead auctions. Auctioneers are instructed by the auction manager, who provides them with information about the kinds of products to be sold. When an auction has ended, auctioneers inform the auction manager of the sold products.

5. Furthermore, someone is needed who brings products from the good storage to the auction room, so customers can have a look at what is going to be auctioned next. This person also informs the offer manager in case a product was sold, so it wont be put on auction again.

6. Someone who accepts payment for purchased products is needed as well. This task is conducted by a cashier in most auction houses. As soon as customers have paid for the products, they are either handed out to the customer or are delivered to the customer's home by the transport staff.

7. A customer is the user of the auction house. She is eligible to participate within an auction with auctioneers and other customers.

It is evident that a role for registering and validating customers is missing. However, we have not listed this role because the presented auction house is part of an Electronic Market which already provides a registration and validation functionality out-of-the-box by using the proposed 3D Electronic Institution Framework presented in Section 2.6.2.

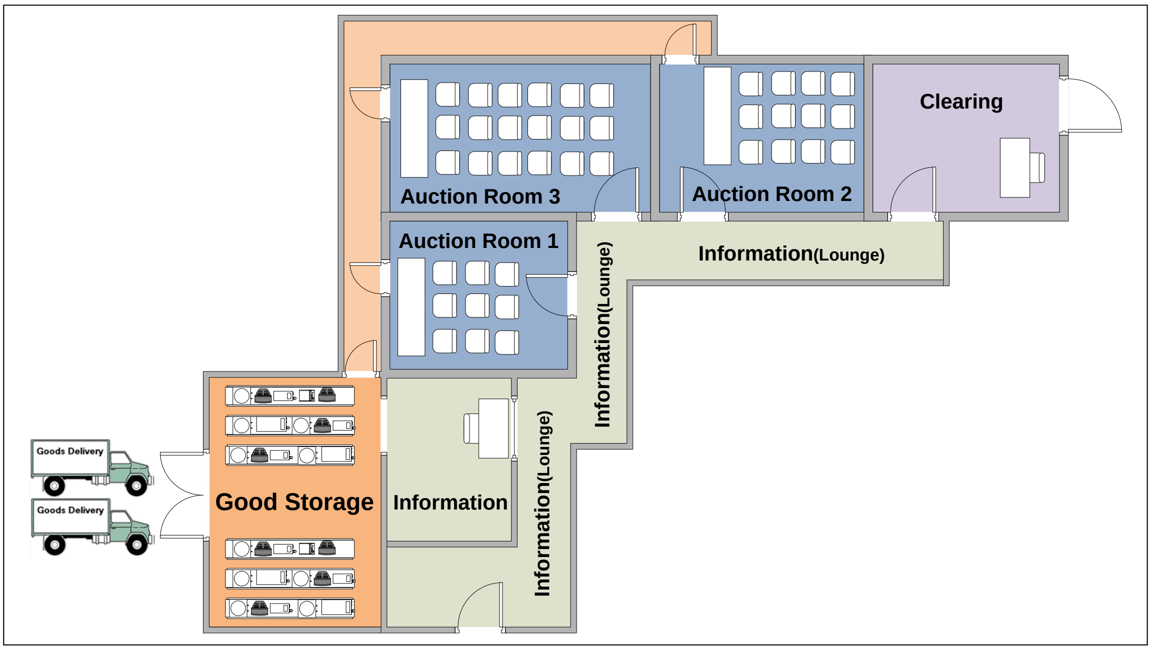

Figure 2.4: Auction house blueprint

Having identified the roles and their duties it is straight forward to build a supporting environment around them which can be visualized as blueprint of the organization. Figure 2.4 depicts a blueprint of a supporting environment for the previously defined roles. There is an information(lounge) room where customers can inform themselves about upcoming auctions and reside while waiting for an auction to start. For storing and managing goods the good storage room is used. It is equipped with a loading platform where goods are delivered by trucks from different companies. The good storage room accommodates the transport staff and the offer manager. The transport staff unloads the goods and puts them into shelves. The offer manager keeps track of all goods and shares product information as well as the intended date of auctioning with the information staff. Right next to the good storage room is the information room located. It is connected to the good storage room in order to provide customers with the latest product and auction information. At the information desk auction catalogues keep customers informed. There are three different auction rooms where products are auctioned. Finally there exists a clearing room with a cashier that is used by the customers to pay and receive their purchased goods.

Specification

The blueprint of the auction house will now be transformed into an Electronic Institution. First, we will take a look at the institution's rooms and identify which kind of conversations are performed within them. This information will be used to create an ontology for the Electronic Institution. Since the described scenario is fairly complex we will concentrate on the most interesting rooms and summarize the others.

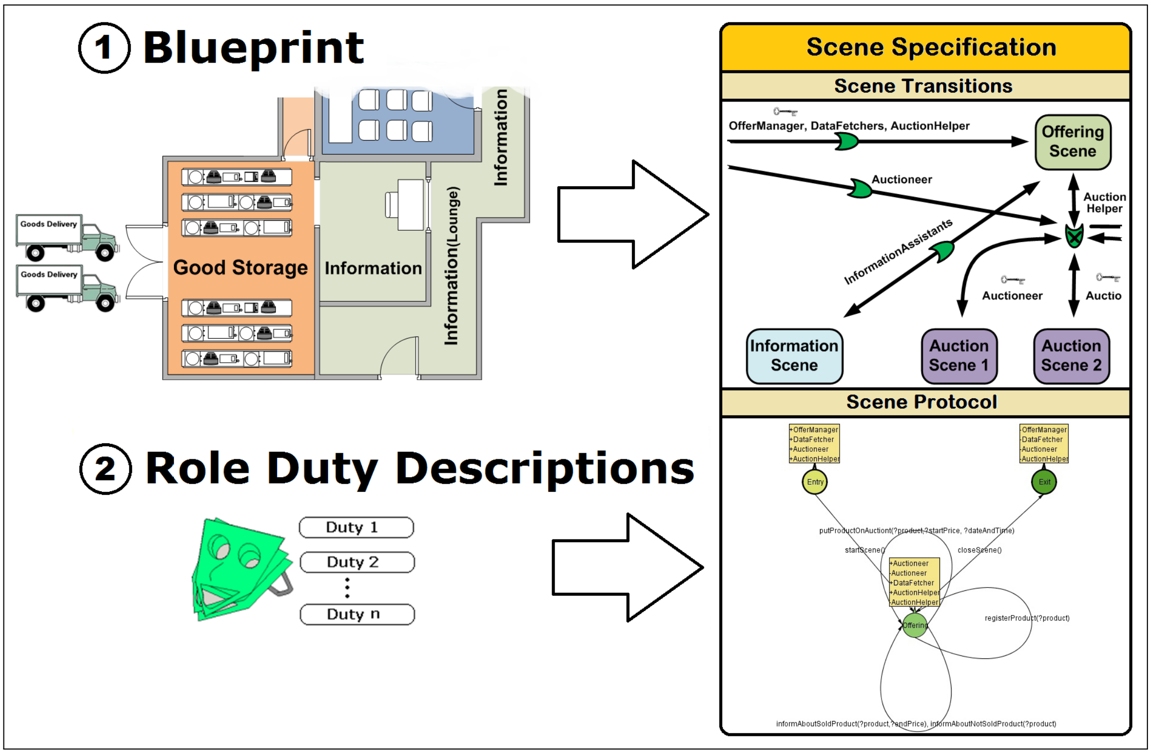

Figure 2.5: Derive Scene from auction house blueprint and duty description

Based on the role duty descriptions and the auction house blueprint we derive the Electronic Institution. The auction house's blueprint tells us how the rooms have to be mapped to scenes and how these scenes need to be connected with each other. The role duty descriptions are used to create the ontology, scene protocols and the Dialogical Framework. The result is the Performative Structure of the Electronic Institution. Figure 2.5 illustrates the conversion process. First we use the auction house's blueprint to determine which entries and exits in the scene are needed. Entries to the scene correspond to doors or connections between rooms in the blueprint. The blueprint of the auction house also tells use where to place the scene in the Performative Structure. By combining the role description and the auction house's blueprint we can tell which entry will be used by which role. Due to this fact we can define the roles for each transition and the roles that are valid in the `Entry` and `Exit` state of a scene. For example, in Figure 2.5, there is a door between the information and the good storage room which is modeled as a transition between the `Information` and `Offering Scene` for the role *Information Assistant* in the Performative Structure. The role *Information Assistant* has been introduced to cover the duties of the information staff in the auction house. As there are multiple *Information Assistant* roles which have slightly different duties, the transition was labeled with "Information Assistants" instead of "Information Assistant" in the Performative Structure. We will list the possible sub-roles of *Information Assistant* in Section 2.6.2. A simplified version of the Performative Structure is depicted in Figure 2.6. A key symbol near a role name means that the scene is opened by an agent playing in this role. According to the role duty descriptions the scene protocols can be derived. A role's duty description is used to extract messages and corresponding parameters for the scene protocol. The process of deriving these messages on behalf of a role duty description and the auction house's blueprint is demonstrated for the auction rooms. The scene specifications for the remaining scenes are listed in Appendix A.

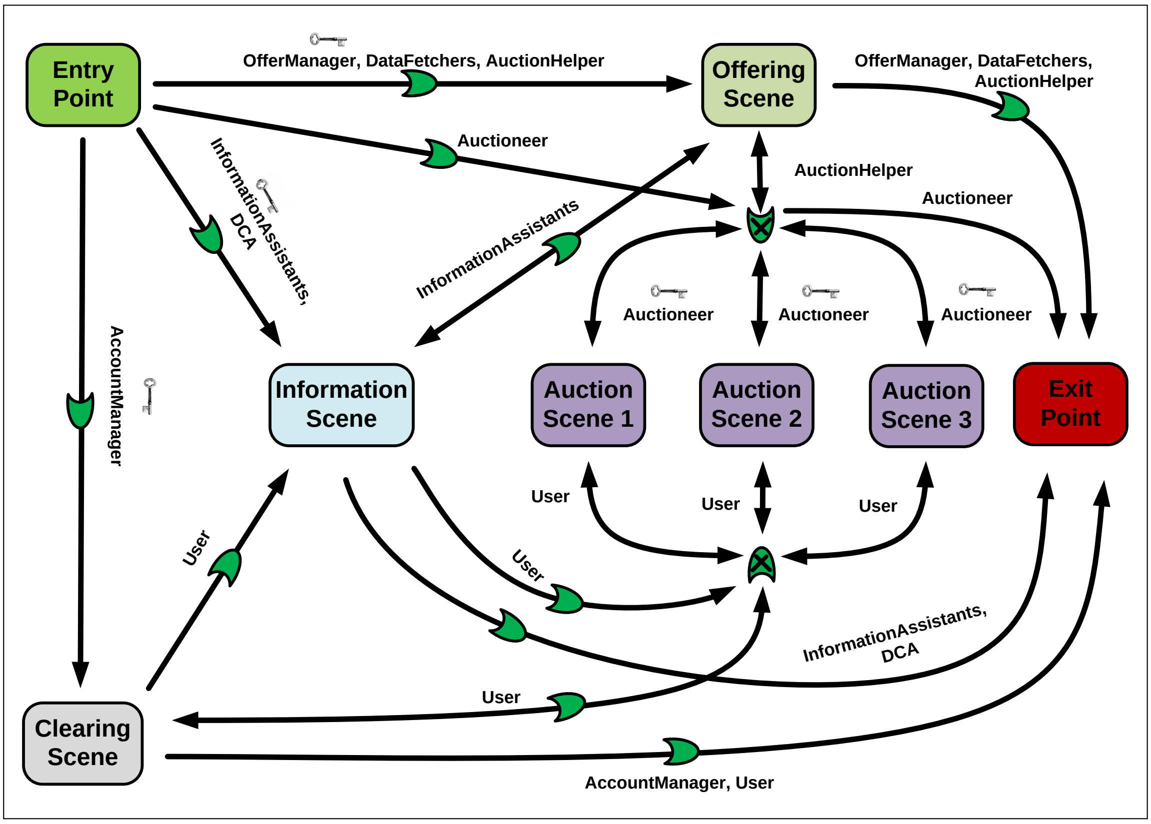

Figure 2.6: The Performative Structure of the Auction House EI

In our auction house blueprint three different rooms are used for auctions. However, all auctions follow the same protocol, namely the English auction protocol. The protocol includes the announcement of participation, the announcement that an auction has been started, the bidding for a product, the handing out of a ticket (representing a winning bid) and the closing of the auction. The required ontological terms for an auction scene that implements this protocol are listed in Table 2.1.

Sender	Illoc.	Ontology Term	Params	Receiver
Auctioneer	inform	`startScene`	{}	*all*
AuctionHelper	request	`registerProductForAuction`	product,startPrice,dateAndTime	*Auctioneer*
Auctioneer	inform	`putProductOnAuction`	product,startPrice,dateAndTime	*all*
Auctioneer	inform	`startAuction`	product,startPrice	*all*
User	inform	`bid`	amount	*Auctioneer*
Auctioneer	inform	`currentOffer`	amount,agentName,counter	*all*
Auctioneer	inform	`productSold`	price	*all*
Auctioneer	inform	`productNotSold`	{}	*all*
Auctioneer	inform	`handOutProduct`	product	*User*
Auctioneer	inform	`closeScene`	{}	*all*

Table 2.1: Ontological terms and messages used in the Auction Scene

Before any auction can be started the auction scene itself needs to be started by using the term `startScene`. Next, we need a term that can be used by the *Auction Helper* to register a new product with the *Auctioneer*. This is done via the `registerProduct-ForAuction` term that holds the parameters product, startPrice and dateAndTime. Subsequently, a term is needed that informs *Users* about the product that is going to be auctioned. This is accomplished by the `putProductOnAuction` term which holds the parameters product, startPrice and dateAndTime. As soon as the actual auction is started, a term is needed to inform *Users* about this event. For this purpose the term `startAuction`, holding the parameters product and amount, is used. The term `bid`, holding the parameter amount, is used to issue a bid. Furthermore, a term is needed to inform *Users* about the current price of a product, the current "bid-holder" and counter. The counter can either be 0, 1 or 2. If the counter is set to 2 the product has been sold. Every time a bid is issued that is higher than the current price of the product the counter is reset to 0. The term `currentOffer`, holding the parameters amount, agentName and counter, is used for this purpose. When the auction has ended two terms are needed to inform *Users* whether a product has been sold or not. For this purpose the terms `productSold` and `productNotSold` are used. The term `productSold` holds the winning bid's price. Finally, a term to hand out the product is needed. This is accomplished by the `handOutProduct` term, which references the auction product via the product parameter. This signals the participants of the auction also that it has ended.

To close the auction scene, the term `closeScene` is used. The final scene protocol represented as Finite State Machine is depicted in Figure 2.7.

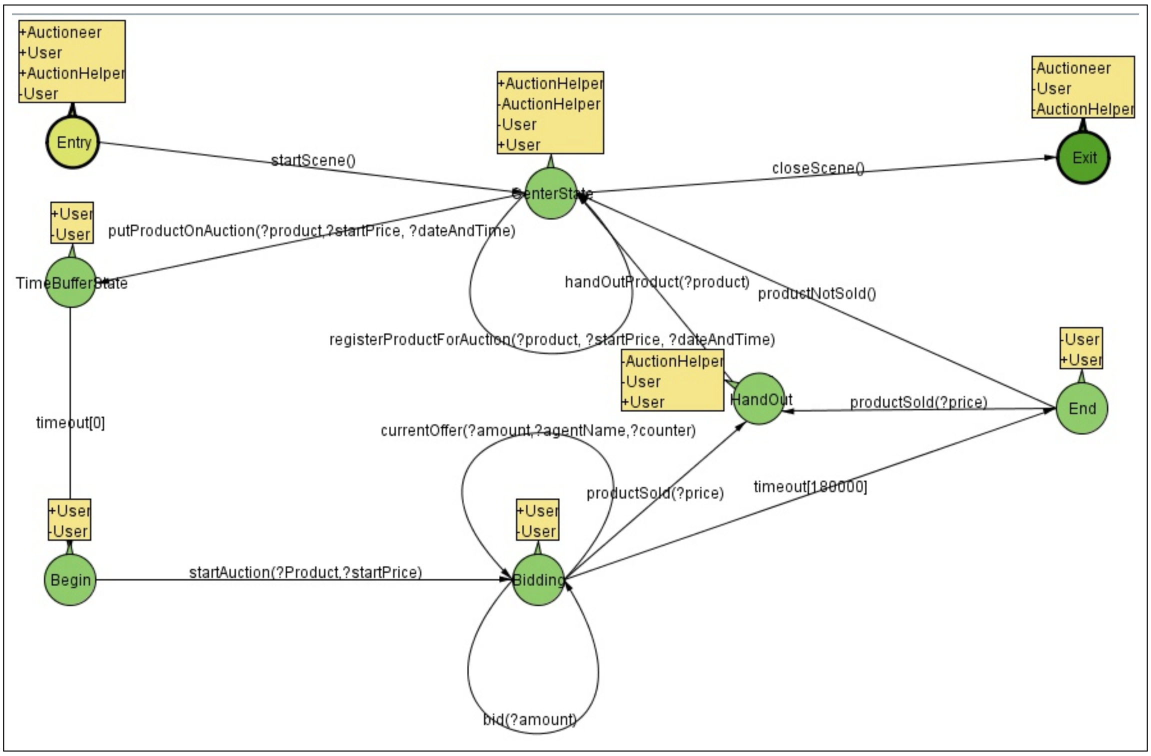

Figure 2.7: Auction Scene

The remaining rooms are summarized in the following. The good storage room is mapped onto the `Offering Scene` which is the only scene that is not accessible for agents playing in the *User* role. It is solely used for coordinating the internal work processes of the `Auction House` Electronic Institution. The duties of the transport staff are carried out by agents playing in the *Data Fetcher* role. Since we are dealing only with immaterial goods in the Electronic Institution, *Data Scout* agents fetch product data from various sources and convert it to a standardized format used within the Electronic Institution. The duties carried out by the auction manager are covered by an agent playing in the *Offer Manager* role in the `Offering Scene`. The `Offering Scene` corresponds to the good storage room in the auction house's blueprint.

The clearing room's purpose is to accept payment and hand out purchased goods. This room is realized as `Clearing Scene` in the Electronic Institution. The duty of the cashier, who is located in the clearing room is carried out by an agent playing in the *Account Manager* role. The information (lounge) room suits the general purpose of providing a meeting facility for customers.

The information room suits the purpose of keeping customers informed about traded goods in the auction house. We mapped the information room and information(lounge) room to the `Information Scene` in the Electronic Institution. The duties of the information staff in the auction house are carried out by several agents playing in the *Information Assistant* role. The role *User* accommodates all duties of a customer in the auction house. The role duty descriptions, which have been defined in Section 2.6.1, are used to derive roles and their relationships in the Electronic Institution.

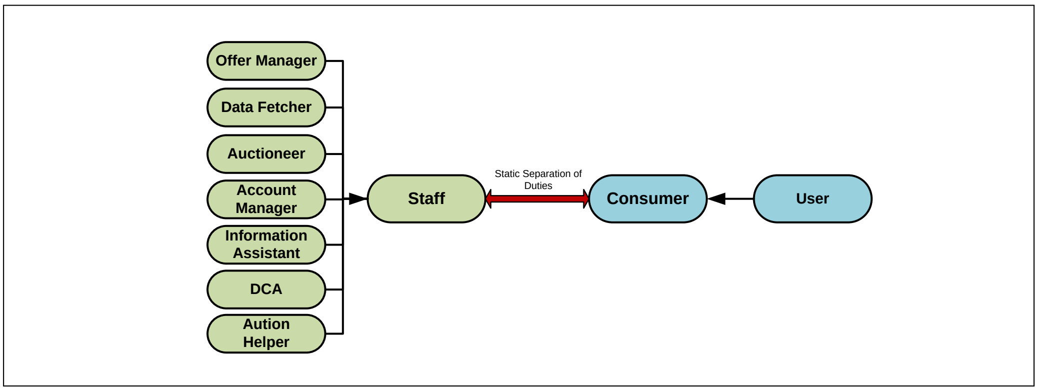

Figure 2.8: Auction House Agent roles

In Figure 2.8 the Dialogical Framework including the relationships among roles is presented. Agents can play 8 different roles whereof *Staff* and *Consumer* are base roles, depicted by generalization arrows from the more specific roles to the base roles. Static separation of duty means that an agent is not able to play roles connected via this relation at the same time. As depicted in Figure 2.8 *Offer Manager*, *Data Fetcher*, *Auctioneer*, *Account Manager*, *Information Assistant*, *DedicatedControlledAgent (DCA)* and *AuctionHelper* are subroles of *Staff*, meaning they have more specific duties but that all of them belong to the *Staff* role group. The *User* role is a subrole of *Consumer*. All roles depicted on the left side are subroles of *Staff* and all roles shown on the right side are subroles of *Consumer*.

Modeling & Mapping

To enable users to intuitively interact with the Electronic Institution, a 3D Electronic Institution is created based on the blueprint. A 3D Virtual World provides the supporting visual environment for the 3D Electronic Institution. The 3D model of the Electronic Institution can either be generated manually by using the created blueprint as guideline or via generic algorithms [34]. In order to enable users to participate in 3D Electronic Institution the actions of the user in the 3D Virtual World must be transformed into actions in the Electronic Institution. For this purpose a so-called controlled agent is assigned to every user. This agent executes the user's actions in the Electronic Institution. The message exchange between the user and the agent is carried out by the Connection Server (as depicted in Figure 2.3). Buildings are the visual representation of Electronic Institutions in the 3D Virtual World. By using a straight forward mapping approach, scenes of the Electronic Institution are mapped on rooms in the 3D Electronic Institution and transition are mapped on doors. However, only those scenes that are accessible to agents playing in the *User* role are modeled in the 3D representation of the Electronic Institution. Consequently, the `Offering Scene`, resembling the "good storage room" (and the "corridor" that connects the "good storage room" with the "auction rooms") in the blueprint, is not modeled in the visual representation of the Electronic Institution. The 3D representation of the Electronic Institution is depicted in Figure 2.9. For a detailed description of the middleware component and the Visualization Layer we refer to the PhD thesis of my colleague Ingo Seidel [84].

2.6.2 The Itchy Feet EM: Putting VOs to practice

The Itchy Feet Electronic Marketplace is dedicated to organizations in the e-Tourism domain. All organizations within the marketplace are Virtual Organizations realized as Electronic Institutions and software agents implement their business processes. The marketplace comprises three organizations that differ in terms of services and products offered. The creation of the Auction House was demonstrated in the previous section. In the Auction House users can participate in auctions to purchase flight or hotel products. In the Travel Agency fixed price products can be booked and users can get advice from professional travel agents. The Forum offers communication services and builds the cornerstone for the creation of a lively community. It encourages users to stay in Itchy Feet and therefore indirectly attracts potential customers to the other two organizations in the marketplace. The Itchy Feet Electronic Marketplace is depicted in Figure 2.10. The Travel Agency is realized as Virtual Organization comprising a B2C organization that offers customer centric services and a B2B organization that provides product data and booking facilities to the B2C organization. The customer centric services offered by the Travel Agency are flight and hotel search facilities, the advice of professional travel agents and a payment functionality that automatically performs the settlement with the B2B organization that supplies the products. The member organizations of the Virtual Organization use the `Ether` to exchange data with each other. The B2B organization itself can fetch product data from various external sources such as Internet travel agencies and provides an interface to the B2C organization to perform searches on these sources.

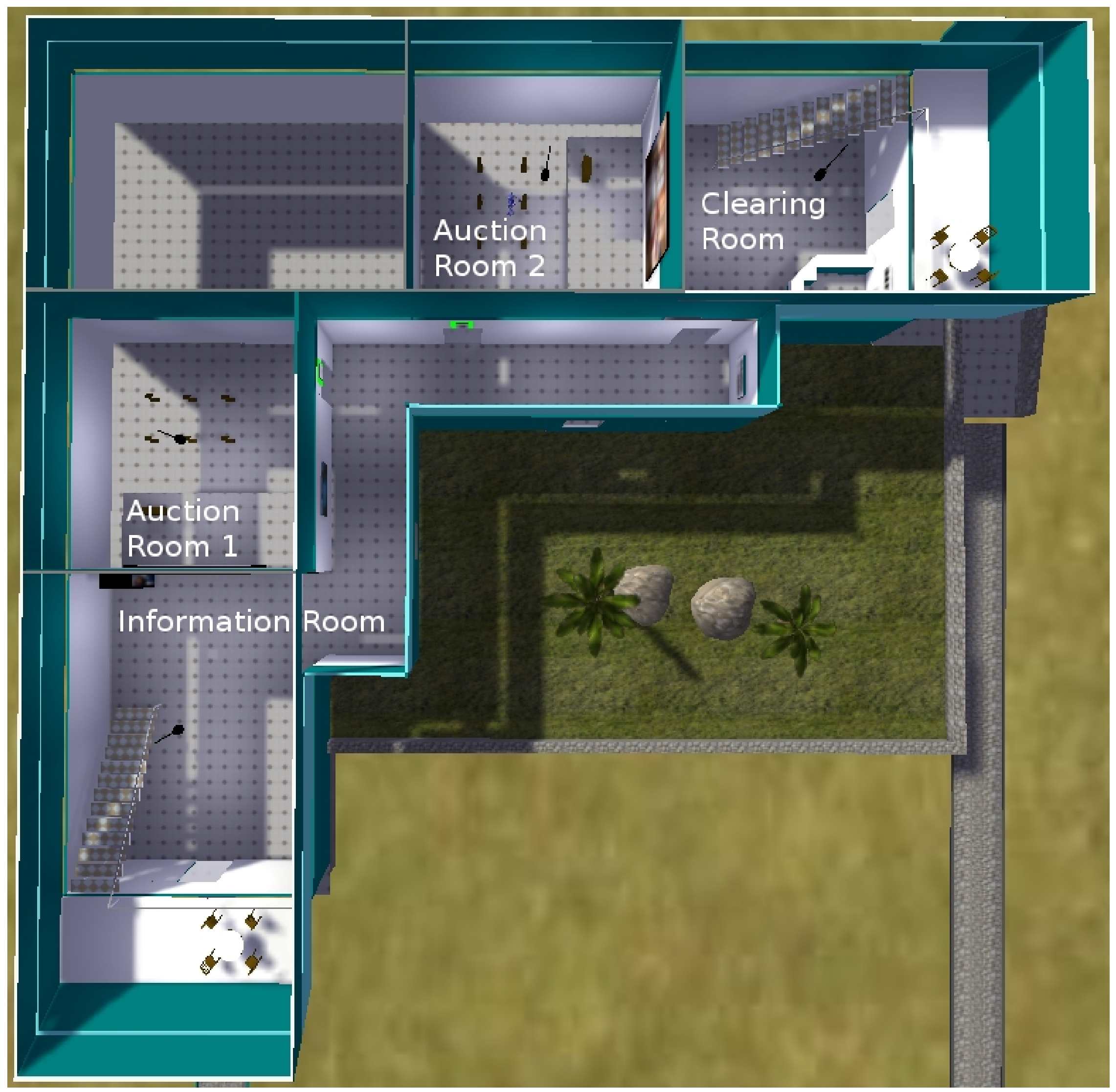

Figure 2.9: Floor plan of the Auction House

The B2B organization can be extended with additional external data sources via a defined interface (depicted by the "uses" relations in Figure 2.10). Furthermore, it is possible to connect additional B2B organizations with their own data fetching routines and workflows to the Travel Agency's Virtual Organization. The Auction House and Forum are Virtual Organizations that are realized via single Electronic Institutions, still offering the possibility to be extended by an arbitrary number of additional member organizations realized as Electronic Institutions.

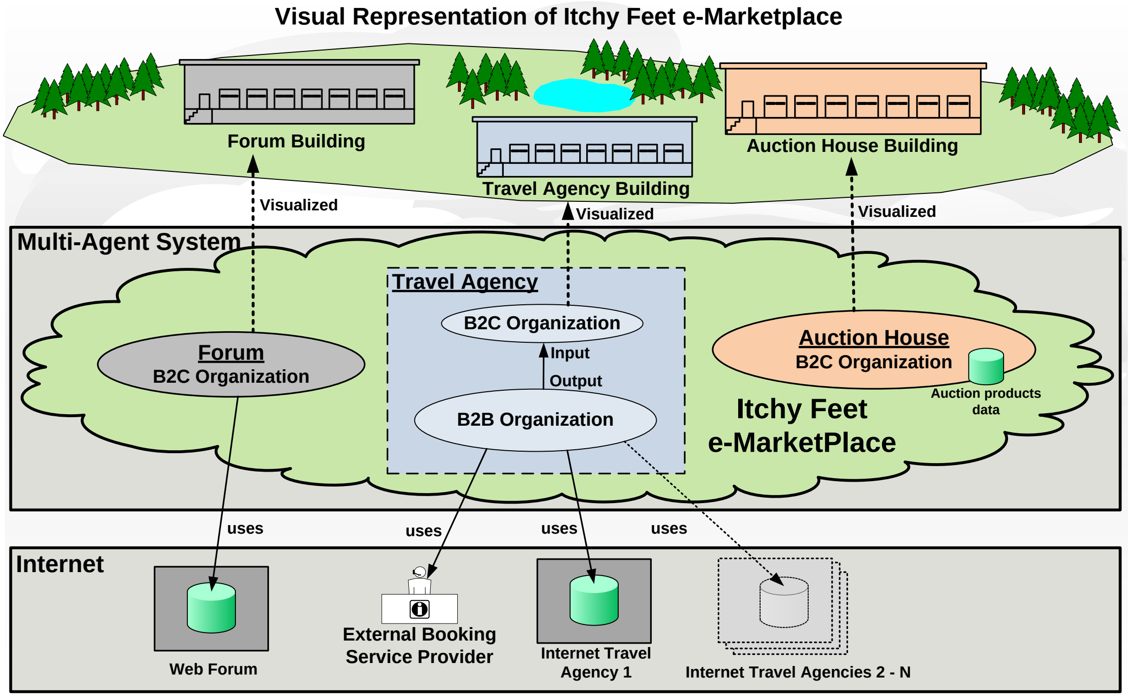

Figure 2.10: Organizations of the Itchy Feet e-Marketplace

Roles and access control within the Electronic Marketplace

The role concept of the Electronic Institution methodology is utilized to realize access control within the Electronic Marketplace. Access control is crucial to maintain security in an Electronic Marketplace, because certain users or autonomous agents may need access to scenes and transitions whereas others need to be kept away from these scenes and transitions. For example, users playing in the *User* role may not have access to internal scenes that are used by autonomous agents to exchange confidential information. Any Electronic Institution which is part of the Electronic Marketplace may define multiple roles for their scenes and transitions. Only those agents who play in appropriate roles are able to access these specific scenes and transitions. Roles can be grouped and dependencies such as *Static Separation of Duties* can be defined. Two roles that are related via *Static Separation of Duties* can never be played by the same agent. The role hierarchy and role dependencies are defined for the entire Electronic Marketplace. In case of Itchy Feet, roles are grouped into three main roles. The *Staff* role can be only played by autonomous agents, but the *Consumer* role and the *Employee* role can only be played by controlled agents, receiving their instructions from users, or from autonomous agents.

It is important to be aware that the term "*User*" printed in italic refers to a role that can either be played by a controlled agent or by an autonomous agent. However, the (non-italic) term "user" refers to the actual human user who can play in the role *User*.

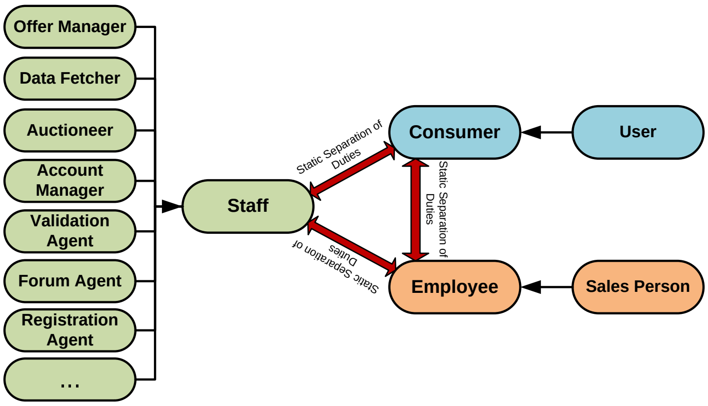

Figure 2.11: Roles and access control within the Electronic Marketplace

The *Employee* group has been added to Itchy Feet's role definitions to provide business partners with the possibility to define custom roles. Business partners may define their own *Employee* roles which are played by agents providing services to *Consumers*. Figure 2.11 depicts the main roles and sub-roles that can be played by autonomous and controlled agents in Itchy Feet. The roles presented in Section 2.6.1 are a sub sum of all roles that can be played in the Itchy Feet EM. A detailed role description is available in [51].

EIs of the Itchy Feet EM

The following detailed descriptions of the Electronic Institutions and Virtual Organizations of the Itchy Feet Electronic Marketplace has been published in [87]. The Finite State Machines of all presented scenes can be found in Appendix A.

The Ether EI The Ether Electronic Institution offers a validation and registration service. Every user who wants to participate within the Electronic Marketplace has to register in the Ether Electronic Institution first. Every time a user logs into Itchy Feet her credentials are validated in the Ether Electronic Institution. The Ether Electronic Institution has three scenes: the `Registration`, the `Validation` and the `Ether Scene`. The `Registration` and `Validation Scenes` implement the registration and validation service. The `Ether Scene` provides communication services and services for listing a user's inventory, shopping cart and profile. These services can be used inside and outside of any building in the 3D Virtual World. Figure 2.12 depicts the Performative Structure of the Ether Electronic Institution. The key symbols identify those agent roles that instantiate (open) new scenes. The different agent roles of the Ether Electronic Institution are shown in Table 2.2.

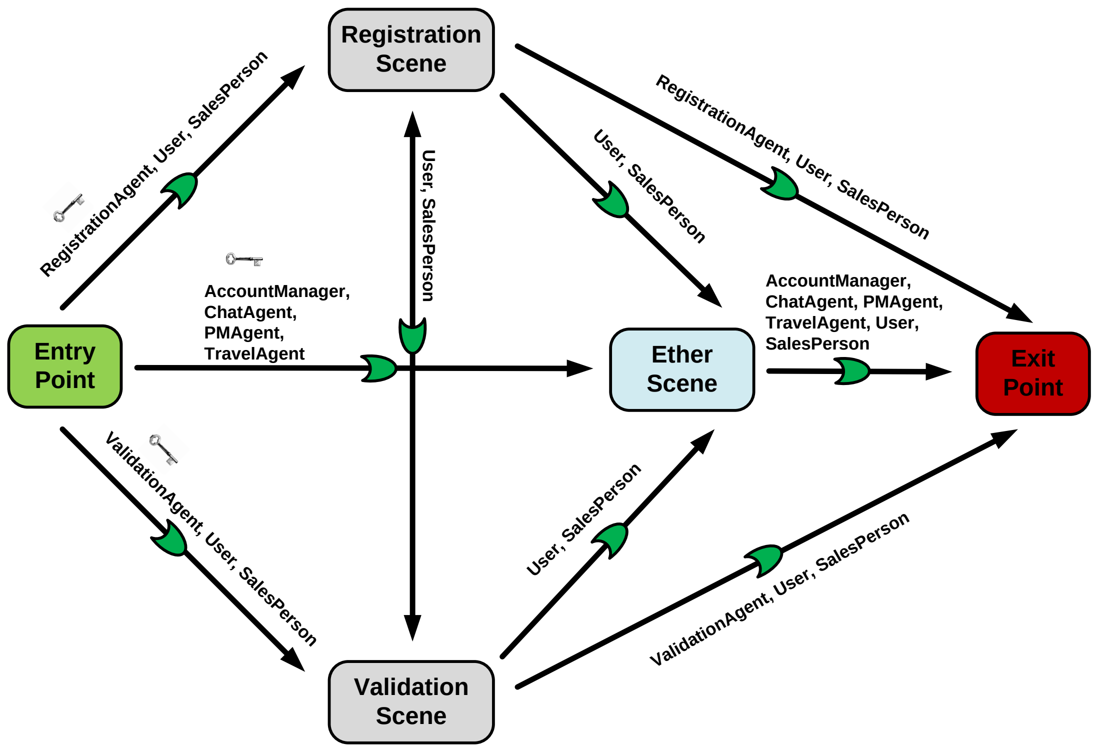

Figure 2.12: The Performative Structure of the Ether Electronic Institution

Agent role	Responsibilities
RegistrationAgent	Register new users
ValidationAgent	Validate existing users
PMAgent	Deliver private messages between users
ChatAgent	Provide a chat facility to users
SalesPerson	Offer advice to users
User	Participant of e-Marketplace
AccountManager	Offer shopping cart and inventory administration services
TravelAgent	Offer travel advice to users

Table 2.2: The agent roles of the Ether Electronic Institution

Private messages are delivered by the *PMAgent*. A chat is established by the *ChatAgent*. An agent playing in the *SalesPerson* role offers advice to users of the chat. The *AccountManager* provides services for listing the contents of the shopping cart and inventory. This agent also participates in the Travel Agency Electronic Institution and in the Auction House Electronic Institution where it accepts payments of users. The *AccountManager* is informed of every financial transaction in Itchy Feet. To illustrate the

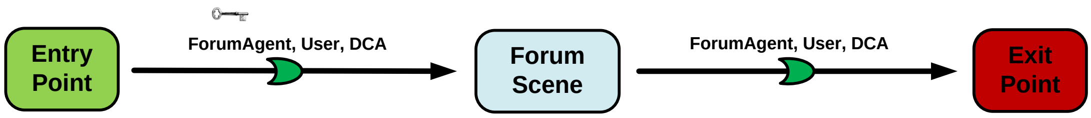

Figure 2.13: The Performative Structure of the Forum Electronic Institution

Ether's functionality, imagine a user playing in the *User* role. In this example as well as in the upcoming examples we assume that all scenes of the Performative Structure have already been instantiated by their scene owners (marked with the key symbol). The User moves from the `Entry Point` to the `Registration Scene` and registers with the *RegistrationAgent*. The *RegistrationAgent* creates a new user account and profile. At every subsequent visit of Itchy Feet, the user moves to the `Validation Scene` instead of the `Registration Scene` to get her credentials validated and to gain access. After the registration or validation procedure, the agent of the user splits into two instances and one instances moves directly to the `Ether Scene`, whereas the other instance of the agent can enter other Electronic Institutions of the EM. The agent instance in the `Ether Scene` stays there as long as the user is participating within Itchy Feet in order to be able to use the Ether's services. When the User exits Itchy Feet the agent leaves the Ether by moving from the `Ether Scene` to the `Exit Point`.

The Forum VO The Forum Virtual Organization is realized via a single Electronic Institution because it offers only customer centric processes. The Electronic Institution has only one scene and is the least complex of Itchy Feet. The Performative Structure, depicted in Figure 2.13, consists of the `Forum Scene` which is led by the *ForumAgent*. The Forum Scene is directly connected to the Entry and Exit Scenes of the Electronic Institution. The forum functionality has been realized by means of an open source forum, namely jForum[6]. A Web Service has been implemented that exposes the main functions of the open source forum such as topic or thread creation and fetching content from the forum to other software components. The *ForumAgent* uses a Web Service client to access these functions. Users that participate in the EM solely interact with the *ForumAgent*, which forwards their requests to the actual forum. Furthermore, users that do not participate in the EM can access the open source forum via an ordinary Web Browser. The different agent roles of the Electronic Institution are shown in Table 2.3.

An agent playing in the *DedicatedControlledAgent (DCA)* role provides information received from the *ForumAgent* to the public. Such an agent could be visualized as a panel

[6]http://jforum.net/

Agent role	Responsibilities
ForumAgent	Pass-through services of jForum
DedicatedControlledAgent (DCA)	Provide information from the ForumAgent to the public
User	Participant of the Forum

Table 2.3: The agent roles of the Forum Virtual Organization

in the 3D Virtual World where the most current or most interesting threads of the forum are displayed. To illustrate the Forum's functionality, imagine a user playing in the *User* role. The user moves from the `Entry Point` straight to the `Forum Scene`. In the `Forum Scene`, the user queries the *ForumAgent* about the latest threads. She browses the threads and wants to comment on a certain discussion. A request to create a new forum posting is sent to the *ForumAgent* which creates the posting in the external Web forum. The user then leaves the `Forum Scene` and moves to the `Exit Point`. The forum in Itchy Feet is somewhat special with respect to the other Virtual Organizations, because users may either interact with it via the 3D Virtual World or via a Web Browser. Hence, users participating in the 3D Virtual World can communicate with users on the Internet which use a Web Browser and the HTML-based presentation of the forum.

The Travel Agency VO The Travel Agency Virtual Organization is realized via two Electronic Institutions. It offers fixed price products. Customers may book flights, hotel rooms or all inclusive tours. The Performative Structure of the Travel Agency's B2C Electronic Institution is depicted in Figure 2.14. All customer centric processes are realized by agents in the Travel Agency's B2C Electronic Institution. In the `Information Scene` customers can inform themselves about products that are sold within the institution. Three `Sale Scenes` have been implemented for advising customers individually. Every `Sale Scene` is equipped with a *SalesPerson*, an external human travel agent, who shares its knowledge and expertise with customers helping them to establish a customized travel arrangement. Flights or hotel rooms can be booked in the `Booking Scene` and are paid in the `Clearing Scene`. Product information is transfered to the Travel Agency by the B2B Electronic Institution behind the scenes. The B2B Electronic Institution realizes business processes via agents that access a database and external data from travel agencies on the Internet to gather data. The payment is performed by credit card in the institution's `Clearing Scene`. The different agent roles played in the Electronic Institutions comprising the Travel Agency Virtual Organization are shown in Table 2.4.

An agent playing either in the *HotelDataFetcher* or *FlightDataFetcher* role provides

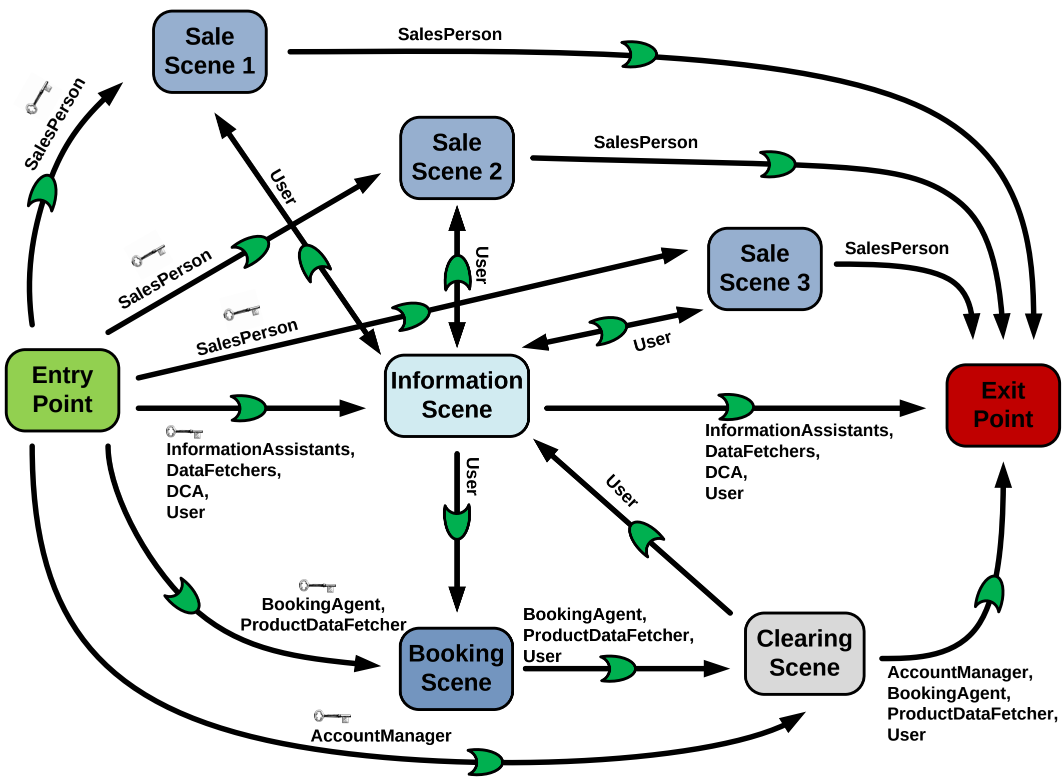

Figure 2.14: The Performative Structure of the Travel Agency Electronic Institution

Agent role	Responsibilities
HotelDataFetcher	Fetch hotel and room data from external providers
FlightDataFetcher	Fetch flight data from external providers
ProductDataFetcher	Fetch detailed product information from external providers
HotelInformationAssistant	Provide hotel and room information to users
FlightInformationAssistant	Provide flight information to users
BookingAgent	Offer booking services to users
SalesPerson	Offer advice to users
AccountManager	Offer payment services to users
DedicatedControlledAgent (DCA)	Offer hotel, room and flight data to the public
User	Participant of the Travel Agency

Table 2.4: The agent roles of the Travel Agency Electronic Institution

the Travel Agency with hotel and room data from external providers. These agents are participating in the B2B Electronic Institution of the Travel Agency Virtual Organization and interact with agents playing in the *InformationAssistant* role in the B2C Electronic Institution of the Travel Agency Virtual Organization. When a user performs a search, the *InformationAssistant* agent will query one of these agents to get corresponding data from registered external providers. A user who is undecided about her travel plans can move to one of the `Sale Scenes` and get advice from a *SalesPerson*. If a user spots a flight or hotel she is interested in, the product can be booked in the `Booking Scene`. To buy the flight or hotel arrangement a user moves to the `Clearing Scene` and settles the invoice with the `AccountManager`. An agent playing in the *DedicatedControlledAgent* role provides flight and hotel arrangements offered in the `Travel Agency` to the public, for example by means of a ticker. To demonstrate the functionality of the Travel Agency, consider a user playing in the *User* role. Starting at the `Entry Point` the user moves to the `Information Scene`. In the `Information Scene` the user decides to move to the `Sale2 Scene` to get advice on traveling to Scotland. Well informed the user leaves the current scene and heads back to the `Information Scene` where she queries the *FlightInformationAssistant* about flights to Scotland. Having identified a suitable flight, the user moves to the `Booking Scene` and books the flight. The flight is then paid in the `Clearing Scene` and the user leaves the B2C Electronic Institution by moving to the `Exit Point`.

The Auction House VO The creation of the Auction House VO by using the 3D Electronic Institution Process has been demonstrated in Section 2.6.1. The Auction House Virtual Organization is realized as single Electronic Institution because it uses an internal product database instead of an external information supplier. In the Electronic Institution, flight and hotel room arrangements are auctioned. The Performative Structure, depicted in Figure 2.6, shares similarities with the Travel Agency. The Auction House's `Offering Scene` is the central meeting place for the *OfferManager*, the *DataFetchers*, the *AuctionHelpers* and the *Auctioneers*. *DataFetchers* register products that have been fetched from the internal database with the *OfferManager*. The *OfferManager* then contacts the *Auctioneers* via the *AuctionHelpers* to sell them. Demand and supply is overseen by the *OfferManager*, which may instruct *DataFetcher* agents to bring in new products. The `Information Scene` is populated with *Users* and *InformationAssistants*. Users may query *InformationAssistants* about currently offered products. The products are auctioned in one of the three distinct `Auction Scenes` that use the English auction protocol, henceforth named `English Auction Scenes`. Every

Agent role	Responsibilities
OfferManager	Oversee demand and supply, instruct DataFetcher agents
Auctioneer	Lead auctions
AuctionHelper	Registers products that should be auctioned with the Auctioneer agents
DataFetchers	Fetch detailed product information from external providers
HotelInformationAssistant	Provide hotel and room information to users
FlightInformationAssistant	Provide flight information to users
AccountManager	Offer payment services to users
DedicatedControlledAgent (DCA)	Offer hotel, room and flight data to the public
User	Participant of the Auction House

Table 2.5: The agent roles of the Auction House Electronic Institution

`English Auction Scene` is populated with an agent playing in the *Auctioneer* role and several users playing in the *User* role. If a user has purchased a product, she has to move to the `Clearing Scene` to pay for it. The `Clearing Scene` is led by the *AccountManager* which accepts payment and assigns the ownership of a product to a user. The different agent roles of the Auction House and their responsibilities are summed up in Table 2.5. To illustrate the functionality of the Auction House, consider a user playing in the *User* role. Starting at the `Entry Point`, the user moves to the `Information Scene` and informs herself about ongoing auctions. An auction for a flight to the Canary Islands is held in the `English Auction Scene 2`. The user is interested in travel arrangements for the Canary Islands and directly moves to this scene. In the `English Auction Scene 2` the user engages in the auction, bids several times and is finally announced the winner. The user then moves to the `Clearing Scene` where she pays for the product and the ownership is transferred to the user.

The Auction House is independent from any B2B organization and uses an internal data storage to retrieve product data of auction products. The framework offers an administration tool to add new products to the internal data storage of an organization. Suppliers may use this tool to modify product descriptions or incorporate new products that can be sold via their organization in the Electronic Marketplace. The Auction House as well as the Forum have been realized via a single B2C organization. The Forum is connected to an instance of jForum[7] which has been adapted to support access via Web

[7]http://jforum.net/

Services. The Forum offers knowledge of professional travel experts and tourists.

The entire Electronic Marketplace is modeled in the virtual environment, ranging from the product search to the product selection to the actual payment. This is an advantage over existing systems such as Second Life[8] where the user is required to switch to the Web shop to buy a product. In our environment this system change is not necessary and the immersive experience is sustained. The virtual environment enables users to be aware of other users within the marketplace. This awareness brings a variety of advantages. Users are able to infer the attractiveness of a product by the amount of users that stroll through the supplier's organization. Similarly, the interest in an auction product can be assessed by the amount of users in the auction room. Furthermore, users are able to directly interact with each other and can share information about bargains or interesting products in the marketplace.

The functionality, which is available in the buildings of the 3D Virtual World, is defined in the Multi-Agent System. Every building corresponds to exactly one B2C organization which is realized as Electronic Institution in the Multi-Agent System. The Travel Agency building is the visualization of the `Travel Agency` B2C Electronic Institution, the Auction House building is the visualization of the `Auction House` B2C Electronic Institution and the Forum building is the visualization of the `Forum` B2C Electronic Institution. The scenes of a B2C Electronic Institution are visualized as rooms of a building and the transitions are visualized as doors. A user, impersonated as an avatar in the 3D Virtual World, may explore the world and has access to various functions depending on the avatar's current location in the 3D Virtual World. The user is the principal of an agent in the Electronic Institution which maps the movement of the user to the entities in the Electronic Institution. Whenever the user moves in the 3D Virtual World, the user's agent mimics the movement in the Electronic Institution.

2.6.3 Auctions in the 3D Virtual World

In this section the user interaction with the Electronic Marketplace, in particular the Auction House, is showcased. The `Auction House` Electronic Institution comprises five scenes: the `Offering Scene`, the `Information Scene`, the `Clearing Scene` and two `Auction Scenes`. We have only visualized two `Auction Scenes` as rooms in the building, because two `Auction Scenes` were sufficient to evaluate the e-Tourism setting. The `Offering Scene`, led by the *OfferManager*, is the control unit of the `Auction House`. The *OfferManager* overlooks the commodity flows and ensures that every product is put on auction at the scheduled time. The *AuctionHelper* is respon-

[8]www.secondlife.com

sible for delivering products from the `Offering Scene` to the `Auction Scene`. *Users* inform themselves about available products in the `Information Scene` and pay for auctioned products in the `Clearing Scene`. Two types of agents, namely *User* and *Auctioneer*, participate in the `Auction Scene`. The interactions between these agents are defined by the scene protocol, which is implemented as a Finite State Machine. Those scenes that are accessible by *Users* are visualized as rooms in the Auction House building. Consequently, all but the `Offering Scene`, which is not accessible to *Users*, are visualized as depicted in the floor plan in Figure 2.9.

The `Auction Scene` is visualized as one room of the Auction House in the 3D Virtual World. Before a user is able to actively participate in an auction, she needs to create an account and provide her credit card information. As long as no auction is taking place users are able to enter and exit the auction room. During this time users may use the auction room to have a chat about the product in the upcoming auction and find out why other users are interested in it. When an auction has started no additional users can enter the auction room. Users who have issued a bid are not eligible to leave the auction room before the auction has ended. This behavior is enforced by the auction protocol where software agents are not allowed to enter or exit the scene in this state. When a user tries to enter or leave the auction room during an auction, an error message is displayed and the door does not open for the user. In case the user disconnects from the 3D Virtual World, the user's agent takes control over the auction process and remains in the auction scene until the end of the auction.

In the following we are observing the user Christine who is participating in an auction via the 3D Virtual World. On the right side of Figure 2.15 the auction room and the auction interface are depicted. The auction interface provides the user with information about the current status of the auction and the item to be auction. Furthermore, it is equipped with a text box to enter a bid. Two of the avatars depicted in Figure 2.15 are the visualizations of autonomous agents and one is the visualization of a human user who participates in the auction. Every role has a specific outfit that is worn by the avatar playing in that role. The different outfits help users to identify the duties of each avatar.

Before Christine participates in an auction she informs herself which auctions are currently taking place by consulting the information panel in the information room of the Auction House. Having chosen the appropriate auction Christine enters the room in the Auction House. This action triggers a series of actions resulting in the user's controlled agent entering the `Auction Scene` in the Electronic Institution. Figure 2.16 depicts the actions that take place in the `Auction Scene` in Ameli, the Multi-Agent System execution environment, from Christine's controlled agent's perspective.

Figure 2.15: The 3D Virtual World with the Auction House building and the Auction Room

Before the *Auctioneer* starts the auction at a predefined time it checks whether there are at least two users participating in the auction. This is the case, since the users Christine and Erich participate in the auction. Before the auction starts the *Auctioneer* informs Christine and Erich about the product that is going to be auctioned and its minimum price of 60 Euro by issuing the `putProductOnAuction` message. This message triggers an action to display the product information and initial price in Christine's and Erich's user interface. Subsequently, the *Auctioneer* starts the auction by sending the message `startAuction`. To bid Christine and Erich make use of the auction interface in the auction room. As soon as one of them submits a bid a corresponding "bid" message is send by the user's controlled agent in the `Auction Scene` of the Electronic Institution. As can be seen in Figure 2.16, Christine issued a bid of 60 Euro. The avatar of Christine waves its hand to signal other users in the room that she issued a bid. In addition, to inform other users about the bid amount, a pop-up holding the bid amount is displayed above the head of the user's avatar. In general visual cues are used to provide users with feedback to executed actions in the Electronic Institution.

As the bid is valid, the new product price is shown via a message box above the *Auctioneer's* avatar and the auction interface of all users is updated with the current product price. As depicted in Figure 2.16 the 60 Euro bid of Christine is followed by a 70 Euro bid of Erich. Subsequently, the *Auctioneer* starts to count from zero to two in predefined time intervals and informs participants about the current price and counter by issuing a `currentOffer` message. The counter value 0 corresponds to "going once", the value 1

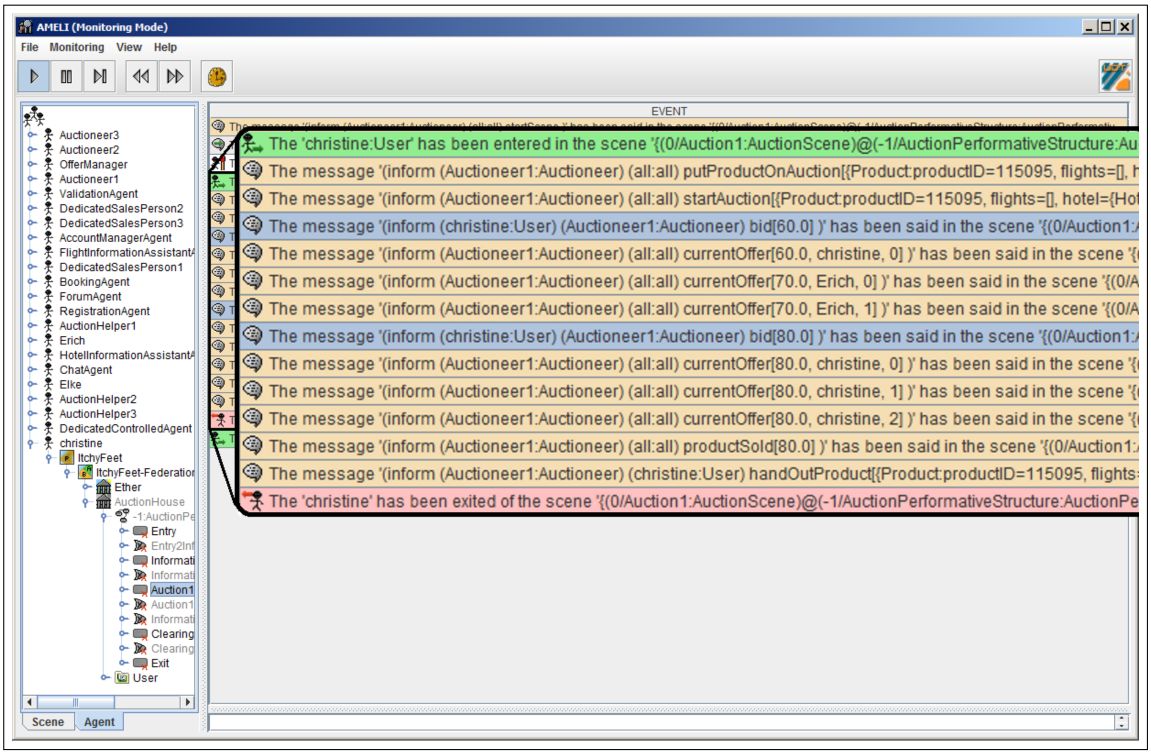

Figure 2.16: Multi-Agent System Perspective of Auction

corresponds to "going twice" and the value 2 corresponds to "sold". In case a user issues another bid before the counter reaches 2, the `Auctioneer` resets the counter to 0 and informs all users about the current offer. The "going once", "going twice" and "sold" actions of the *Auctioneer* are visualized by a gesture where the *Auctioneer's* avatar is knocking on the podium with a small hammer. In the presented showcase Christine issues a bid of 80 Euro before the `Auctioneer` is able to announce the product as being sold (the counter reaches 2) and is announced the winner. Subsequently, the *handOutProduct* message is send from the `Auctioneer` to Christine's controlled agent. When the auction is closed, the product is placed in the shopping cart of Christine. Subsequently, Christine needs to move to the `Clearing Scene`, visualized as clearing room, to pay for the product. In this room she needs to follow the payment protocol as defined in the `Clearing Scene` of the Electronic Institution. Christine may only leave the Auction House if she paid for the product or returned the product in case she accidentally bought it.

2.7 Discussion

In the following we will discuss our findings to the research questions defined in Section 1.2.1. Every research question is addressed in a separate subsection.

2.7.1 Enabling inter-EI communication

To answer ***Research Question 1***, asking which communication mechanism is suitable for information exchange in Virtual Organizations that are based on Electronic Institutions, we initially created two different approaches we deemed applicable. The first approach makes use of one major Electronic Institution to regulate the entire Electronic Marketplace and the second approach uses a special Ether Electronic Institution that regulates inter-Electronic Institution communication and provides common functionality to the Electronic Marketplace's participants. Subsequently, we reviewed the characteristics and restrictions of both approaches and incorporated the most suitable into our framework. Finally, we applied the approach to the Electronic Marketplace reference implementation presented in Section 2.6. The results are summarized in the following.

Due to similarities between Electronic Institutions and real-world institutions we mapped Electronic Institutions to buildings in the 3D Virtual World. As a consequence, scenes of an Electronic Institution are mapped onto rooms in order to ensure that any action performed by a user within a room can be validated by the Electronic Institution. Therefore, as long as a user is in a building, his actions are validated by the Electronic Institution. The problem is that the Electronic Institution methodology does not provide mechanisms to regulate actions outside an Electronic Institution. As a consequence the terrain outside a building in the 3D Virtual World cannot be regulated.

Similarly no facility for inter-Electronic Institution communication is available. In order to support the construction of Virtual Organizations, inter-Electronic Institution communication is essential for the exchange of data between the member organizations of a Virtual Organization. If an agent wants to communicate with an agent that resides in a different Electronic Institution it needs to join that one as well. In case an agent needs to communicate with several agents spread across multiple Electronic Institutions it has to join all of them. This is needed for example if one B2C organization needs to get information from two different B2B organizations in a VO. This would result in multiple instances of the same agent which only act as communication endpoints. In terms of the 3D Virtual World the absence of global communication functions implies that users can only communicate with each other if they are in the same building. Furthermore, without the possibility to define global functions that are shared by all Electronic Institutions,

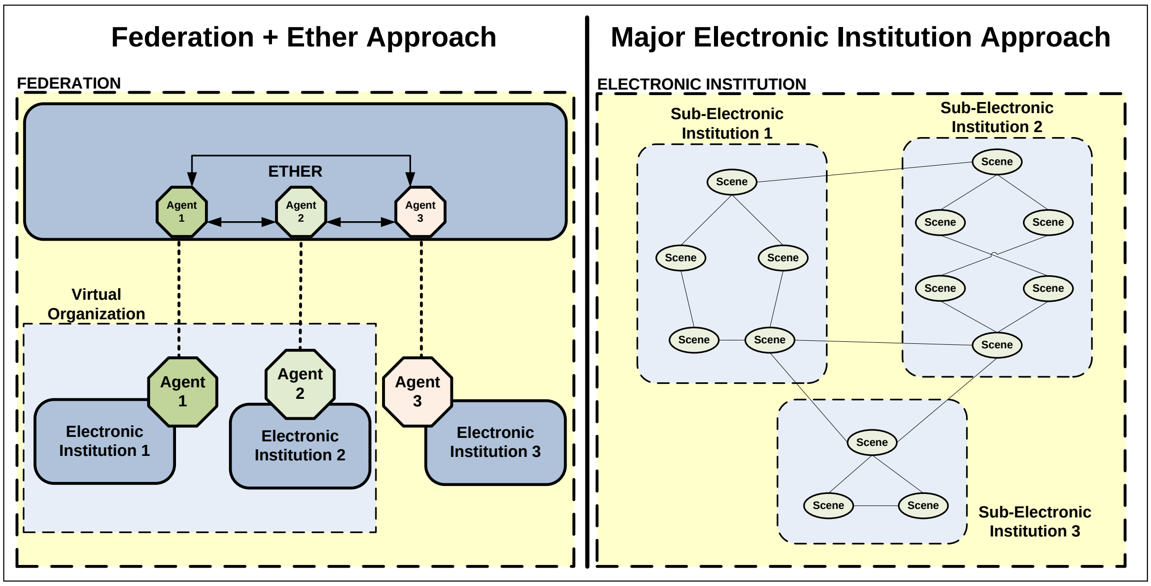

Figure 2.17: Design Approaches

every Electronic Institution has to implement functions that are vital for any successful Electronic Marketplace such as a shopping cart or an inventory separately.

To overcome the absence of a satisfactory inter-Electronic Institution communication facility, we considered two different approaches that are depicted in Figure 2.17. First, the Electronic Marketplace could be realized via one major Electronic Institution that comprises units that represent sub-Electronic Institutions. This approach has the advantage that mechanisms specific to Electronic Institutions such as controlled message exchange, role-based constraints and scene protocols can be used to model global functions. The major disadvantage of this approach is that the units (sub-Electronic Institutions) are not separated on the Multi-Agent System Layer and that no clear boundaries exist between them. Therefore, it is difficult and error-prone to extend the specification with additional units. Furthermore, since there is only one single specification of the entire Electronic Marketplace, it lacks extensibility, making it difficult for developers to define sub-Electronic Institutions independently. A developer may have already defined parts of a unit that interfere with another developer's specification. Since one of our main goals is to create an easy extensible marketplace we decided not to use the major Electronic Institution approach.

The second approach is based on the Federation construct. A Federation is used to group Electronic Institutions and defines a local name space for the creation of the Electronic Marketplace. This approach offers a high degree of extensibility as every Electronic Institution and Federation has a separate specification. A new Electronic Institution can be designed and developed independently from the Federation and existing

Electronic Institutions. The drawback of this solution is the impossibility to create global functions that can be shared by all members of the Federation such as a shopping cart, inventory or chat. Furthermore, the Federation does not provide any communication facilities to its members. To overcome these limitations we introduced an extension to the Federation concept named `Ether`.

The `Ether` is an enhancement of the Federation construct and is an Electronic Institution itself. It provides global functions that can be used by all Electronic Institutions that are part of the Federation. Furthermore, we incorporated a communication functionality to facilitate inter-Electronic Institution communication which is vital for the support of Virtual Organizations in the Electronic Marketplace. All agents of any Electronic Institution may communicate with all other agents in the Electronic Marketplace by using the `Ether`. We realized this by creating two instances of an agent: a global one that automatically joins the `Ether` and a local one which joins a specific Electronic Institution. The global instance is shown as small octagon and the local instance as big octagon in Figure 2.17. These two instances appear as one agent. The local instance is used to participate in the agent's native Electronic Institution, the global instance resides in the `Ether` and is used to exchange data with agents of other Electronic Institutions.

2.7.2 Pitfalls

To answer ***Research Question 2***, asking what needs to be considered when connecting a 3D Virtual World with an Electronic Marketplace realized as Multi-Agent System, we identified the pitfalls that occurred during the implementation of the Electronic Marketplace reference implementation presented in Section 2.6. The results are summarized in the following.

When connecting a 3D Virtual World with an Electronic Marketplace realized as Multi-Agent System some challenges need to be dealt with that do not occur when 3D Virtual Worlds and Multi-Agent Systems are considered separately. In the following we propose our solutions and best practices to overcome the identified restrictions and discuss design considerations and decisions we made when creating the reference implementation.

Multiple Agent Instances In Electronic Institutions, similar to other Multi-Agent System methodologies, agents may split into multiple instances and participate in multiple Electronic Institutions simultaneously. This behavior must be reflected in the 3D Virtual World to maintain a consistent state. Imagine an autonomous software agent that participates in multiple Electronic Institutions. In order to visualize this agent in the 3D Virtual

World multiple avatars are needed since every instance of the agent has its own behavior. For example, the *Account Manger* in Itchy Feet participates in the Auction House and in the Travel Agency Electronic Institution. Since users interact separately with both instances two avatars are needed in the 3D Virtual World for the visualization.

Human controlled agents in Itchy Feet are realized as software agents as well. Technically these agents could also split into multiple instances, but this raises several issues. Whenever a user's agent would split into a new instance, a new avatar would need to be created. Since a human user is not able to control avatars in parallel, the user would have to decide which avatar he wants to control. This places an additional burden on the user and makes it more complicated to use the environment. Furthermore, the 3D Virtual World would soon be crowded with inactive avatars and it remains unclear what happens if a user tries to interact with an inactive avatar of another user. Finally, the merging of multiple avatars into one avatar is another point of confusion for the user and further reduces the usability. Due to these drawbacks we defined that human controlled agents, contrary to autonomous agents, must not split into multiple instances to maintain usability.

Specifics of using a 3D Virtual World Even though a 3D Virtual World offers new interaction possibilities, it needs to be ensured that rather simple tasks do not get complicated due to awkward interaction patters imposed by the 3D Virtual World. Consider a user who wants to buy a product in a 3D Virtual World. There are multiple possibilities to implement a purchase process in a 3D Virtual World. One possibility would be to present the user with a virtual store and a 3D shopping cart which she can move through the store. When she decides to buy a product, she puts it in the 3D shopping cart. This resembles the shopping process customers are used to from real world. However, another possibility would be to present the user with a 2D interface form in the 3D Virtual World where she can choose what she wants to buy and put it in an electronic shopping cart. This resembles the purchase process users are used to from e-Commerce Web Sites. Comparing those two approaches the second one is more efficient, even though the first one might be more fun to some users. However, a couple of users might find the rather simple task of purchasing a product too complicated or tedious when implemented via a virtual store and 3D shopping cart.

2.7.3 Mapping buildings to EIs

To answer ***Research Question 3***, asking to which extent buildings in the 3D Virtual World can be mapped onto Electronic Institutions in the Multi-Agent System, we applied the 3D

Electronic Institution Implementation Process presented in Chapter 2.6.1 to the three Virtual Organizations that are part of the Electronic Marketplace reference implementation presented in Section 2.6. However, a few concessions had to be made to accomplish a concise mapping but still maintain usability. The results are summarized in the following.

To accomplish a concise mapping of buildings in the 3D Virtual World onto Electronic Institutions and still maintain usability we had to introduce two mechanisms, namely *advanced room mappings* and *transition constraint validations*.

Advanced room mappings The more complex an Electronic Institution is, the more difficult it becomes to perform straight forward mappings from scenes to rooms and from transitions to doors. In some cases it makes sense to map multiple scenes to one room for usability reasons. This raises the problem that even though the agent's avatar moves into just one room, the actual agent in the Electronic Institution has to move through multiple scenes.

To overcome this problem we incorporated a functionality to divide a room of a 3D Electronic Institution into distinct areas. If a user's avatar moves from one area to another, the agent moves into the scene that corresponds to that area. By using this functionality, it was possible to map the `Information Scene` and the `Clearing Scene` to the same room in the Travel Agency building. If a user's avatar approaches the ticket counter, representing a distinct area in the information room, the agent moves from the `Information Scene` to the `Clearing Scene`. Furthermore, we extended the functionality to map single actions performed in the 3D Virtual World to multiple actions in the Electronic Institution. Therefore, tedious tasks that are executed by a user's controlled agent in the Electronic Institution need not to be mimicked by the user in the 3D Virtual World.

Transition constraint validation Another problem we faced was the fact that Ameli provides no information about the validity of transitions during runtime. Transitions are used to move from a scene to a subsequent scene in an Electronic Institution and are mapped onto doors. The Performative Structure of an Electronic Institution defines which transitions can be used to move from one scene to another and which constraints must be satisfied to move along the transition. Constraints are expressions that have to be evaluated during runtime. Only if a constraint is met, an agent may pass through the transition. Ameli offers a function to retrieve the outgoing transitions from a scene, but this returns all transitions that have been defined in the specification without taking the runtime constraints into account. A constraint is evaluated by Ameli only at runtime when an agent actually moves through the transition. Imagine a user who approaches the exit door in the 3D Electronic Institution with his avatar and that he has not paid for the products in his shopping cart. The user's agent issues a call to Ameli to get all valid outgoing transition. One of them is the transition that is mapped onto the exit door the user is currently approaching. This transition has a constraint that all products in the shopping cart of an agent need to be paid before the agent is allowed to pass through the transition. Since Ameli does not evaluate the constraints of the transition at this point it returns the transition as valid even though the user's agent has unpaid products in its shopping cart. As a result the avatar in the 3D Virtual World is able to move through the door, but the agent in the Electronic Institution gets stuck in the transition because the constraint is violated. As a result the agent can neither move forward nor backward and has to wait until the target scene becomes accessible. This leads to an inconsistent state between the 3D Virtual World and the Electronic Institution.

Therefore the validation of constraints needs to be done before the agent actually tries to move through them. To overcome this shortage we introduced a pre-validation logic with so-called “Back-Transitions”. Before an avatar can move through a door in the 3D Electronic Institution, the agent first enters the corresponding transition in the Electronic Institution. Only if the agent is allowed to pass through the transition, the door will open and the avatar can move through the door. If the user then decides not to walk through the door, the agent uses the “Back-Transition” to get back into the origin scene. In case the agent cannot pass through the transition, the “Back-Transition” is used by the agent to get back to the origin scene and the door will not open for the avatar. An illegal move of the user is also resolved by these “Back-Transitions”. If a user slips through a door that has been opened by another user, the agent automatically enters the transition. If the agent is not allowed to move on, it uses the “Back-Transition” to get back to the origin scene. As a consequence, the avatar is teleported back to its origin position in the 3D Virtual World and an error message is shown to the user.

2.7.4 Usability

To answer ***Research Question 4***, asking whether a Multi-Agent System that is used to regulate an Electronic Marketplace affects the Electronic Marketplace's usability, we conducted an usability study with the showcased reference implementation of an Electronic Marketplace. The results are summarized in the following.

As jointly elaborated with my colleagues and presented in [51, 84] a usability study was conducted in June 2009. A total of 20 people participated in the user evaluation. Every user had to complete 18 different tasks. The 18 tasks were integrated in a scenario that resembled the typical usage of an e-Tourism Electronic Marketplace. The majority of users did not have any problems navigating in the virtual environment. The few people who had difficulties in controlling their avatar in the beginning of the evaluation, got accustomed to this type of navigation towards the end of the evaluation. Regarding the Multi-Agent System and 3D Virtual World combination, there were no slowdowns perceivable by the participants while moving through the 3D Virtual World. The message exchange between the framework components was fast and no communication errors occurred. Furthermore, the state of the Multi-Agent System and the 3D Virtual World were consistent throughout the entire evaluation. It was completely transparent to the test persons that they are actually participating in an Electronic Marketplace that is regulated via a Multi-Agent System. However, a couple probands stated that it takes a considerable amount of time to get from one building to another. This is partly imposed by the 3D Electronic Institution Framework due to the metaphor of a building that is used for the visualization of VOs and Electronic Institutions. A possible solution would be to model one big Virtual Organization that accommodates all Electronic Institutions and which is visualized as single Building. Another solution would be to provide the users with a teleportation functionality that helps users to get quickly from one building to another. In a nutshell, the result of the evaluation was that the probands did not experience any major difficulties participating in the Electronic Marketplace controlled via the Multi-Agent System. For a complete discussion of the user evaluation we refer to [84].

2.7.5 Suitability of VOs to regulate EMs

To answer ***Research Question 5***, asking whether Electronic Institutions are suitable for forming Virtual Organizations to regulate an Electronic Marketplace, we draw a conclusion from the results of the previously listed research questions. The results are summarized in the following.

Based on the presented results the conclusion can be drawn that Virtual Organizations are suitable to regulate Electronic Marketplaces to a certain extent. By using the 3D Electronic Institution Framework and a supporting implementation process such as the 3D Electronic Institution Implementation Process, Virtual Organizations can be defined in a straight forward manner. As Virtual Organizations are based on Electronic Institutions they can make use of the same mechanisms to regulate the interaction of users and autonomous agents. The issues of regulating agents outside Electronic Institutions and missing inter-Electronic Institution communication has been overcome by introducing the `Ether`, which provides a regulated environment when agents move outside an Electronic Institution and enables inter-Electronic Institution communication by the usage of local and global agents. However, when Virtual Organizations are visualized in a 3D

Virtual World certain design approaches and mechanisms such as transition constraint validation, advanced room mappings and how to deal with multiple agent instances need to be considered. Still, it is up to the creator of the Electronic Marketplace whether to visualize its Virtual Organizations by means of a 3D Virtual World or just offer a traditional form based Web Application that enables users to delegate actions such as purchasing a product to agents participating in a Virtual Organization.

2.8 Summary

We have presented the 3D Electronic Institution Implementation Process supporting the creation of (3D) Electronic Institutions based on a formal role description of agents and a blueprint of the organization to be modeled. To showcase the application of the Electronic Institution Implementation Process, we created a prototype of an auction house realized as 3D Electronic Institution. We described the process of deriving an Electronic Institution's specification from a (hypothetical) real auction house. We identified roles of a real auction house, designed a supporting environment and used this information to specify the Electronic Institution at the Multi-Agent System layer. We demonstrated the functionality of the 3D Electronic Institution by means of a bargaining example involving two customers.

Furthermore, we showcased how organizations, realized as Electronic Institutions, can be connected to form Virtual Organizations that are visualized as 3D Electronic Institutions on the basis of the Travel Agency in Itchy Feet. To this end, a framework has been developed for the connection of Virtual Organizations and Electronic Institutions with 3D Virtual Worlds and a prototypical e-Tourism environment has been implemented. The marketplace in this environment is realized by means of auctions and fixed price product trade. The business processes are hereby implemented by software agents which are regulated by Electronic Institutions and Virtual Organizations. In particular, we have illustrated how auctions are conducted, how users are able to participate in a natural way and how the connection between users and Virtual Organizations and Electronic Institutions works. We emphasized the need for a construct that regulates the behavior of users when they are not inside a 3D Electronic Institution. We introduced the Ether as an extension to the Federation construct, in order to facilitate inter-Electronic Institution communication and to create global functions that can be used by all member of the Electronic Marketplace. Furthermore, we discussed the problems that arise when multiple instances of user controlled agents are permitted. We assessed the usability of our Electronic Marketplace reference implementation. Finally, we pointed out pitfalls that need to be considered when visualizing Virtual Organizations in a 3D Virtual World and proposed best practices to overcome them.

The presented framework supports the creation of complex Electronic Marketplaces by means of Electronic Institutions and their grouping to Virtual Organizations. The 3D Interface provides users with an easy and intuitive way to participate in the Electronic Marketplace by means of buildings in a 3D Virtual World. However, to fully leverage the possibilities offered by an Electronic Marketplace and to get the biggest benefit out of it customers need to make informed decisions. Hence, customers need to know the answer to questions such as “what to buy”, “when to buy”, “when to sell”, “from whom to buy” and “at which price to buy”. Sellers need to know the answer to question such as “what to sell”, “when to sell”, “to whom to sell” and “at which price to sell”. In a nutshell users of the Electronic Marketplace need relevant information to base their decisions on. Relevant information is crucial to make informed decisions. To stick to our e-Tourism example, consider a customer who wants to go on vacation to Scotland and has a detailed vision about what she expects of her trip to Scotland. She knows what type of commodities the hotel should offer, what kind of attractions should be in walking distance and that the hotel should have a good reputation and be child-friendly. Even tough she knows precisely what she is looking for, she first needs to consult several sources to find a hotel that meets her expectations. Moreover, not all information she needs might be available from a single source but rather scattered across multiple sources. A tremendous amount of information is currently available on the Internet, which can assist users during their decision process. However, information on the Internet is scattered over multiple sites and to a great extent unstructured. Therefore, due to its scattered and unstructured nature it is difficult to get relevant information. However, in the late nineties the Semantic Web emerged as extension to the World Wide Web offering structured information.

In Chapter 3 we will have a closer look at both information sources and introduce an approach that bridges them to use the strengths of one source to reduce the weaknesses of the other and vice versa. This approach can be used to provide users and agents of the Electronic Marketplace with relevant information for their decision process. We present a reference implementation of our approach which can be accessed by autonomous agents via a Web Service or by users via an interactive ontology-aware keyword-based Web interface. The implemented system can be accessed via autonomous agents playing in the role *Sales Agents* from within the 3D Virtual World or by users via a common Web Browser.

CHAPTER 3

Acquiring relevant information

Acquiring relevant information by means of Hybrid Semantic Search

Parts of the following chapter have been submitted to [52, 49].

3.1 Introduction

As briefly discussed in the previous chapter, having the right information at the right time is crucial in e-Business. Customers as well as sellers align their course of action to information they get from various information sources. One source to get most recent information is beyond doubt the Internet. However, when using the Internet as information source various challenges arise. Most information on the Internet is only available in the form of unstructured data, is unfiltered and spread over multiple sites. To make use of this information it needs to be gathered, parsed, structured to a certain degree, consolidated and made searchable in an efficient way. Data on the Internet can be roughly categorized into structured and unstructured data. HTML Web pages, the most widespread unstructured data sources, still make up the majority of data on the World Wide Web. However, in recent years structured data has become more widely available as new content was created, with the goal to make the information readily processable and searchable for machines. This idea was one of the main reasons for the emergence of the Web of Data, the Semantic Web. The Semantic Web can be considered as an extension to the World Wide Web, that provides structured information in the form of meta-data. The RDF[1] data format has become the de-facto standard to make structured data available on the Semantic Web.

[1] http://www.w3.org/RDF/

Hence, there are still two major technologies that coexist: the World Wide Web, which mainly consists of unstructured data and the Semantic Web, which enriches the World Wide Web with structured data. Given that the World Wide Web has been rapidly growing ever since the early nineties, it holds an enormous amount of knowledge that is scattered as unstructured data over billions of sites. However, even though the Semantic Web cannot compete with the World Wide Web in terms of pure amount of data, it has the significant advantage that its data can be accessed and searched more concisely due to its structured nature, resulting in more specific and accurate search results. Therefore, it would be desirable to bridge the structured data of the Semantic Web with the unstructured data of the World Wide Web to get the best of both worlds. Hence, use the concise access to information provided by the Semantic Web and leverage the existing information richness of the World Wide Web. Since it is not feasible to manually transform existing unstructured data on the World Wide Web to structured data suitable for the Semantic Web, several automatic approaches have been proposed [80, 77] that create meta-data by extracting and annotating information originating from the World Wide Web. These approaches use structured data from relational databases or Triples Stores to assist the extraction and annotation process [65]. The generated meta-data can be used to augment unstructured data from the World Wide Web and help to make it searchable in a concise and efficient manner.

The two most common search approaches are keyword-based and concept-based search. The keyword-based search approach is the dominant search form for unstructured data on the World Wide Web and the concept-based search approach is the dominant search form for structured data on the Semantic Web. Therefore, depending on the underlying technology a different search approach is applied. For several years these two approaches have been researched separately [83, 89, 55], but only recently researchers started to explore the possibilities of combining structured and unstructured data and use hybrid search approaches [10, 12, 94, 53].

Bhagdev et al. [10] have created K-Search, one of the most promising hybrid search implementations, which seeks to combine searches upon structured and unstructured data. In their approach unstructured data is used as a substitute for structured data whenever no structured data is available. The non-existence of structured data may be the result of a lack of suitable concepts in the used ontology or inaccurate Information Extraction (IE) processes, which are not capable of recognizing certain concepts of the ontology. Bhagdev et al. [10] and Bikakis et al. [12] have shown that the combination of keyword-based and concept-based search on structured and unstructured data results in a higher precision and recall when compared to precision and recall of the individual approaches on the same datasets. Furthermore, a user study was conducted that showed that the contestants preferred the combined approach over the pure keyword-based and concept-based approaches due to its expressiveness. However, K-Search uses two separate indexes instead of a combined index for the hybrid search and combines the result afterwards by using intersection on the document URIs of the result. The keyword-based part of the query is issued against a document index generated by SolR[2] and the concept-based part of the query is issued to a SESAME Triple Store[3].

The usability and acceptance of a search system is highly related to the used input mechanisms [56]. Several different input paradigms are used by keyword-based and concept-based search engines. The prevalent keyword-based search engines such as Google[4], Yahoo[5] and Bing[6] use a clean and intuitive interface that consists of an input text field and a search button. Advanced input options are mostly hidden and are only presented on demand to the user. In contrast there are several Semantic Search Engines, which either use a custom input mechanism [7] or can be queried by using high level query languages such as SPARQL [79], SeRQL [16] or other RDF-based query languages [21]. The major drawback of Semantic Search Engine user interfaces is that they are either complex and somewhat overcrowded with various input fields or that users need to know a rather complex high level query language, preventing the average Internet user from making efficient use of the Semantic Search Engine. Most users are used to the clean input paradigm of keyword-based search engines. Users may need a considerable amount of time before they get used to the custom input mechanisms of Semantic Search Engines. Furthermore, the average Internet user is not willing to learn a specific query language such as SPARQL or SeRQL to make use of a Semantic Search System. Even tough systems have been introduced that transform natural language into SPARQL queries [62], the average response times of these systems are very high and not comparable to response times of keyword-based search engines such as Google, which have response times in the area of fractions of a second. Another issue of many Semantic Search Systems is the lack of result ranking mechanisms. Without result ranking, users do not get the most relevant results presented first. The ranking of the result is common practice in purely document based search systems, where the most prominent ranking mechanisms use variations of the Vector Space Model and *tf-idf* weighting [82].

[2]http://lucene.apache.org/solr/
[3]http://www.openrdf.org/
[4]http://www.google.com/
[5]http://www.yahoo.com/
[6]http://www.bing.com/

3.1.1 Our contribution

The majority of hybrid search systems conduct their concept-based searches upon a Triple Store, which maintains structured data as RDF triples. In addition the keyword-based query is executed on a document index and both results are merged afterwards. However, Triple Stores tend to have higher query response times as they grow in size than systems that solely use document indexes. Still, Triple Stores are more flexible and can handle more complex queries. Therefore, we seek to overcome this issues by using a specific combined index structure based on SIREn [29] that can be used to conduct efficient hybrid searches without using two different indexes and still maintain the flexibility of a Triple Store to cope with complex queries.

Another issue with systems that seek to perform a combined search on structured and unstructured data is the acquisition of suitable documents containing unstructured data that complements the structured data stored in the Knowledge Base (KB). It is a tedious task to manually identify and add documents that are valuable with respect to the used ontology and KB. Hence, we seek to utilize the information richness of the World Wide Web to automatically acquire additional knowledge, by fetching relevant unstructured data to complement the structured data in the KB. Subsequently, we use the KB and ontology to enrich the acquired information with meta-data and bridge structured and unstructured data, to facilitate hybrid searches.

It is common practice for search systems to rank results from the most relevant to the least relevant, so users get the most relevant information presented first. In contrast, Semantic Search Systems hardly use ranking mechanisms to sort their results. Ranking becomes even more relevant when unstructured and structured information is combined. We use a ranking approach that is based on the user's query formulation and relevant data in the KB and document index.

Finally, we introduce a novel input mechanism for hybrid semantic search that combines the clean and concise input mechanisms of keyword-based search engines with the expressiveness of the input mechanisms provided by Semantic Search Engines. In addition, this interface can be used to interactively formulate queries without prior knowledge of the underlying ontology. To the best of our knowledge no other Semantic Search Engine offers this possibility.

In this work, we propose the Hybrid Semantic Search System (HS^3) which seeks to bridge structured and unstructured data to facilitate efficient hybrid semantic search by combining keyword-based and concept-based search approaches. This is accomplished by i) automatically fetching relevant unstructured information from the World Wide Web using an arbitrary ontology and KB, ii) annotating the information with meta-data from the KB, iii) creating a combined index that facilitates efficient hybrid semantic search, iv) providing an interactive ontology-aware keyword-based input mechanism that

HS^3 can be applied to an arbitrary domain. Even though it has been developed for the tourism domain the general applicability of the system has been demonstrated by applying it to the news domain.

We start by presenting related work in Section 3.2. In Section 3.3 we discuss the advantages of a hybrid search approach. In Section 3.4, we give a formal definition of hybrid semantic search and an overview of HS^3's architecture and its components. This is followed by an in-depth presentation of the combined index structure, the hybrid search approach and the ranking mechanism. In Section 3.5, we showcase the functionality of the interactive ontology-aware keyword-based input mechanism. In Section 3.6 we present the application of HS^3 to the tourism domain followed by the presentation of performance test results based on a tourism dataset. In Section 3.7 we demonstrate that HS^3 can easily be applied to an arbitrary domain and present answers to the research questions defined in Section 1.2.1.

3.2 Related Work

3.2.1 Semantic Search Systems and Hybrid Search Systems

In this section a detailed description of Semantic Search Systems and hybrid search systems are given. Every subsection is dedicated to a specific system and describes its features as well as mechanisms being used. A summary of features and drawbacks of the described Semantic Search Systems is presented in Table 3.1.

Ontology-based Information Retrieval System

Castells et al. [21] developed a Semantic Search System that is capable of performing keyword-based and concept-based searches. Their main intent was to develop a Semantic Search System that retains precision and recall of keyword-based search when information in the KB is incomplete or even not available. Mayfield et al. [70] stated that semantic search should be a complement to keyword-based search as long as not enough ontologies and meta-data is available. In addition to the system's KB, a document repository is used in which every document is annotated with at least one entity from the KB. A Vector-Space Model is used to rank search results and to overcome the specific characteristic of unstructured information in a structured information system. The system uses the RDF Query Language (RDQL), which is a graph-based query language for RDF. Apparently, RDQL is the only way to interact with the system. The system interprets the RDQL query and issues a request to the KB to retrieve a list of entities that match the query. For this step the system uses a Boolean model, having the consequence that only exact matches are returned. Subsequently, documents annotated with entities that are included in the result set are retrieved, ranked and presented to the user. The system uses the cosine distance for concept-based and keyword-based search result ranking and the *CombSUM* strategy, which computes the ranking score by a linear combination of the input scores, to combine both rankings. A disadvantage of the system is that it may take up to 30 seconds until a result is presented to the user. HS^3 uses also a KB and a document repository where every document is annotated with at least one concept from the KB. In contrast, HS^3 accomplishes response times that are less than one second using a KB and document repository that are similar in size to the ones used by Castells system.

ESTER

Bast et al. [7] developed a Semantic Search System named ESTER (Efficient Search on Text, Entities, and Relations) that leverages keyword-based search to speed up the retrieval process. The system can be used with an arbitrary ontology and corresponding KB. Nevertheless, only the usage of the YAGO [90] ontology and a KB, created from semi-structured information extracted of Wikipedia, has been documented so far. The system uses a semi-supervised method that extracts instances and their relation from the semantic information encoded in Wikipedia links and stores them in the KB. ESTER consists of three components, namely the query engine, the entity recognizer and the user interface. The entity recognizer relates words or phrases in the document corpus to the corresponding entities in the KB. The query engine is aware of documents and words and uses the basic operations prefix search and join to conduct searches. ESTER uses so-called artificial words that are inferred from the entities contained in the KB and annotated to the documents in the document corpus. The index used by ESTER consists of natural and artificial words. Every artificial word is equipped with a prefix. An example for an artificial word is `Musician:John_Lennon`, which defines that *John Lennon* is a musician by using the `Musician` prefix. These artificial words are used during the retrieval process to perform semantic searches on the document corpus. A detailed example and how combined searches are conducted is given in [7]. HS^3 and Ester use an interactive user interface to formulate queries. In contrast to ESTER the user interface of HS^3 allows the user to build graph-like queries and provides the possibility to explicitly label input terms as concepts, instances or keywords.

KIM

The semantic annotation platform KIM developed by Popov et al. [77] provides also a semantic search functionality. However, KIM's main area of application is the extraction and annotation of entities in documents. GATE [25], a Natural Language Processing (NLP) and IE platform, is used by KIM for Information Extraction. The platform is equipped with an upper-level ontology, namely the KIM Ontology (KIMO), and KB that build the base for further extraction of knowledge from a corpus of documents. KIMO defines entity classes, relations, attributes and lexical resource types. The latter are used to identify new entities in documents. These share similarities with the bags of labels used by Castells et al. [21]. An example of a lexical resource type is a list of common first names, which aids the Information Extraction process in identifying textual references to persons in a document. Other examples are organization suffixes and currency prefixes. The KB of KIM is pre-populated with entities of general importance. These entities aid the Information Extraction process to perform well on inter-domain Web content. The semantic annotation platform comprises the KIM Ontology, the KB, the KIM Server and front-ends to browse and search the KB. The KIM API, which is part of the server, provides semantic annotation, indexing and retrieval services as well as persistence functionality. Lucene[7], a publicly available Information Retrieval engine, has been customized to index documents by entity types. KIM performs ranking by using statistics about the frequency of occurrence of a specific Named Entity (NE) such as "Bill Clinton" in latest news articles. Therefore, ranking is based on the assumption that NEs occurring in more news articles will be more likely of interest to a user. The annotation platform KIM has been incorporated into HS^3. It is possible to use the KIM Annotation Pipeline via Java RMI.

NAGA

Kasneci et al. [61] developed a Semantic Search System named NAGA. NAGA, similar to ESTER, uses the YAGO ontology, but is not limited to knowledge extracted from Wikipedia. NAGA's KB contains data derived from a number of semi-structured and unstructured Web sources such as Wikipedia and the Internet Movie Database (IMDB). The system comprises a KB, Information Extraction tools, a query and a ranking unit as well as a user interface for casual and expert users. Furthermore an interface is provided that enables users to browse the KB by means of a hyperbolic visualization of the knowledge graph.

[7] http://lucene.apache.org

Any instance in the KB carries a reference to a WordNet[8] concept. The ontology consists of 3 kinds of concepts namely `Word`, `Individual` and `Class`. A `Word` is a textual representation that refers to an entity via a *means* relation. An `Individual` is a real world object and a group of similar `Individuals` is represented by a `Class`. The model of NAGA can be applied to any KB that uses this structure. NAGA's query language syntax and semantics have been derived to a great extend from SPARQL. NAGA uses a scoring model to perform ranking of the result entries. Every result entry is scored by aspects such as confidence, informativeness and compactness. The confidence score depicts how trustworthy the actual result entry is. Informativeness expresses how informative the result entry is with respect to the given query. The compactness defines the grade of compactness between two entities. Therefore the less nodes are in-between two entities the higher the compactness score. In contrast to NAGA the KB of HS^3 does not require a special structure. The annotation and indexing components of HS^3 can be customized to work with an arbitrary structure.

AVATAR

AVATAR is a semantic search engine that is based on a database approach [59]. The main intent of AVATAR is to explicitly model a user's intent encoded in a keyword query by using annotations. Like most semantic search engines AVATAR also uses Information Extraction techniques to extract entities from documents and creates annotations to reference textual representations of them. AVATAR uses the publicly available UIMA framework [43], which is a framework to create so-called annotators that are part of a document processing workflow. Documents are processed via this processing workflow and are annotated with references to entities in the KB. These annotators store annotations in a structured data store that has been realized as a thin layer on top of a commercial Database Management System (DBMS). The process of transforming keywords into one or more precise queries is named *keyword query interpretation*. This process transforms keyword queries submitted by users into structured queries that model a user's intent. The components used by AVATAR to accomplish this task are a Semantic Optimizer and a User Interface Engine. The Semantic Optimizer produces a semantically meaningful interpretation of a keyword query and interpretations that are ranked highest are forwarded to the User Interface Engine. The User Interface Engine's purpose is to present and display interpretations as well as result documents to the user. Similar to AVATAR, HS^3 also uses a processing workflow to annotate documents. However, HS^3 also uses the data available in the KB to automatically fetch suitable documents from the World

[8]http://wordnet.princeton.edu/

Wide Web. Furthermore, HS^3 can make use of annotation components from the UIMA and GATE [25] framework.

3.2.2 Hybrid search approaches

Several other systems that specifically focus on searching structured and unstructured data have been presented. A hybrid search approach, that makes use of structured and unstructured data has been proposed by Bhagdev et al. [10]. The reference implementation K-Search uses unstructured data as complement whenever no suitable structured data is available. Bhagdev et al. showed that hybrid search outperforms the separate keyword-based and semantic-based search approaches. User studies have been conducted which indicated that the hybrid search approach is favored by users over the individual keyword-based and concept-based approaches. However, the findings in [10] are based on the assumption that the semantic search part of the hybrid search delivers results of high precision. Otherwise, the approach of using keyword-based search whenever no suitable semantic data is available might not be the most effective strategy. HS^3 also uses unstructured data as complement whenever no suitable structured data is available. Structured data is preferred over unstructured data, because a higher search result precision can be accomplished.

Bikakis et al. [12] proposed GoNTogle, a framework for document annotation and retrieval, which is capable to overcome restrictions of keyword-based and concepts-based search by using a hybrid search strategy. The framework uses the strength of the concept-based approach to compensate the weakness of keyword-based approach and vice versa. Hence, it uses the keyword-based approach to compensate partial or not existing annotations and the concept-based approach to deal with polysemy and synonymy. However, according to the description of the semantic search facility only concept annotations are considered during the semantic search process, but relations (properties) between concepts are ignored. Therefore, in case a user searches for two or more concepts that are related via properties (and not all of them have corresponding annotations in documents) they would be missing in the result. Bikakis et al. have also shown that the hybrid search approach outperforms the individual keyword-based and concept-based search strategies in terms of precision and recall. In contrast, HS^3's annotation and search process make use of relations (properties) between concepts to provide more accurate search results.

Giunchiglia et al. [53] proposed Concept Search, which shares similarities with the work of [10], but does not use separate indexes for concept-based and keyword-based search. To integrate the two search approaches, Concept Search substitutes words from the keyword-based approach with suitable concepts. Hence, it uses the data structures

System Name	Features	Characteristics
Vector Space Model-based Semantic Search System [21]	* Can cope with missing information in the KB by falling back to a keyword-based search approach * An arbitrary domain ontology can be used * System can use inference techniques to extract implicit knowledge from the KB	* Uses a strict Boolean model to perform searches * User can only use RDQL to interact with the system * Response times of queries can be up to 30 seconds
ESTER [7]	* Response times are fractions of a second * Interactive User Interface * Can operate with big data sets consisting of millions of triples and documents	* Only application with semi-structured data from Wikipedia demonstrated * Not clear how results are ranked
KIM [77]	* Builds upon freely available components such as GATE and SESAME * Uses established standards such as RDF, RDFS and OWL Lite * Comes with a pre-populated KB holding basic entities and facts using the KIM Ontology (KIMO) * Provides very good results on NER tasks with the KIM Ontology * System is available for research from the developers' Website * Provides API for keyword-based and concept-based search	* Tightly coupled to the KIM Ontology and KB * No complex semantic queries supported out of the box * No combined search available out of the box * Ranking is based on entity popularity timelines analysis
NAGA [61]	* Uses an advanced scoring model for ranking by incorporating information about confidence, informativeness and compactness * Demonstration system available	* Only application with semi-structured data from Wikipedia demonstrated * Demonstration system presented on Website has high response times even for simple queries
AVATAR [59]	* Uses the freely available UMIA framework for IE * System is specialized in the identification of entities in user queries to derive user's intent	* Only entity class recognition is performed but no annotation to specific entity instances * Demonstrated application of the system is bound to a corpus of email messages

Figure 3.1: Feature and drawback summary of described Semantic Search Systems

and models of the keyword-based search, but substitutes the syntactic keyword matching with the semantic matching of concepts during the retrieval process. Experiments conducted with the TREC ad-hoc document collection (disks 4 and 5 minus the Congressional Record documents) indicated that Concept Search performs better than the pure keyword-based search. Hence, the hybrid approach of using semantic information in addition to syntactic information performs better or at least as good as the keyword-based approach if no additional semantic information is available.

Fernandez et al. [42] presented a system that consists of a semantic unit, which uses a natural language query to retrieve semantic data from a KB and a document retriever that provides a ranked list of documents annotated with semantic data to the user. Similar to other systems that use semantic information for the keyword-based search, the hybrid approach performs better compared to the pure keyword-based search approach. Precision and recall results of HS^3 also indicate that the hybrid search approach is superior to the keyword-based approach.

3.2.3 Interaction with Semantic Search Systems

Kaufmann and Bernstein [63] evaluated the usability of natural language interfaces that are used to interact with KBs. Even tough the main advantage of natural language interfaces is that users are familiar with this type of input, the evaluation has shown that users need to know what is possible to ask beforehand to effectively make use of the system. The knowledge of possible input phrases and the terminology is crucial, because these systems rely on specific query formulation. Kaufmann and Bernstein introduced the Habitability Hypothesis which proposes that query interfaces should not overly control the user with an excessively formalistic language but rather impose some structure on the casual user to guide her during the query formulation process. This hypothesis perfectly resembles our intention behind the interactive ontology-aware query formulation mechanism presented in Section 3.5.

Zenz et al. present QUICK [100], a system for incremental query construction that assists the user in constructing semantic queries from keywords. However, due to the incremental approach of QUICK, users may need to issue several queries to the system until they are able to formulate their actual intent into a final semantic query. Furthermore, users still need a certain understanding of how to construct semantic queries which makes the interface not intuitive to the average Internet user. It would have been interesting to see if users without prior knowledge of the ontology would have been able to formulate queries that effectively express their intent, which was not part of the evaluation. As QUICK transforms semantic queries into SPARQL queries these may grow in complexity and yield very high response times when issued against a Triple Store. This is especially true as filter operators in combination with regular expressions are used for keyword matches. Even though a join order optimization, has been proposed to deal with this issue, no query performance figures are presented. To evaluate whether average Internet users are able to formulate queries without prior knowledge of the ontology with the interactive ontology-aware input mechanism that was implemented in HS^3, we conducted a usability study and present the results in Section 3.7.

Schreiber et al. [83] present an interesting system named the MultimediaN E-Culture demonstrator, which supports keyword-based queries on annotated cultural-heritage document and image collections. The MultimediaN E-Culture demonstrator uses a JavaScript, HTML/XML and CSS based interactive user interface that offers instant concept and instance suggestions to the user's input. The user may choose among one of the suggested concepts or instances and issue a query to the system. However, it is not possible to issue complex queries that involve multiple concepts or instances, their relations and keywords. We implemented a similar concept and instance suggestion mechanism for HS^3 by using GWT[9].

3.3 Why a Hybrid Semantic Search System?

To demonstrate the advantages of a Hybrid Semantic Search System consider the following scenario from the e-Tourism domain. A tourist who wants to travel to Tyrol with her son is searching for a suitable accommodation, preferably a child-friendly guesthouse, which offers an indoor swimming pool and is located in a place where she can go canyoning. Additionally, she heard from a friend that Mayrhofen is a beautiful place in Tyrol and would prefer to have the guesthouse located there. Therefore, her query to a general-purpose search engine might look like "child-friendly guesthouse in-door swimming pool mayrhofen canyoning". The search engine will return documents that include these terms. In case there are documents that contain all terms they will be returned among the top documents for this search query. If there are no documents that contain all terms, documents that contain a subset of the query terms or are frequently referenced by documents deemed to be relevant will be among the top documents. This approach neglects that other documents might contain information about suitable accommodations as well, but do not explicitly contain these specific terms. Examples are documents that mention guesthouses by their actual name rather than the term guesthouse or documents that do not contain explicitly the term "mayrhofen" even though the guesthouses men-

[9]http://code.google.com/webtoolkit/

tioned in the documents are located in Mayrhofen. Another drawback of the traditional search approach is that the semantics in the query cannot be exploited to find similar accommodations that might also be of interest for a tourist. In the presented scenario the tourist might also be interested in hotels, youth hostels, bed and breakfast accommodations and other accommodations which are located near Mayrhofen or at least in Tyrol, where Mayrhofen is located, even though she did not explicitly state it.

Domain specific search engines such as Tiscover[10] have the advantage that they offer specific information that is maintained by domain experts. Furthermore, the user interfaces of domain specific search engines often assist the user in her search for information by offering drop down lists and check boxes to express her search intent accurately. For example, Tiscover has checklists to define accommodation types (e.g. guesthouse, hotel, bed and breakfast), the environment of an accommodation (e.g. near lake, near mountain, near ski-region) or its facilities (e.g. swimming pool, fitness room, sauna). However, domain specific search engines have the major drawback that they can only leverage information that is stored in their data storage, which comprises a tiny fraction of the information available by general-purpose search engines. In case of the given scenario the user would get no or only approximate results if the data storage holds no information about guesthouses in Mayrhofen that are child-friendly and offer an in-door swimming pool, even though there are accommodations that have this characteristics but mention them only on their Website.

Semantic Search Systems that rely entirely on their KB suffer from the same drawback as domain specific search engines. In case the requested information is not contained in the KB, they cannot leverage any complementing data source to acquire it. Semantic Search Systems that use complementing data sources such as annotated document corpora for the search process are sparse and use the complementing data source only in case the information is not contained in their KB, even though the complementing data source holds valuable additional information that can be used to perform a more specific search. Considering the given scenario, such a system will return information about appropriate accommodations if it is contained in the KB or in the complementing data source. However, in case that part of the information is contained in the KB and part of it in the complementing data source, it will fail to return the corresponding results among the top ones because the sources are not searched conjointly.

Considering the given scenario, an ideal Semantic Search System should be able to deduce that the term “guesthouse” represents an accommodation, indoor swimming pool is a facility of an accommodation, the term “mayrhofen” refers to the city Mayrhofen that

[10]http://www.tiscover.com/

is located in Tyrol and canyoning is a leisure activity. With this information, a system with an appropriate ontology, KB and annotated document corpus that was created from multiple sources, would be able to broaden the search to get the user similar results even though there might be no guesthouse with an indoor pool, located in Mayrhofen with the possibility to go canyoning. Furthermore, the system may use information that is not part of the KB or even the ontology, but encoded in documents of the corpus. For example if a user submits a query for accommodations in Mayrhofen that are child-friendly, the system is able to retrieve accommodations located in Mayrhofen that are child-friendly even though the concept child-friendly is not included in the ontology. This could be accomplished by using a combined search on the document corpus and the KB. A document that mentions any accommodation such as Elisabeth Hotel in Mayrhofen and contains the term “child-friendly” would be returned in the result, even though the terms “accommodation” and “mayrhofen” are not contained in the document, but the KB holds the information that Elisabeth Hotel is an accommodation and located in Mayrhofen. To the best of our knowledge there is currently no Semantic Search System that leverages the information richness of the World Wide Web in combination with semantic data available in a KB to automatically fetch relevant complementing data and use a combined search approach to provide users with relevant information for a certain domain.

3.3.1 Characteristics that hinder the use case realization

During the literature review and the evaluation of Semantic Search Systems we identified characteristics of current Semantic Search Systems that possibly hinder the realization of the defined use case which are summarized in the following. Even though for most semantic search systems it is stated that they can be used with an arbitrary ontology all of them have only been tested with one specific domain ontology. KIM is tightly coupled to the KIM Ontology and its KB. KIM’s main focus is laid upon the annotation of documents with entities of general importance such as continents, global regions, countries with their capitals, cities, persons and the biggest world organizations. KIM achieves good results in recognizing entities of general importance. The API of KIM offers the combination of keyword-based and concept-based search, but there is no predefined search service or ranking mechanism that uses this functionality. ESTER and NAGA are tightly coupled to the YAGO ontology and the KB that was generated from the Wikipedia corpus. To the best of our knowledge ESTER, the Semantic Search System proposed by Castells et al. [21], the system presented by Bhagdev et al. [10] and the system presented by Giunchiglia et al. [53] are the only ones that provide a search service that can use a combination of keyword-based and concept-based search out of the box. However, the system of Castells executes the different search techniques separately and combines the result with the CombSUM [46] strategy. The other systems suffer from issues such as the separation of keyword-based and concept-based input in the user interface, non-intuitive and overcrowded user interfaces or search inflexibilities due to the separation of the keyword-based and concept-based index. Techniques to exploit the information available in a document corpus in combination with the semantic information of a KB for conducting hybrid searches are still unexplored to a great extent. None of the presented Semantic Search Systems provides an end-to-end process that includes all steps that are needed to automatically build a document corpus from Web sources, a KB and a combined index to conduct hybrid searches upon. The majority of Semantic Search Systems use custom or complex input mechanisms, which are simply not applicable for the average Internet user. Users are used to keyword-based input mechanisms where keywords or phrases are typed into a text field such as the input mechanisms of major search engines such as Google, Yahoo or Bing.

3.4 Bridging structured and unstructured data

In this section we start with a formal definition of hybrid semantic search and put it into practice by introducing the Hybrid Semantic Search System (HS^3). The following formal definitions in Description Language (DL) follow the definition in [5]. Let $\mathcal{K}$ be a Knowledge Base defined by $\mathcal{K}$ = ($\mathcal{T}$, $\mathcal{A}$), where $\mathcal{T}$ is the TBox and $\mathcal{A}$ is the ABox. The TBox contains intensional knowledge and the ABox contains extensional knowledge. Intensional knowledge is built by declarations that describe general properties of concepts whereas extensional knowledge is knowledge that is specific to the individuals of the domain. Hence, the TBox holds all concept and role definitions whereas the ABox holds instances (individuals) of the concepts and roles (binary relations between instances). Consider the TBox presented in 3.1, holding the concept definition of the concept `Guesthouse`, which declares that the concept `Guesthouse` is a combination of the concept `Hotel` and the concept `Restaurant`.

$$\texttt{Guesthouse} \equiv \texttt{Hotel} \sqcap \texttt{Restaurant} \tag{3.1}$$

The ABox presented in 3.2 holds an instance of the concept definition of the TBox presented in 3.1. The instance “Pendl” is an instance of the concept `Hotel` and the concept `Restaurant`. According to the concept description of the TBox it can be asserted that the instance “Pendl” is also an instance of the concept `GuestHouse`.

$$\texttt{Hotel} \sqcap \texttt{Restaurant(Pendl)} \tag{3.2}$$

We concentrate on $\mathcal{SHIF}^{(D)}$, a DL variation, because our KB uses OWL Lite[11] which conforms to the $\mathcal{SHIF}^{(D)}$ Description Language. In OWL a DL concept is known as class and a DL role is known as property. We use concept descriptions to represent queries that describe the objects of interest in a domain Δ^I. Hence, the interpretation of the query in Δ^I are those instances that satisfy the concept description. In case of HS^3, objects of interest are documents and instances of domain concepts. The interpretation is denoted as $\mathcal{I}$, consisting of the non-empty set Δ^I, representing the domain of the interpretation, and an interpretation function $\cdot^\mathcal{I}$ which assigns every concept C a set $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$, every keyword K a set $K^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and every role R a binary relation $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. Where $K^\mathcal{I}$ are the keyword instances in $\Delta^\mathcal{I}$. A keyword may be considered as a concept which can have multiple keyword instances. Its instances are the representations of the keyword in documents. For example there is only one keyword represented via the term “quiet”, but there are multiple instances of this keyword contained in multiple documents. Hence, $K^\mathcal{I} \subseteq \Delta^\mathcal{I}$ is a set of instances of the keyword K, which is a subset of all keyword instances in the domain $\Delta^\mathcal{I}$ that are distributed over multiple documents. The following inductive definitions from Baader et al. [5] are used to extend the interpretation function to concept descriptions :

$$\begin{aligned}
\top^\mathcal{I} &= \Delta^\mathcal{I} \\
\bot^\mathcal{I} &= \emptyset^\mathcal{I} \\
(\neg A)^\mathcal{I} &= \Delta^\mathcal{I} \backslash A^\mathcal{I} \\
(C \sqcap D)^\mathcal{I} &= C^\mathcal{I} \cap D^\mathcal{I} \\
(C \sqcup D)^\mathcal{I} &= C^\mathcal{I} \cup D^\mathcal{I} \\
(\exists\, R.C)^\mathcal{I} &= \{a \in \Delta^I \mid \exists\, b.(a,b) \in R^\mathcal{I} \wedge b \in C^\mathcal{I}\} \\
(\forall\, R.C)^\mathcal{I} &= \{a \in \Delta^I \mid \forall\, b.(a,b) \in R^\mathcal{I} \rightarrow b \in C^\mathcal{I}\}
\end{aligned} \tag{3.3}$$

The inductive definitions are explained in the following. The interpretation of the universal concept ($\top^\mathcal{I}$) equals the domain of the interpretation. Therefore, all individuals of the domain are included in the set of individuals. The interpretation of the bottom concept ($\bot^\mathcal{I}$) equals the empty set. Therefore, no individual that is part of the domain is included in the set. The interpretation of the atomic negation ($(\neg A)^\mathcal{I}$) equals

[11] To be more precise, the Backend of the KB uses OWL Horst, a subset of OWL Lite, because OWL Horst is OWLIM’s most expressive DL at the moment

the domain of the interpretation minus all individuals of concept A. The interpretation of the intersection ($(C \sqcap D)^{\mathcal{I}}$) equals the set of individuals that are instances of concept C and concept D. The interpretation of the union ($(C \sqcup D)^{\mathcal{I}}$) equals the set of individuals that are instances of concept C or concept D. The interpretation of the full existential quantification ($(\exists\, R.C)^{\mathcal{I}}$) equals the set of individuals that have a relation to individuals, where at least one of the individuals has a specific concept type. An example would be a set of `Accommodation` instances that have a relation *hasFacility* and at least one of the related instances is of type `Restaurant`. This can be defined as `GuestHouse` $\equiv$ `Accommodation` $\sqcap \exists\, hasFacility.$`Restaurant`. Therefore, only those accommodations are deemed as guesthouses which have at least one facility which is a restaurant. The interpretation of the value restriction ($(\forall\, R.C)^{\mathcal{I}}$) equals the set of individuals that have a relation to individuals which have all the same concept type. An example would be a set of `Accommodations` instances that have a relation *hasFacility* and all related instances are of concept type `SportFacility`. This can be defined as `SportHotel` $\equiv$ `Accommodation` $\sqcap \forall\, hasFacility.$`SportFacility`. Therefore, only those accommodations are deemed as sport hotels that have nothing but sport facilities. (Note that the $\forall$ restriction might not be of particular relevance for the presented tourism domain and that the previous example was just given for completeness reasons).

We want HS^3 to process keyword queries KQ that consist solely of keywords, concept queries CQ which may contain concepts, instances of concepts and roles and hybrid queries HQ which may contain concepts, instances, keywords, keywords in the context of concepts and roles. In the following an example per query type from the tourism domain is given :

$$KQ_1 = (\{BIG\} \sqcap \texttt{Keyword}) \sqcup (\{LAKE\} \sqcap \texttt{Keyword}) \tag{3.4}$$

$$CQ_1 = (\texttt{Hotel} \sqcap \exists\, locatedIn.(\{SALZBURG\} \sqcap BroadLocation)) \tag{3.5}$$

$$\begin{aligned} HQ_1 = \; & (\texttt{Hotel} \sqcap \exists\, offers.\texttt{Steambath}) \sqcup \\ & (\texttt{Hotel} \sqcap \exists\, locatedIn.(\{MAYRHOFEN\} \sqcap BroadLocation) \sqcup \\ & (\texttt{Hotel} \sqcap \exists\, describedAs.(\{CHILD-FRIENDLY\} \sqcap Keyword)) \end{aligned} \tag{3.6}$$

Hence, the search queries are concept descriptions in the form of a disjunction ($\sqcup$) of conjunctions ($\sqcap$). The keyword query KQ_1 is a composition of the two keywords “big” and “lake”. The concept query CQ_1 is a composition of the concept `Hotel`, the relation *locatedIn* and the instance Salzburg which is of type `BroadLocation`.

The hybrid query HQ_1 is a composition of concept `Hotel`, the relation *offers*, the concept `Steambath`, the relation *locatedIn*, the instance Mayrhofen which is of type `BroadLocation`, the relation *describedAs* and the keyword "child-friendly". The keyword "child-friendly" is used to describe the concept `Hotel` in more detail.

The structure of the index is determined by the type of queries that should be supported by the system. To answer queries of type KQ a simple inverted index can be used. To answer queries of type CQ, without considering roles between concepts and instances an inverted index that stores concept and instances descriptions with document IDs can be used. However, in case of queries of type CQ which include roles between concepts and queries of type HQ a more advanced index structure is needed to facilitate an efficient retrieval of relevant information. To handle queries of type CQ and HQ we created the combined index which is based on SIREn [29]. To facilitate an efficient search over the combined index we create a document's structure as an ordered set d as follows :

$$\begin{aligned}
d = \{K' \cup IG \cup CC \mid K' \subseteq K,& \\
AI = \{ai \mid ai \in C^{\mathcal{I}} \cup \upsilon(ai) \in K'\},& \\
IG = \{ig \mid \exists\, ai \in AI\colon (ai, sc, s, r, oc, o) \wedge& \\
sc, oc \in C, r \in R^{\mathcal{I}}, s, o \in C^{\mathcal{I}},& \\
CC = \{cc \mid \exists\, ai \in AI\colon (ai, sc, s, k) \wedge& \\
sc \in C, s \in C^{\mathcal{I}}, k \in K\}&
\end{aligned} \tag{3.7}$$

Where K' is a set of keywords, IG is a set of *Index Graphs*, CC a set of concept contexts, AI is the set of instances that have annotations in d, and υ is a function that returns the textual description of an instance. The set of keywords K' holds all keywords that are contained in the document. The set of *Index Graphs* IG holds *Index Graphs* in the form of (ai, sc, s, r, oc, o) tuples. One *Index Graph* may have an arbitrary number of these tuples. One tuple encodes the relation between an annotated instance ai, a subject s, a role r and an object o. Furthermore, the concept type of the subject sc and the concept type of the object oc are contained in a tuple. URIs are used to define sc, s, r, oc and o. An example of an *Index Graphs* represented as tuples is depicted in the last box of Figure 3.2.

For the formal definition of an *Index Graph* let $O = (V, E)$ be the TBox's representation as graph, where V are the vertexes corresponding to concepts C of the TBox and E be the edges corresponding to the roles R of the TBox. In the *Index Graph* example labeled with the number 1 in Figure 3.2 the vertexes V are depicted as oval shapes

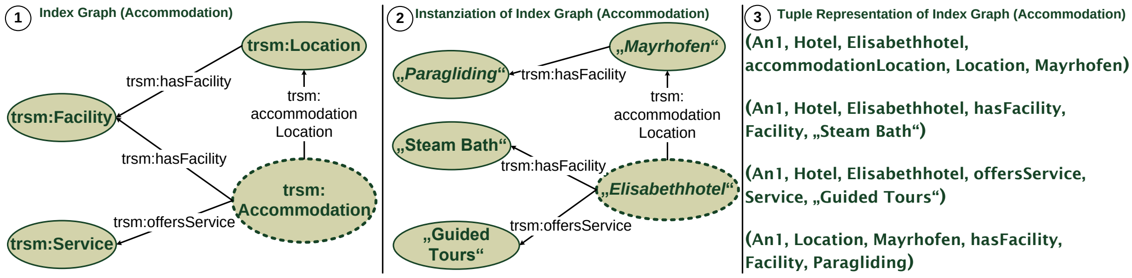

Figure 3.2: Index Graph Definition, Index Graph Instance and Tuple Representation

that represent concepts such as `Accommodation` and `Location` and the edges E are depicted as lines that represent properties such as `accommodationLocation` which connect these concepts.

Furthermore let $O^{\mathcal{I}}$ be the interpretation of O. An *Index Graph* is a star-shaped subgraph $IG' \subset O^{\mathcal{I}}$, which has a root vertex $RV \in V^{\mathcal{I}}$, that has a representation in the document's annotation set AI, which holds instances that have annotations. In the *Index Graph* example interpretation labeled with the number 2 in Figure 3.2 the root vertex RV is represented by the instance "Elisabethhotel". Finally, marked with the number 3 in Figure 3.2, the corresponding tuples for annotation *An1* (which is part of the annotation set AI) of instance "Elisabethhotel" in document d are listed.

The purpose of an *Index Graph* is to encode relevant information for a specific instance that is mentioned (annotated) within a document. It is a subgraph of the instance's complete graph in the KB that is relevant for the retrieval process. Every concept in C can be equipped with an *Index Graph*. A knowledge engineer or domain expert defines the *Index Graph* for a concept, because she knows best what concepts and properties are relevant to users and need to be indexed. To conduct searches upon the star-shaped *Index Graph* it is transformed into a set of tuples in the form (ai, sc, s, r, oc, o) that is related to d, as specified in Equation (3.7), and indexed.

The concept context set CC contains a set of tuples (ai, sc, s, k) as specified in Equation (3.7). Every tuple relates a keyword k from the surrounding context of the annotated instance ai to a concept sc or an instance s. The set CC is indexed for every document d to facilitate hybrid searches including keywords and concepts. Formally, the query answer $A(HQ)$, where $KQ \subset HQ$ and $CQ \subset HQ$, is defined as an unordered set :

$$
\begin{aligned}
A(HQ) = \{d|(KQ \subset HQ \wedge KQ \in K' \in d) \vee \\
(\rho(HQ) \in IG \in d) \vee \\
(\eta(HQ) \in CC \in d)\}
\end{aligned} \tag{3.8}
$$

Where ρ is a function that transforms the concept and role part of the query into a set of tuple representations (ai, sc, s, r, oc, o) and η is a function that transforms the keywords related to concepts and instances in HQ to a set of tuple representations (ai, sc, s, k) as specified in Equation (3.7). Therefore, according to $A(HQ)$ documents are included in the unordered set where at least one of the following criteria applies:

1. One or more keywords of the query are part of d's keywords (K')
2. One or more concepts, instances or combinations of concepts, instances and properties of the query are part of d's *Index Graph* (IG)
3. One or more keywords of the query are contained in the context of any concepts and instances that are part of d's Concept Context (CC)

The variable ai is never bound during query generation, because ai represents an annotated instance, which is not known at the query's generation time. The variables sc, s, r, oc and o are either bound or unbound depending on the actual query. The unordered set is transformed to an ordered set via the ranking algorithm that assigns weights to every tuple which will be discussed in more detail in Section 3.4.1. Consider the previously stated user query "*child-friendly hotel that offers steam bath and is located in mayrhofen*", which has been formally defined as HQ_1. The tuples generated by $\rho(HQ_1)$ and $\eta(HQ_1)$ would look as follows :

$$
\begin{aligned}
&\rho(HQ_1) \rightarrow \{ \\
&(?, Hotel, ?, locatedIn, Location, \text{“Mayrhofen”}) \\
&(?, Hotel, ?, hasFacility, Facility, SteamBath)\}
\end{aligned} \tag{3.9}
$$

$$
\begin{aligned}
&\eta(HQ_1) \rightarrow \{ \\
&(?, Hotel, describedAs, \text{“child-friendly”})\}
\end{aligned} \tag{3.10}
$$

There are two unbound variables represented by *?* in every listed tuple, because the user has specified the concept `Hotel` in the user query graph, but not a specific instance of `Hotel` and the variable ai is always unbound in a query. Hence, to search upon

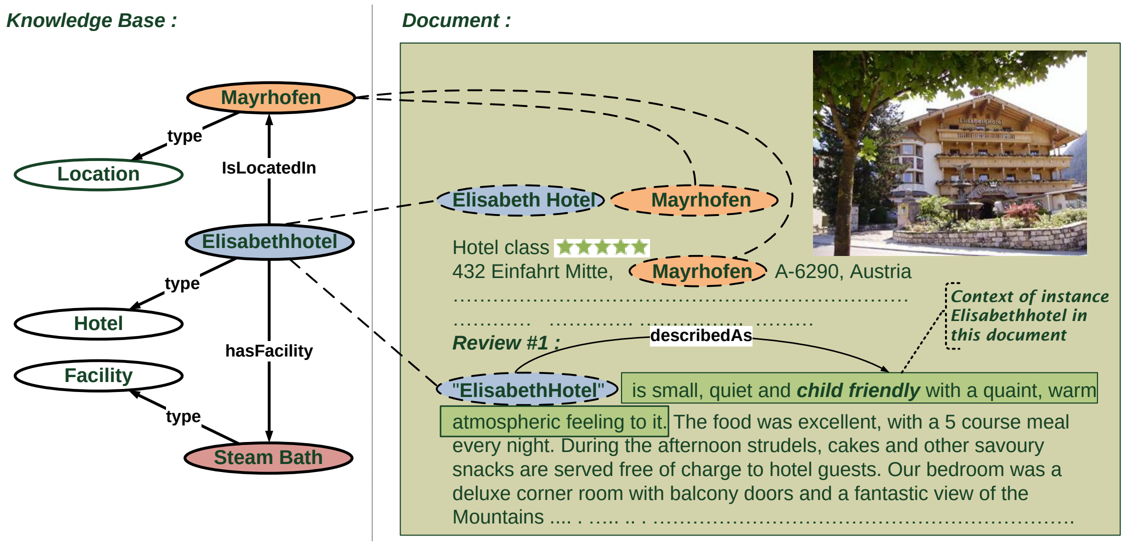

Figure 3.3: Supplementation of knowledge in KB and World Wide Web

the index, HQ_1 needs to be transformed into tuples conforming to (ai, sc, s, r, oc, o) and (ai, sc, s, k). This is accomplished by transforming the user query into a tuple representation which is basically a disjunction of conjunctions (the tuples). This representations is matched against documents d stored in the combined index.

A ranked list of documents is created based on the exactness of the match of the user query graph's tuples and the tuples of document d. By using this combined index structure, knowledge from the KB and data fetched from the World Wide Web can be combined to facilitate an efficient search by exploiting the clear structure of the ontology and the information richness of the World Wide Web.

To demonstrate the advantage of the combined index structure, consider the instance of an *Index Graph* and the document depicted in Figure 3.3. This document would be returned by query HQ_1 and ranked among the top documents in the result set, even though the document does not mention anything about a steam bath, nor does the KB hold any information that Elisabethhotel is a child-friendly hotel. However, the KB holds the information that Elisabethhotel is equipped with a steam bath and that it is located in Mayrhofen, and the document holds the information that Elisabethhotel is child-friendly. Hence, the document is returned because the combined index amalgamates this information via index graphs and concept contexts. Therefore, the user gets what she was looking for. In case separate indexes would have been used, the document would not have been ranked amongst the top documents.

3.4.1 The Hybrid Semantic Search System HS^3

In this section we present HS^3 which puts the combined index structure into practice. Suitable techniques to realize the system's components have been evaluated and are discussed in this section. We describe the system by first giving an introduction to the system's components and their collaboration, followed by a detailed discussion on how the system uses these components to meet the goals defined in Section 3.3 and show how the example use case of the tourism domain is supported by the system.

The description of the system's architecture and functionality has been submitted to [52]. The architecture of the system is depicted in Figure 3.4. The system consists of 5 main components, namely the Persistence components, the Data Fetching components, the Annotation & Indexing components, the Search & Ranking services and the Interface Layer. Every component may have additional external functions that provide it with data. The Persistence components are provided with data from the Ontology Enricher and the Transformation Engine. The Ontology Enricher adds textual representations of concepts to the ontology and the Transformation Engine transforms custom data structures to RDF (or OWL) and stores it in the KB. A textual representation of a concept contains one or more terms that describe a concept. For example terms such as "canyoning" and "rafting" are textual representations of the concept `Canyoning`. The Transformation Engine is an integral part of the system, because it is used to create the initial dataset of the KB and can deal with different custom data structures. We share the opinion of Popov et al. [77] that in cases where it is possible to engineer basic knowledge in advance with reasonable effort, it should be preferred over extracting and inferring basic knowledge with uncertain methods. That is the main motivation for creating a Transformation Engine that is capable of transforming basic knowledge stored in custom data formats to RDF (or OWL). The Persistence components are the ontology, the KB and the Document Store. In contrast to other work in which the terms KB and ontology are used interchangeably, we explicitly distinguish between the KB and the ontology. The ontology defines concepts, their properties and relations and the KB includes instances of these concepts and their properties. The ontology therefore only holds a couple of entities (concepts) whereas the KB may contain millions of entities (instances of concepts).

The Document Store holds copies of fetched documents and the respective metadata. Annotations of documents are stored in the Document Store and can be modified or extended by annotators or indexers. The work queues are managed by so-called Work Queue Managers which distribute work to the components of the system. Any component exposes services that can be used by other components. The Data Fetching components use services of the Persistence components. Data Fetchers and Metadata Fetchers read

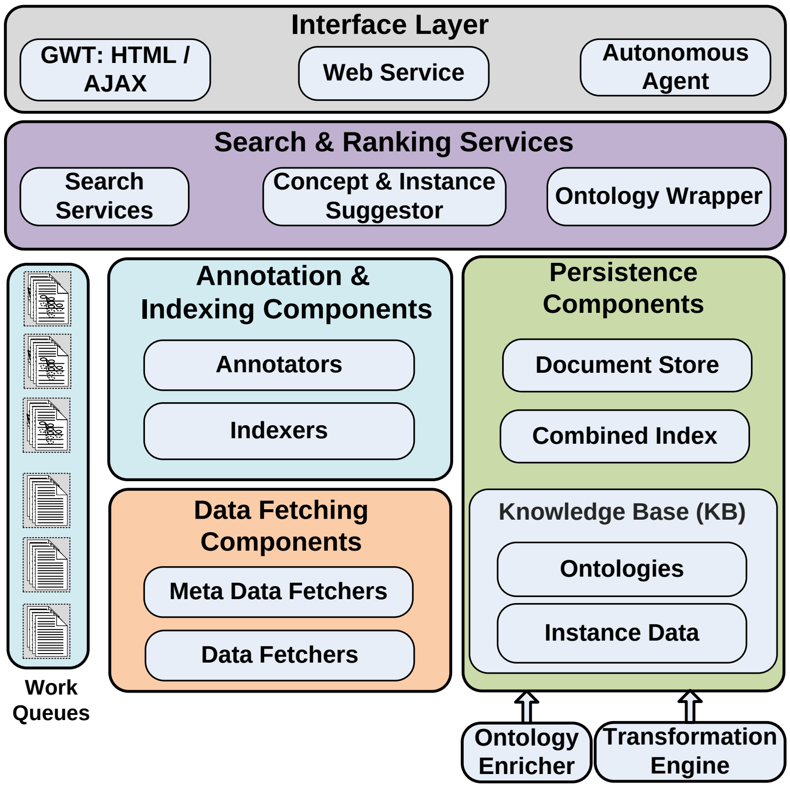

Figure 3.4: HS^3 System Architecture

data from the KB and write fetched document data to the Document Store. The Metadata Fetcher is used to fetch meta-data for instances of concepts that are stored in the KB. This information is subsequently used by Data Fetchers to fetch data such as HTML documents and store them in the Document Store for further processing.

The Annotation & Indexing components use data that was fetched and filtered by the Data Fetching components. Annotators use the ontologies and the KB to annotate documents in the Document Store. The Annotators use so-called Annotation Pipelines that adhere to a specific structure and can have multiple processing resources that operate on the data to identify concepts and instances and create annotations. These processing resources may either be local or remote. It is possible to use an external annotation service from within an Annotation Pipeline and process the result. The Annotation Pipeline includes a mandatory final processing step where the data is transformed into a structure that can be used by the Indexers. It is possible to include custom Annotation Pipelines. Indexers operate on the ontology, KB and the Document Store which holds the annotated documents. The system includes a Semantic Indexer, but custom Indexers can be added to the system as well. Custom Indexers just need to implement a pre-defined interface.

The Semantic Indexer creates a combined index that consists of a full text index and a concept index that holds concepts, instances and their relations.

The system is service-oriented and multi-threaded to support parallelism and facilitate scalability. Services can be grouped into system services and consumer services. System services are solely communicating with other system services. Consumer services are accessed from other systems via Web Services or by end users via a graphical user interface. Additional instances of a service can be instantiated if the workload increases. The Search & Ranking Services hold consumer services such as the Search Services, Instance Suggestor and Ontology Wrapper, which are accessed via the Interface Layer. The system includes a default search service, namely the Semantic Search Service, but it can be extended with any custom implementation of a search service as long as it implements the pre-defined interfaces. The Instance Suggestor service can be used to get all instances of the KB that match a certain textual representation. The Ontology Wrapper is a service to access the ontologies managed by the system. Any service is part of a service pool holding multiple instances of it. A new service is instantiated if the workload increases. The Interface Layer can make use of any consumer service. It offers three different types of interfaces; a GWT[12] based Web User Interface, a Web Service Interface that enables access to the Semantic Search Service and a communication facility for autonomous software agents, which is realized as Web Service as well.

In the following we present the mechanisms of the Hybrid Semantic Search System that are used to automatically generate a KB holding basic domain knowledge, to build a Document Store including domain relevant documents from Web resources, to annotate them with domain knowledge and to create a combined index that is used to enable hybrid searches on the concept-based KB and keyword-based Document Store. The system was designed to be independent of a specific domain, hence ontology.

Persistence Components

To address the requirement that the HS^3 is capable of dealing with arbitrary ontologies, we had to ensure that the Persistence components are kept general and can be extended by custom plug-ins that handle specifics of certain ontologies. The ontology defines the domain upon which HS^3 operates and the KB holds knowledge that conforms to the structure defined by the ontology. Figure 3.5 depicts the Persistence components, the Ontology Enricher and the Transformation Engine. The Persistence components contain the KB, the ontologies and the Document Store.

[12]http://code.google.com/webtoolkit/

In order to create a KB an appropriate storage facility is required. Widely used storage systems are Relational Database Management Systems (RDBMS) and Triple Stores. NAGA uses a commercial RDBMS and KIM the open source Triple Store SESAME [17]. Our Hybrid Semantic Search System uses a Triple Store for its KB, because a Triple Store is more flexible than an RDBMS. Still, a Triple Store has the disadvantage that it is not as scalable and fast as an RDBMS. The most popular and mature frameworks currently available are Jena [20] and SESAME. We chose SESAME over Jena, because it offers an abstraction layer named SAIL that separates the implementation of the storage and inference layer from the rest of the system. Furthermore, there are already a couple of high performance implementations for the storage and inference layer available. We evaluated three different SAIL implementations, namely the native Store provided by SESAME, BigData[13] and OWLIM [64] from OntoText. The disadvantage of SESAME's native store is that it only supports RDF(S) and offers no advanced inference capabilities. The SAIL implementation of BigData focuses on distributed storage, is highly scalable and looked promising. Unfortunately, by the time we started our evaluation only sparse documentation was available and we were not able to get the BigData storage up and running. However, the BigData documentation has been extended recently. Finally we evaluated SwiftOWLIM[14] which is a high-performance semantic repository implemented as SAIL. We decided to use SwiftOWLIM because we wanted to make use of the advanced ontology modeling constructs available in OWL and also wanted to add custom rules for inference. The possibility to use custom rules for inference can be seen as a prerequisite for using arbitrary ontologies, because custom rulesets can be used to exploit specific capabilities of certain ontologies. Furthermore, by using a back-end that uses an OWL dialect it is possible to perform native ontology mapping that can be used to combine multiple ontologies to assist the Information Extraction process [97].

In order to make use of the KB it needs to be filled with instances of ontology concepts, also referred to as entities. These instances need to be extracted and transformed to conform to the structure of the ontology. The Transformation Engine is capable of transforming custom data structures into RDF to populate the KB. The Transformation Engine is an external tool that offers an interface for custom plug-ins. The Transformation Engine uses these plug-ins to transform custom data structures to instances of ontology concepts. We developed three plug-ins, the RDBMS plug-in, the HTTP/Web Service plug-in and the File plug-in. The RDBMS plug-in is used to transform entities of a relational database into instances of ontology concepts. The File plug-in is used to read and transform data from plain text or XML files and the HTTP/Web Service plug-in

[13]http://www.systap.com/bigdata.htm
[14]http://www.ontotext.com/owlim/

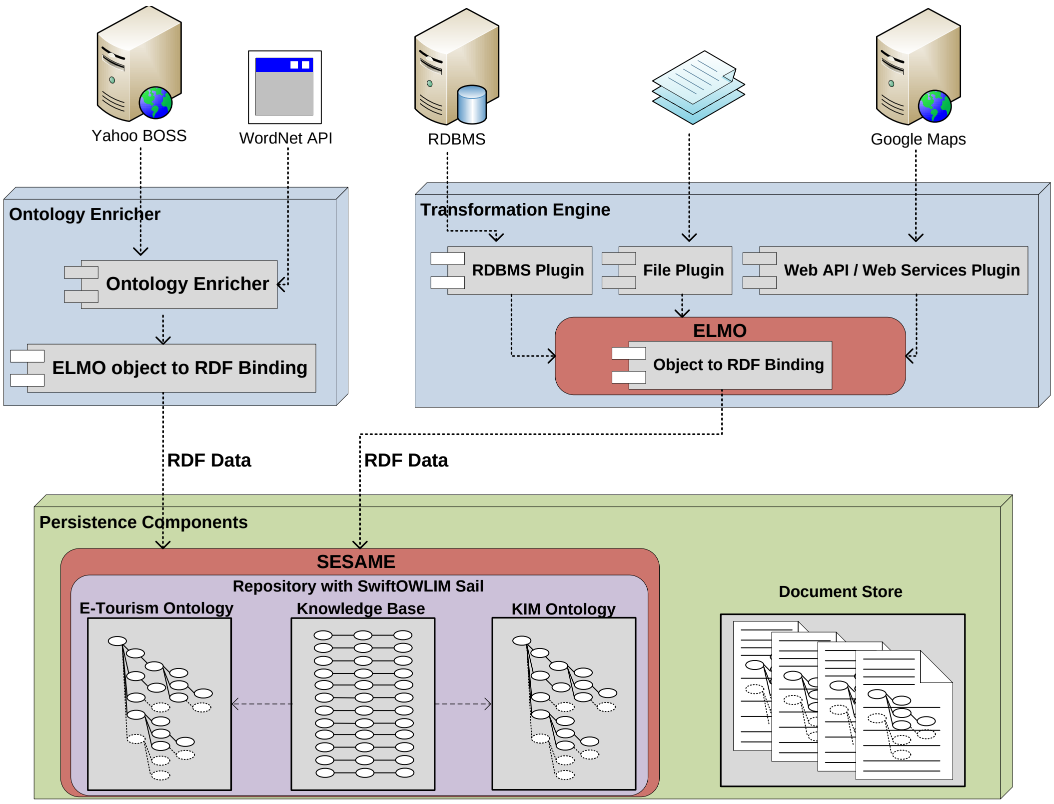

Figure 3.5: Ontology Enricher, Transformation Engine and Persistence Components

is used to transform data provided by HTTP based data protocols and Web Services. A prerequisite for the Transformation Engine is that all concepts of the ontology are available as objects via an Object to RDF binding framework. These objects are exposed to the plug-ins and can be used to create, read or modify instances in the KB. The currently most prominent object to RDF binding frameworks are Jenabean[15], ELMO[16] and So(m)mer (Semantic Object (Metadata) MappeR)[17]. Jenabean is meant to be used with the Jena Semantic Repository [20] and ELMO with the SESAME Semantic Repository [17]. So(m)mer is a rather simple framework that is not bound to any Semantic Repository and uses Java annotations to map Java classes (objects) to their RDF counterparts. We decided to use ELMO because it supports OWL out-of-the-box, uses SESAME and offers benefits such as automatic code generation based on the ontology.

The Ontology Enricher issues a query to the WordNet API for every concept in the ontology, creates a textual representation and stores it as part of the ontology in the `prefTextRepresentation` property. WordNet contains cognitive synonyms

[15] http://code.google.com/p/jenabean/
[16] http://www.openrdf.org/doc/elmo/1.5/
[17] http://java.net/projects/sommer/

(synsets), where each of them expresses a distinct concept. Every concept in the ontology has an optional label that represents the human readable name of the concept. This name is used to perform the lookup in WordNet. If an entry is found, all synonyms and semantically related expressions of this concept are retrieved. It is possible to configure the Ontology Enricher to only use those semantically related expressions that have the original term in their expression. For example, if the original term is hotel then only expressions such as “sport hotel”, “wellness hotel”, etc. will be used as textual representation of the `Hotel` concept. After the Ontology Enricher completed the WordNet lookups, it uses the Yahoo BOSS (Build your Own Search Service) API[18], to add additional textual representations to the ontology concepts. The Yahoo BOSS API offers access to the document index and meta-data of Yahoo. For a given search request Yahoo BOSS returns all key terms of every document that is part of the result list. The Ontology Enricher leverages this functionality by issuing a search request including the textual representations of a concept and adds the key terms included in the result as additional textual representations to the concepts. To reduce noise in the returned key terms the Ontology Enricher only uses the top ranked key terms and only if they are included in at least 70 percent of all key terms of all result documents for this concept. The idea is that those terms are synonyms or closely related to the query terms. It is possible to disable this source of information and only use the WordNet source to enrich concept descriptions. For example, when the textual representation of the concept `Hotel` is issued at Yahoo BOSS, several result entries are returned where the key terms contain similar textual representations such as “Spa Hotel” or “Luxury Hotel”. By using the noise reduction, “Spa Hotel” would only be used as additional textual representation if it is included in 70 percent of all results. It is possible to configure how many results shall be considered and to state the percentage of result entries that need to have the textual representation in their key terms. For example, it would be possible to state that the first 5 results have to be considered and that 40 percent must contain the textual representation in their key terms. Therefore, if two of the five results contain the textual representation “Spa Hotel” in their key terms, it would be added to the `Hotel` concept as additional textual representation. In ontologies with few concepts it is more accurate to annotate concepts by hand, but in case of a considerable amount of concepts the automated approach should be used and only those concepts where the automated approach produced poor results should be annotated manually.

[18]http://developer.yahoo.com/search/boss/

Data Fetching Components

The prerequisite for the Data Fetching components to perform their tasks is the existence of at least one ontology where every concept has a textual representation, and a KB with basic knowledge. The Data Fetching components and their connections to the Persistence components are depicted in Figure 3.6. The main components are the Metadata Fetcher and the Data Fetcher. The Metadata Fetcher operates on the data in the KB. Currently the Metadata Fetcher may use the Yahoo BOSS API or Microsoft Bing API[19] to retrieve meta-data for concepts and instances in the KB. The Yahoo BOSS API accepts text queries and optional parameters that define what information should be returned and what information must be filtered. Besides the URLs of the documents that match the query criteria, the service also returns available meta-data for every result entry. The URL of a document is used by the Data Fetchers to fetch the actual page content. The meta-data contains the type of the document such as HTML, PDF, text, etc., its key terms, its creation/modification date, a summary of the document's content, its language and embedded semantic markup such as microformats or common RDF. Microformats are represented in RDF as well and represent data such as geographic coordinates (Geo), personal profiles (FOAF) or reviews of products, services, businesses and events (hReview) to name a couple of the most popular.

The Metadata Fetcher can be equipped with plug-ins that are specialized in retrieving results for a specific concept. Therefore, it is possible to customize the data fetching routines according to the ontology that is used. For example, we implemented an Accommodation plug-in that was used to fetch related data off the Web for the e-Tourism reference implementation presented in Section 3.6. As input the plug-in uses instances of the `Accommodation` concept and their related `Location`, stored in the KB, to generate specific search queries that are issued to Yahoo BOSS. It uses attributes such as the preferred and alternative textual representation of the `Accommodation` and `Location` as well as the zip code of the `Location` to get as specific results as possible. To retrieve documents that contain permutations of the textual representation of `Accommodations` or `Locations`, the queries are expanded with permutations of the original textual representation to a certain extent before they are issued to Yahoo BOSS. Only permutations with a distance to the original textual representation that is below a predefined threshold are used for query expansion. For example, the permutation "Hotel Best Western Vienna" of the original textual representation "Best Western Hotel Vienna" will be part of the expanded query, but "Western Vienna Best Hotel" not. For every instance in the KB, data is fetched from Yahoo BOSS and written to an XML file which is

[19]http://msdn.microsoft.com/en-us/library/dd251056.aspx

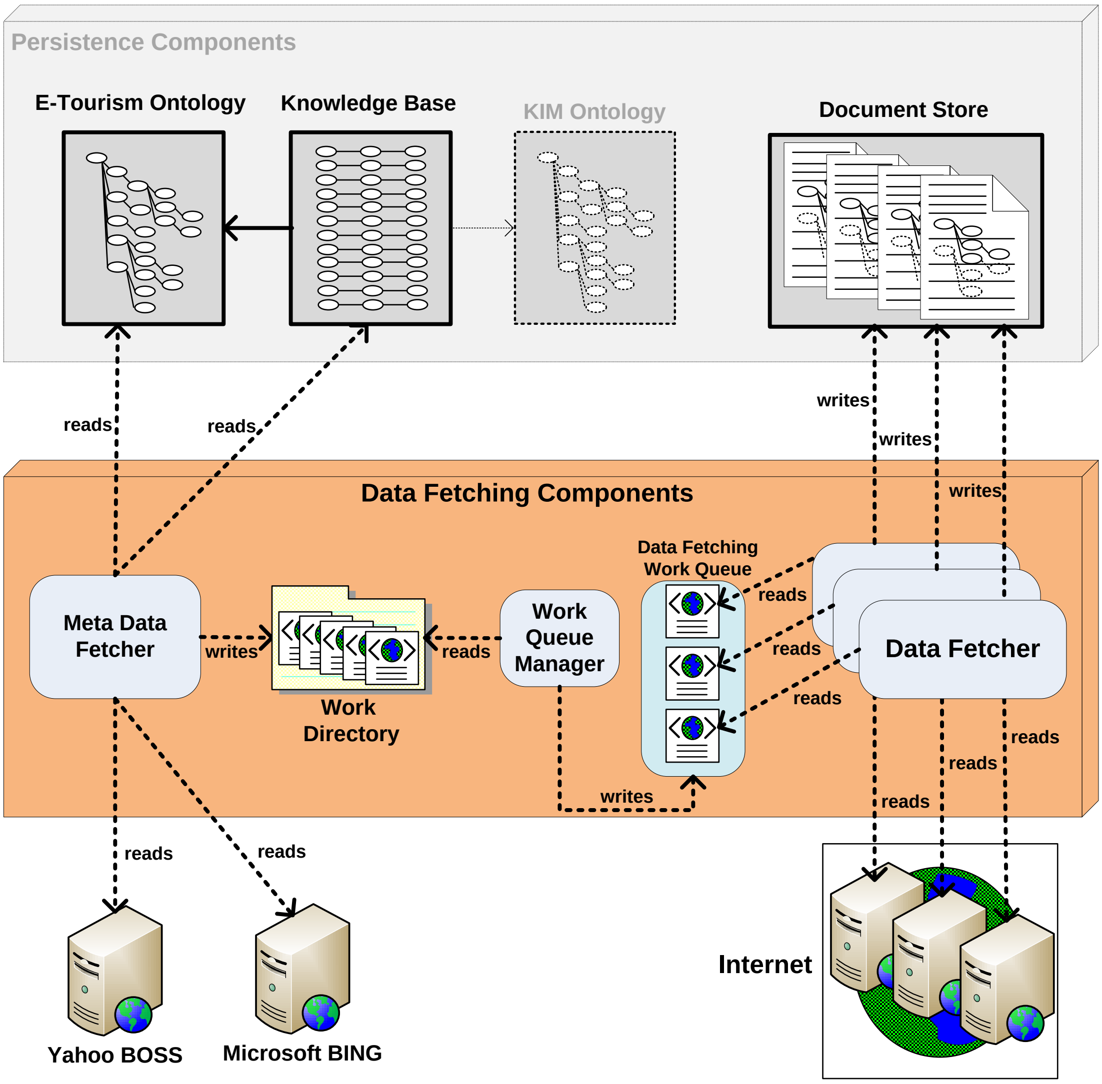

Figure 3.6: Data Fetching Components

stored in a work directory.

One file may contain meta-data for multiple instances. We decided to use this approach instead of letting the Metadata Fetchers directly forward the meta-data to the Data Fetchers, because in case the Document Store needs to be rebuilt, the Metadata Fetchers do not need to fetch all data again. Furthermore, it is possible for external services or users to upload XML files, containing meta-data in the pre-defined format, that can be used by the Data Fetchers to populate the Document Store. The Work Queue Manager depicted in Figure 3.6 is an integral part of the system and distributes work to the different components of the system. It is responsible for populating the Data Fetching Work Queue with work packages generated from the meta-data files stored in the temporary work directory. Data Fetchers fetch documents by accessing the URLs stored as part of the meta-data. They analyze the size and type of a document and discard it in case of a wrong document type or if it is too big in size. The size of a document is checked because most documents which exceed a certain size do not contain any valuable information and are just “spam pages” that contain just mere listings of non-related keywords and links to boost the rank of these sites in search engines.

Fetched documents are stored in their original and processed form. The original form is e.g. the HTML data of the document and the processed form the document object model used by GATE [25]. GATE is an extendable framework for text engineering that provides various useful plug-ins. HS^3 uses GATE to parse and process documents. GATE uses an object model to represent documents; this object model offers the possibility to associate meta-data with a document and to annotate the document. Data Fetchers group similar documents that were fetched for a specific instance in the KB in a so-called GATE Document Corpus. The Work Queue Manager uses the status variable to track the current status of a GATE Document Corpus.

Fetching data from Web resources The Metadata Fetcher starts fetching meta-data from Yahoo BOSS, by using the plug-ins defined for the ontology, as soon as the KB is populated with basic knowledge. This data is written to the work directory. The work directory is monitored by the Work Queue Manager. If a new meta-data file is placed in the work directory, the Work Queue Manager processes the file and places a work package in the Data Fetching Work Queue. One of the Data Fetchers takes the work package from the queue and starts fetching data for the corresponding concept instance. When a Data Fetcher has finished fetching documents related to a certain concept instance of the KB, it sets the corresponding GATE Document Corpus to the state `DOCUMENTS_FETCHED` to signal the Work Queue Manager that the fetched data is ready to be annotated by the Annotations components. For example, if the instance “Elisabeth Hotel” of the concept `Hotel` is part of the KB, the Metadata Fetcher uses the Accommodation plug-in to retrieve the textual representation of the instance “Elisabeth Hotel”, the textual representation of its location, namely “Mayrhofen”, and the zip code to generate a search query. This query is issued at Yahoo BOSS. Yahoo BOSS returns several results which are parsed by the Metadata Fetcher, stored in a work package and placed in the work directory. The Work Queue Manager places the work package in the Data Fetching Work Queue. The Data Fetcher uses the meta information, which also includes the URLs of the documents that may contain information about “Elisabeth Hotel”, and fetches the corresponding documents. The Data Fetcher places all fetched documents in a GATE Corpus, marks it with the instance URI of “Elisabeth Hotel” and sets its status to `DOCUMENTS_FETCHED`.

Annotation Components

The Annotation & Indexing components of the Hybrid Semantic Search System are depicted in Figure 3.7. Annotators process GATE Document Corpora that have been created by Data Fetchers. To process a corpus every Annotator instantiates its own Annotation Pipeline which is realized by means of GATE Processing Pipelines. A GATE Processing Pipeline can hold multiple Processing Resources (PRs). Every Processing Resource operates on the document that is handed to the Annotation Pipeline. A Processing Resource of an Annotation Pipeline can either be local or remote. Local Processing resources are instantiated and initialized when the Annotation Pipeline is created.

For accessing remote resources we developed wrappers that wrap the remote resource and use Java Remote Method Invocation (Java RMI) or Web Services to transfer data. Annotators may also use a Meta-Annotation Pipeline that consists of multiple other Annotation Pipelines. A Meta- Annotation Pipeline forwards a document to all its encapsulated Annotation Pipelines and aggregates the result. The default Annotation Pipeline used by Annotators is a Meta-Annotation Pipeline that consists of the system’s custom Annotation Pipeline and a remote Annotation Pipeline that accesses the annotation functionality of the KIM Server via Java RMI. The remote Annotation Pipeline has the advantage that the workload can be distributed across several machines. The system’s custom Annotation Pipeline can either use Apolda [96] or the Concept Mapper of UIMA [43] to annotate documents. Apolda utilizes finite state machines to identify occurrences of concepts and instances in documents. The original Apolda was bound to the ontology and KB implementation of GATE version 4 and 5 where it was not possible to use a KB that is located on a different server than the one where the annotator was running. Therefore, the system and the KB would have had to reside on the same server which would have eventually led to a resource shortage. We developed a modified version that can access the KB remotely and adapted Apolda’s internal data representation to save memory by creating a lightweight model for ontologies that is sufficient for annotations tasks. This lightweight model includes only information that is actually needed for annotation tasks. The Concept Mapper of UIMA has been integrated into GATE by using a Wrapper that transforms UIMA Annotations to GATE Annotations. The default behavior of the Annotation Pipeline is to use the Concept Mapper of UIMA, because it offers more configuration possibilities and recognizes permutations of concept and instance descriptions in the documents out of the box. After the Annotation pipelines have processed the corpus, the Meta-Annotation pipeline aggregates the annotations of both pipelines. Consider the following example. The Data Fetcher has fetched several documents that may contain information about the instance "Hotel Elisabeth", stored them in a GATE Document Corpus, created a work package and set the appropriate status of the work package. Now the Work Queue Manager places the work package into the Annotation Queue. The Annotator reads the work package and uses the default UIMA Pipeline to annotate every document within the GATE Document Corpus. In the presented example the UIMA Pipeline uses the Concept Mapper to annotate all occurrences of the instance "Hotel Elisabeth" in every document. The Concept Mapper also recognizes permutations of the textual representation of "Hotel Elisabeth" such as "Elisabeth Hotel", "Elisabeth-hotel" or "Elisabeth-Hotel". Every created annotation holds the URI of the instance "Elisabeth Hotel" and the URI of the corresponding concept, namely `Hotel`. Finally, when all documents of the GATE Document corpus have been annotated, the Annotator sets the status of the corpus to `DOCUMENTS_ANNOTATED`.

Instance disambiguation by the Annotator Identifying instances of concepts within documents is a complicated task, because textual descriptions of instances can be ambiguous, especially when their descriptions contain common terms. For the purpose of disambiguation and to calculate an Annotation Score we introduced so-called *Concept Identification Graphs* (CIG), which utilize the information of related concepts to identify a specific concept. A CIG is overlaid on a sub-graph of the original ontology graph. We use the RDF Reification functionality to define CIGs. Figure 3.8 shows an example of a simple CIG that is used to disambiguate tourism accommodations. The graphical representation of the CIG is depicted on the left side and an excerpt of the RDF Reification statements on the right side.

Every Reification statement describes two nodes (concepts) and an edge (property) of the original ontology. The statement can be equipped with a `mandatory` and a `score` attribute. If the `mandatory` attribute is set to "true", the textual representation of the instance of this concept must occur in the document, otherwise the concept attached to this CIG will not be matched during the annotation process and no annotation will be created. In the example in Figure 3.8 the Reification statement `<Accommodation, accommodationLocation, Location>` is equipped with a `mandatory` attribute set to "true", therefore an instance of `Accommodation` would only be annotated in case the textual description of the location where the accommodation resides is also mentioned in the document. This simple approach helps to disambiguate several accommodations with the same name that reside in different locations. All other concepts which are part

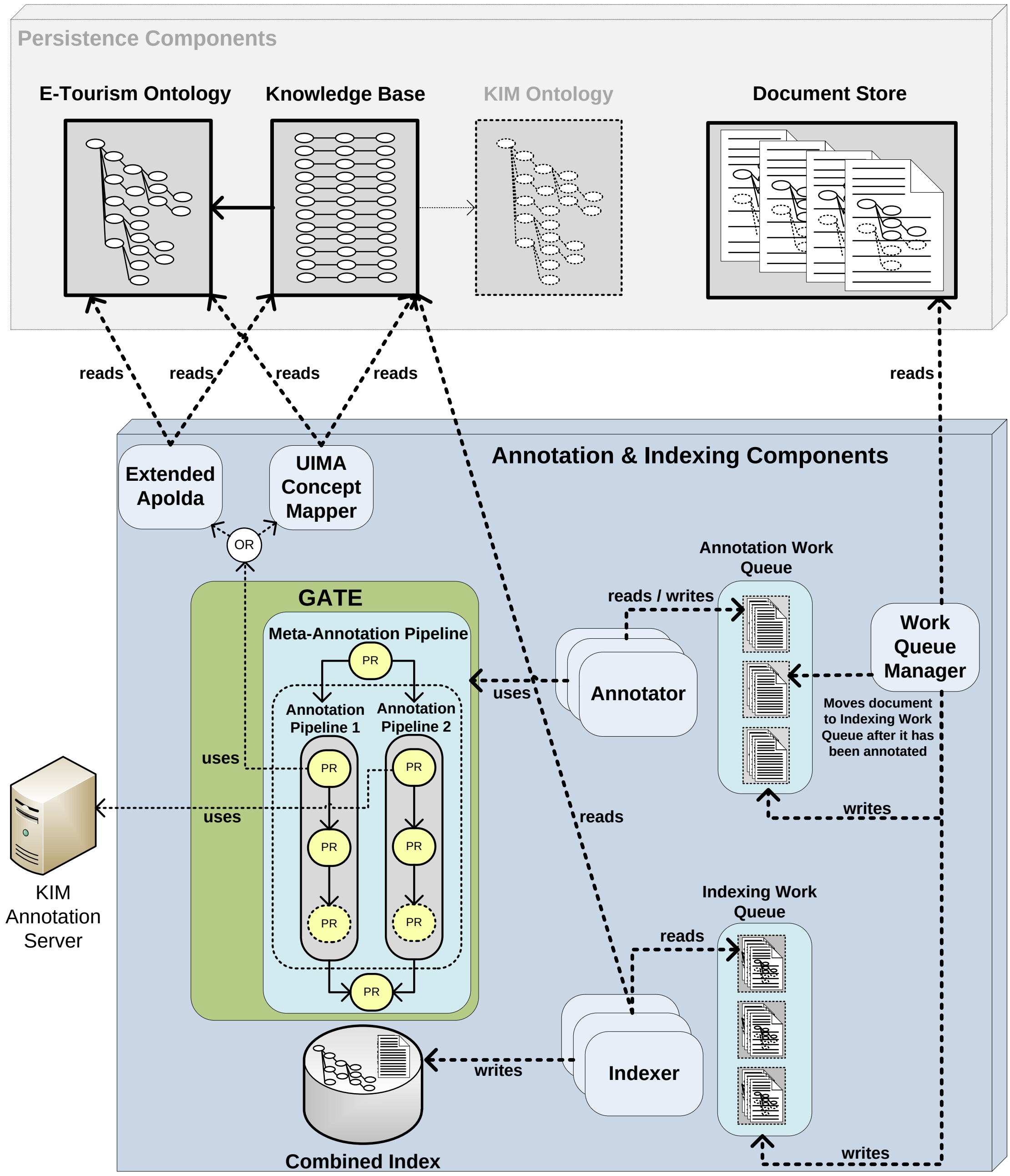

Figure 3.7: Annotation & Indexing Components

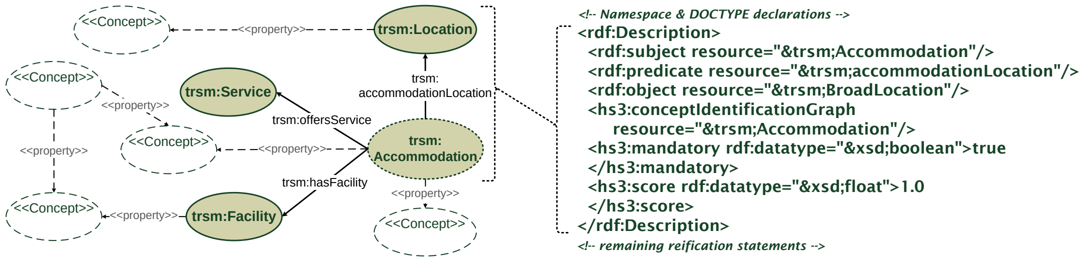

Figure 3.8: Concept Identification Graph and corresponding reification statements

of the CIG, which either don't include the `mandatory` attribute, or have it set to "false" are solely used to calculate the Annotation Score. The score of a concept in the CIG defines to which extent the occurrence of the specific concept in a document contributes to the overall Annotation Score of the concept. Therefore, those concepts which help to identify the CIG's main concept the most will get high scores. Currently, the score of a concept is set by a Knowledge Engineer. Another approach would be to use Machine Learning to analyze annotated documents and generate the scores automatically. The Annotation Score is incorporated into the Relevance Score of the combined index which is described in Section 3.4.1.

Indexing Component

The Indexing Component of HS^3 operates on the annotated document corpus. For scalability reasons multiple Indexers can access the annotated document corpus and Triple Store simultaneously, and modify the combined index in parallel. To generate the combined index, Indexers analyze the annotations and the text of a document. The mechanism is described as pseudo code in Algorithm 3.9 and explained in the following paragraphs.

An Indexer loads an annotated document from the document corpus and creates a new Lucene document. For every annotation in the document it gets the corresponding `IndexGraph` and `IndexGraphTuples`. Every concept in the ontology may carry an `IndexGraph` that defines the graph that should be indexed for this specific concept. This graph is used by the search mechanism to conduct searches for the specific concept. The `IndexGraph` is attached as an RDF Construct Query to the concept. The `IndexGraphTuples` are similar to a set of rows with columns. Figure 3.10 depicts the index structure. Every `IndexGraphTuple` has a `Subject`, an `Object` and a `Predicate`. The `Subject` contains the URI of an instance and the URI of the concept of this instance. The `Object` has the same attributes. The `Predicate` contains the URI of a property. An example would be an `IndexGraphTuple` that

```
Input : Annotated document GateDoc
iW = getIndexWriter();
L = new LuceneDocument();
foreach Annotation i in GateDoc do
    cI = i.getConceptInstance();
    if cI already indexed for L then
        continue with next i;
    end
    iGTP = getIndexGraphTuples(cI);
    if iGTP == null then
        iGTP = loadIndexGraphTuples(cI);
    end
    iGRC = getIndexGraphRootConcept(i);
    r = getInstanceRelevance(cI,GateDoc);
    L.addField(iGRC.name + "_" + r,iGTP);
    iC = getInstanceContext(cI);
    L.addField(iGRC.context + "_" + r,iC);
end
dS = generateSummary(G);
L.addField(iGRC.summary,dS);
iW.index(L);
```

Figure 3.9: Simplified version of the indexing mechanism

has a `Subject` which contains the instance URI of “Hotel Elisabeth” and the concept URI of the concept `Hotel`, a `Predicate` which contains the property URI of the property *locatedIn* and an `Object` that contains the instance URI of “Mayrhofen” and the concept URI of the concept `Location`. An `IndexGraphTuple` correspond to the formal definition of (ai, sc, s, r, oc, o) in Equation (3.7) at the beginning of Section 3.4. The annotation instance ai corresponds to the `Annotation` class depicted in Figure 3.10 which is the same for all tuples that belong to an `IndexGraph` of an annotated instance. The `IndexGraph-RootConcept` is stored in the `IndexGraph` class. The `IndexGraph-RootConcept` is used to realize a grouping mechanism that is needed to conduct efficient approximate searches upon the index. The `IndexGraph--RootConcepts` are used as field names in the Lucene Document. For example, the `IndexGraph-RootConcept` for the concepts `Hotel` and `GuestHouse` is `Acc-ommodation`, which is their super-concept in the ontology as well. Therefore, if a user would search for a hotel the search algorithm would look in the field `Accommodation` instead of `Hotel`, because it is the `IndexGraph-RootConcept` of `Hotel`. Since the `IndexGraphs` for guesthouses are also stored in this field, the algorithm is able to return approximate matches containing guesthouses in case no hotels exist that match the user's search criteria. Another reason for the usage of an `IndexGraph-RootConcept` is that an `IndexGraph` needs to be defined only for a super-concept and can be inherited by all sub-concepts. Therefore, to continue the previous example, the `IndexGraph` is only defined for the concept `Accommodation` and inherited to the concepts `Hotel` and `GuestHouse`. Every `IndexGraphTuple` holds information that is usually encoded in an RDF statement consisting of subject, predicate and object. But, in addition, the subject and object concept types are also included. In RDF this would be accomplished by using two additional RDF statements that declare the type of the subject and object via the `rdf:type` property. For efficiency reasons and to maintain a smaller index this information is encoded in a single tuple in the combined index. Any RDF graph is converted into `IndexGraphTuples` before it is indexed via SIREn. A document may be related to multiple `IndexGraphs` if the root nodes of these `IndexGraphs` are referenced via an annotation in the corresponding document. The root node of an `IndexGraph` is the one that is not referenced by any other node within the graph. By calling the method `loadIndexGraphTuples` the Indexer issues the RDF Construct Query to the Triple Store and transforms the result into the `IndexGraphTuple` structure. The system caches `IndexGraphTuples` to avoid repeatedly querying the Triple Store for the same instances. In addition the Indexer extracts the context of the current instance by calling the `getInstanceContext` method. The context of an instance is defined as the terms which surround the instance's annotation in a specific document. All terms to the left and to the right of an instance's annotation, that have a distance smaller than a specified threshold, are returned as context. These terms are indexed as `IndexGraphContextTuples`. The `IndexGraphContextTuple` class is depicted in Figure 3.10. Every `IndexGraphContextTuple` may contain multiple `Keywords` and has one `SubjectContext` which defines the instance and corresponding concept that are surrounded by the contained `Keywords`. The parameter *LeftOrRight* defines whether the Keywords occur to the left or to the right of the subject. It can either have the value `LeftContext` or `RightContext`. However, the current implementation does not differentiate between the `LeftContext` and `RightContext`, but rather searches both contexts. The differentiation was designed for future use. For simplicity reasons we will refer to the `LeftContext` and `RightContext` as `describedAs`, which means that both contexts are searched.

The `RelevanceMarker` of the `IndexGraph` reflects the relevance of the instance, described by the `IndexGraph`, in the specific document. The relevance of an instance in a specific document is calculated by multiplying its annotation's score

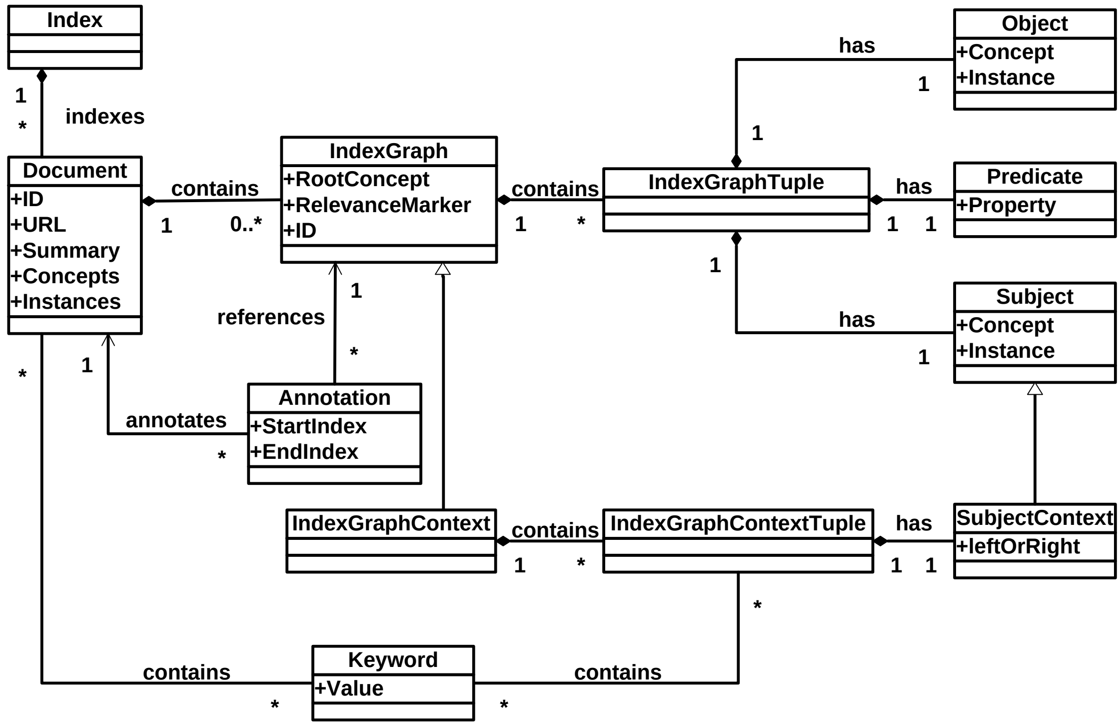

Figure 3.10: Combined Index Structure

(as described in Section 3.4.1) with its annotation's position in the document (e.g. the value 10 for title or the value 1 for body) and its annotation frequency in the document. The `RelevanceMarker` is used during the search process to restrict searches only to the most relevant instances of concepts within a document. This approach helps to reduce noise in the result, because common instances may be mentioned in many documents, even though the documents are mainly about completely different instances. Documents from the tourism domain often contain names of several tourism destinations such as Salzburg, Kitzbühel or France which are annotated by the Annotation Components. However, the documents that contain these terms might mainly hold information about hotels, guesthouses or apartments that are located in these tourism destinations. Consider a document that holds mainly information about the hotel Kitzhof in Kitzbühel. In that case the `RelevanceMarker` for the instance Kitzhof needs to be higher than the `RelevanceMarker` for the instance Kitzbühel for this specific document. However, instances of lower relevance for a specific document need to be kept in the index, because these are used for approximate matches in case no exact matches can be found. Furthermore, instances of lower relevance, might just be of low relevance because the Information Extraction processes were not able to extract accurate information. Still, these instances can be useful in combination with keyword-based searches, which leverage information missed by the Information Extraction process or information which cannot be represented with the used ontology. Finally, the Indexer generates a summary of the document in HTML format, including the annotations and the context surrounding them. To eliminate the need for generating the summary for a document for every search request from scratch, it is generated once and stored in the index. The document summary is included as part of the search result for every result entry. The visual representation of a summary in the Web Browser is depicted in Figure 3.15 below the first bar labeled with "(– show more –)". In addition, all concepts and instances that occur in the document are stored in the index to provide users with a list of concepts and instances.

Search & Ranking Components

The Search & Ranking components use the combined index generated by the Indexers of the Indexing component. HS^3 can handle three different types of search queries i) keyword-only queries ii) concept-based queries and iii) combined queries containing keywords and concepts. However, all queries are transformed and issued against the combined index. Instead of just translating the concept-based and combined queries into high-level query languages such as SPARQL or SeRQL, the query is transformed into tuples and issued against the combined index. Furthermore, keywords which either describe concepts more specifically in the query or represent concepts themselves, which are not part of the ontology, are incorporated as tuples as well. For example the keyword "child-friendly" can be used to describe the concept `Hotel` more specifically. A keyword that describes a concept more specifically is transformed into a tuple that consists of three entries: the concept that is modified, whether it occurs in the left or right context of the concept and the keyword itself. By transforming the query into tuples that match the structure of the combined index, a fast retrieval of documents matching the query can be accomplished. During the transformation process the ontology is used to expand the query with sub-concepts and similar concepts, to return approximate results in case no exact matches have been found. The query expansion mechanism leverages so-called *Realms* to pick only meaningful concepts for the query expansion.

A *Realm* holds concepts that are semantically related and are of interest to the user (e.g. similar concepts). To illustrate the query transformation process consider the query depicted in Figure 3.12. The keyword query could be something such as "child-friendly hotel providing steam bath located in mayrhofen offering rafting". By using the interactive ontology-aware keyword-based input mechanism described in Section 3.5, the query presented in the first box of Figure 3.12 is interactively created. Subsequently this query is transformed into the graph query depicted in the second box of Figure 3.12. This graph representation is transformed into a tuple query, depicted in the third box of Figure 3.12) that consists of a disjunction of conjunctions and is issued against the combined index. Algorithm 3.11, depicts the transformation mechanism, which expects an ordered set of `Resources` as input. The ordered set of Resources is generated by the query formulation mechanism. Every Resource is either an instance of a concept, a concept, a property or a keyword, which can be related to other `Resources` in the query.

```
Input : Ordered Set OS of Resources
Output : Disjunction List DL of conjunctions
DL = new DisjunctionList();
foreach Resource r in OS do
    type = r.Type();
    if type == Concept then
        rR = r.getRelatedResources();
        tCList = createTupleConjunctions(r,rR);
        foreach TupleConjunction tC in tCList do
            DL.add(tC);
        end
    else if type == Keyword then
        DL.add(r);
    else
        skip current r;
    end
end
```

Figure 3.11: Simplified version of the query transformation mechanism

The transformation mechanism iterates through all `Resources` and checks whether the `Resource` is of type `Concept` or `Keyword`. In case the `Resource` is a concept, all related `Resources` are retrieved, which can either be keywords that modify the current concept or other concepts. Related concepts are always related via a property. Therefore, the generated conjunction is a tuple that consists of the current `Resource's` concept, the related `Resource's` concept and the property that connects the two `Resource's` concepts. In case the related `Resource` is a keyword the tuple will contain the current `Resource's` concept, a property that states that the keyword belongs to the concept's context and the keyword itself. The resources and their relations are stored in a TupleConjunction object, as described in Algorithm 3.11.

Every generated TupleConjunction is equipped with a score. The score is used by the

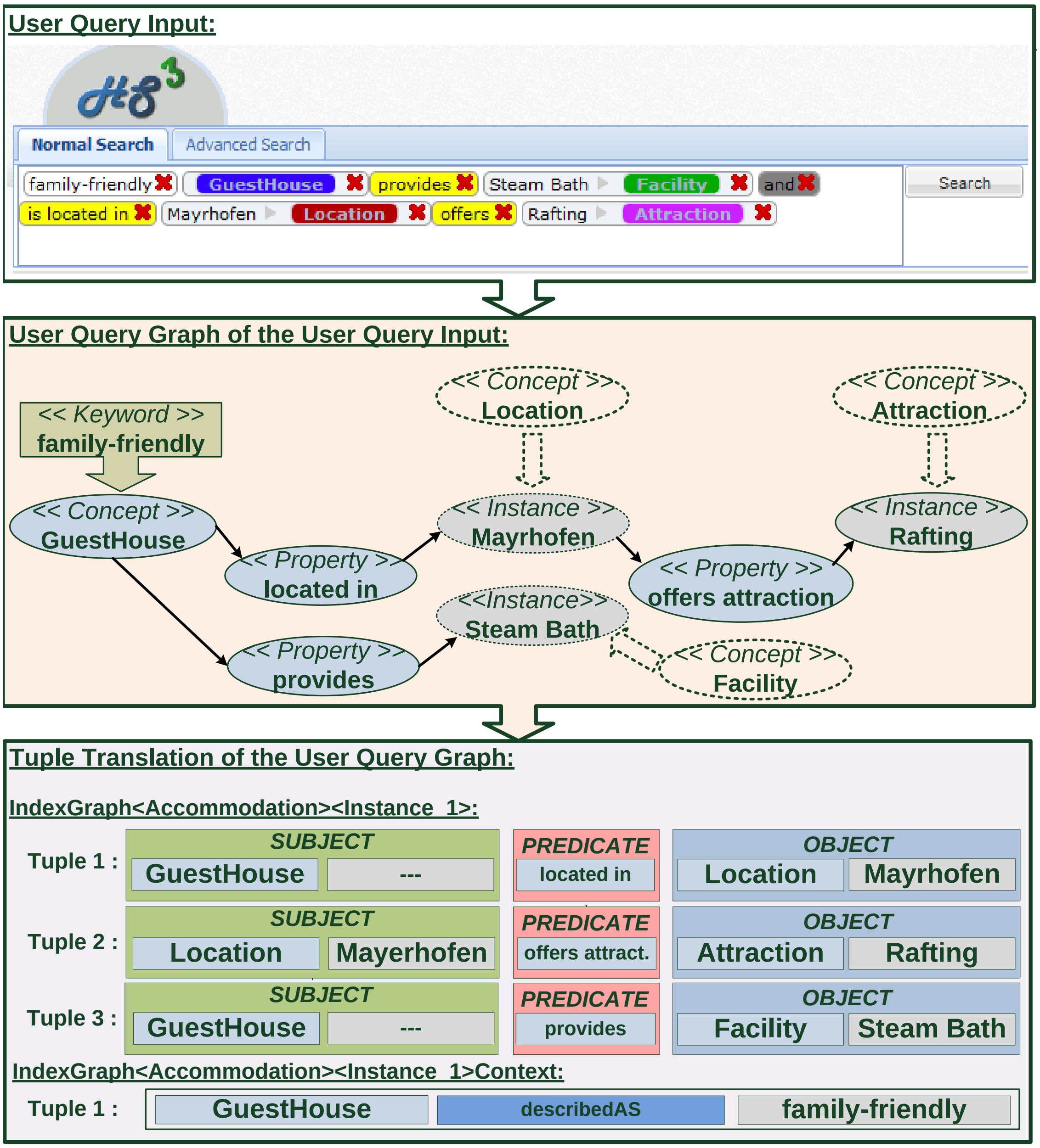

Figure 3.12: Graph Query Translation

ranking mechanism to calculate the relevance of every matched document to the user's query. The base score of a TupleConjunction in the query is determined by the types of the two `Resources` that are connected via the property. The highest score is assigned to a TupleConjunction that contains two instances of a concept and a property, because this TupleConjunction expressed the information need of the user in the most specific way. The next higher score is assigned to a TupleConjunction that contains an instance of a concept and a concept. And the lowest score is given to a TupleConjunction that contains just two concepts and a property. Therefore, TupleConjunctions which represent the user's information need in the most specific way get assigned the highest score in the query. For example, consider a tourism ontology, where a TupleConjunction that consists of the instance "Hotel Elisabeth", the property `located in` and the instance "Mayrhofen", would express the information need of a user for a specific hotel in a specific city. According to this TupleConjunction the user would be looking for information about the hotel named "Elisabeth" which is located in Mayrhofen. Therefore, those documents which are about this specific hotel should be ranked the highest and all other hotels in Mayrhofen that might also be of interest to the user ranked lower.

However, a TupleConjunction that consists of the concept `Hotel`, the property `located in` and the instance "Mayrhofen", expresses the information need of a user in a much broader sense. Therefore, the user is not looking for a specific hotel, but rather for any hotel that is located in Mayrhofen. Hence, the user is invariant about the ranking of documents describing hotels as long as the described hotels are located in Mayrhofen. Any other hotel that is not located in Mayrhofen is of little interest to the user and is ranked lower. An example for a least specific TupleConjunction would be one that contains the concept `Hotel`, the property `located in` and the concept `Near River`. In this case any hotel that is located near a river would be part of the result but ranked arbitrarily.

In terms of query expansion, every TupleConjunction is extended with all possible concept and instance combinations. Therefore, the TupleConjunction "Hotel Elisabeth" - `located in` - "Mayrhofen" is extended with the TupleConjunctions `Hotel` - `located in` -"Mayrhofen", "Hotel Elisabeth" - `located in` - `Location` and `Hotel` - `located in` - `Location`. All TupleConjunction get assigned a score according to their information specificity. Therefore, in the result those documents that are about the hotel named "Elisabeth" which is located in Mayrhofen are ranked first, followed by those about hotels in Mayrhofen, followed by other hotels that are named "Elisabeth" but are not located in Mayrhofen and finally any hotel that is located in any location is listed. Furthermore, the position of a TupleConjunction in the query graph is incorporated into its score. The farther away a TupleConjunction is from the root TupleConjunction of its query graph, the lower is its impact on the overall score of the query graph. Therefore, those TupleConjunctions which are stated first in the query graph are considered as more relevant than those at the outer border of the query graph. Consider the query graph depicted in the upper region of Figure 3.12. In this query graph the root Concept of the graph is `GuestHouse` which is modified by the property `located in` and the instance "Mayrhofen". The instance "Mayrhofen" is modified by the property `offers attraction` and the instance "Rafting". Since "Mayrhofen" modifies the root concept of the graph and "Rafting" only modifies "Mayrhofen", the TupleConjunction containing the instance "Rafting" gets assigned a lower query score than the TupleConjunction containing "Mayrhofen".

In terms of ranking this means that documents which describe guesthouses that are located in Mayrhofen where it is not possible to go rafting will be ranked higher than documents that describe guesthouses located in an arbitrary city where it is possible to go rafting. Since the user specifically stated that she is interested in guesthouses in Mayrhofen, the information that the guesthouse should be in Mayrhofen is deemed as more important than that rafting should be possible, because Mayrhofen is an instance rather than a concept and it directly modifies the root concept `GuestHouse`. If the user would have stated the concept `Location` instead of the instance Mayrhofen, those documents that describe guesthouses that are located in a location where rafting is possible would have been ranked higher than those documents which describe guesthouses which are located in locations where rafting is not possible. A complete showcase of the search and ranking functionality is given in Section 3.6 where HS^3's application to the tourism domain is demonstrated.

3.5 Interactive ontology-aware Query Formulation

The user interface is crucial to the success and acceptance of a system or Website in general [72]. The prevalent user interface of search engines consists of a text field which accepts arbitrary text and an optional search button such as the user interfaces provided by Google, Yahoo or Bing. Considering the widespread usage of these search engines, the simple and clear input paradigm of an interface that mainly consists of one input text field is widely accepted by users of the World Wide Web. Semantic Search Systems offer a variety of user interface types such as keyword-based interfaces [7], natural language interfaces [91], graphical query interfaces [101], interfaces that use various HTML form elements for query formulation [61, 10] or a combination of them. However, Semantic Search Systems introduce an additional level of complexity, because the user needs to know the terms (concepts) and their relations to formulate meaningful queries. Even tough some interfaces might be as simple as a text box, a user needs to know an RDF query language such as SPARQL, SeRQL or even a complicated custom query language to interact with the system. To the best of our knowledge the only Semantic Search Systems which offer an interface that combines keyword-based and concept-based input has been introduced by Bhagdev et al. [10] and Bast et al. [7]. However, the interface presented by Bhagdev et al. separates keyword-based and concept-based input. Therefore, the interface presents no coherent input mechanism to the user. The interface of ESTER presented by Bast et al. is somewhat scattered and uses one input box and three different static input/info boxes that assist the user by formulating her query.

Therefore, all of these interfaces suffer from one or more of the following problems: i) the interface is too complicated for the average Internet user or the user needs considerable time to familiarize herself with the interface ii) the user needs a before-hand knowledge of the ontology to be able to formulate queries to efficiently access the knowledge of the system, iii) keywords and concepts are treated as separate inputs. To overcome these obstacles we introduce an interactive ontology-aware keyword-based input mechanism that uses the established text field input paradigm, combines it with interactive concept, property and instance suggestion and helps the user to formulate an ontology-aware query without prior knowledge of the ontology. Therefore, while formulating the query the user familiarizes herself with the ontology, its concepts, their relations and the instances. Furthermore, we amalgamate keyword and concept-based input by letting the user enrich the query with keywords. One of the main objectives was to create an interface that lets a user input her query, while it concurrently suggests suitable concept and instance without interrupting the user's input flow. Furthermore, the interface automatically structures the input in a graph-like manner.

In the following we describe the mechanisms, used to realize the interactive ontology-aware keyword-based input mechanism and showcase the user interface by means of a query from the tourism domain. Part of the user interface, namely the input text field, is depicted in Figure 3.13. The interface relies on three main components. The first component is the OntologyWrapper which is located on the server side. Upon accessing the Web Site of HS^3 for the first time the OntologyWrapper sends a light-weight model of the ontology to the client. This model holds all concepts, all properties, their relations and structural information. The client side model is used to suggest super-concepts or similar concepts when a concept is recognized in the input text field. Furthermore, the model is used to show a popup list with properties as soon as the user chooses one of the suggested concepts or instances. The second main component is the Concept & Instance Suggestor which resides on the server side. The Concept & Instance Suggestor holds an optimized index of all textual descriptions of concepts and instances. In addition it holds the instance and/or concept URI as well as a reference URL in case of an instance. The reference URL points to the document that has the highest relevance for the specific instance. The relevance is determined by the Relevance Score. Asynchronous calls are issued to the Concept & Instance Suggestor whenever the user hits a key, by using the asynchronous callback functionality of GWT[20], which returns those ten concepts and instances that have a textual description with the smallest Levenshtein [71] distance to the current input term. By using the Levenshtein distance as similarity measurement it is possible to suggest suitable instances and concepts despite the existence of typos in the query.

The final component is the GWT-based user interface on the client side which displays the user query graphically. Concepts, properties and instances are displayed within colored oval entities, which can be edited and removed by the user. For a better understanding of the interface we will showcase it by means of an example query depicted in Figure 3.13. In our scenario the user is interested in accommodations, preferably hotels, which offer a steam bath and are located Mayrhofen. The user gets presented with the text field and starts to type “hot”. As soon as the user starts typing, asynchronous requests including the text field’s current input are sent to the Concept & Instance Suggestor which returns an ordered set of concepts and instances that have a sub-string which matches the current term or a Levenshtein distance below a defined threshold. In case of the string “hot” the concept and instance suggestions depicted in the first box of Figure 3.13 are returned. To display super-, sub- and similar concepts in the suggestion list, we make use of the client side ontology model and so-called *Realms*. We introduced *Realms* to group certain concepts of an ontology. The purpose of *Realms* is to group end-user relevant hierarchical concepts, because not all concepts within the hierarchy of an ontology are useful for query construction. Consider a tourism ontology which has a concept `Accommodation` which is a subconcept of the concept `Root`, whereas `Root` is the super-concepts of all concepts in the ontology. Furthermore the concepts `Hotel`, `GuestHouse` and `Farm` are direct sub-concepts of `Accommodation`. A *Realm* holding the concepts `Accommodation`, `Hotel`, `GuestHouse` and `Farm` but excluding `Root` can be used to suggest only valid super-, sub- and similar concepts. Whenever one of the concepts within the *Realm* is returned, only those concepts which are hierarchically related and part of the *Realm* are displayed to the user. The concept `Root`, which is

[20]http://code.google.com/webtoolkit/

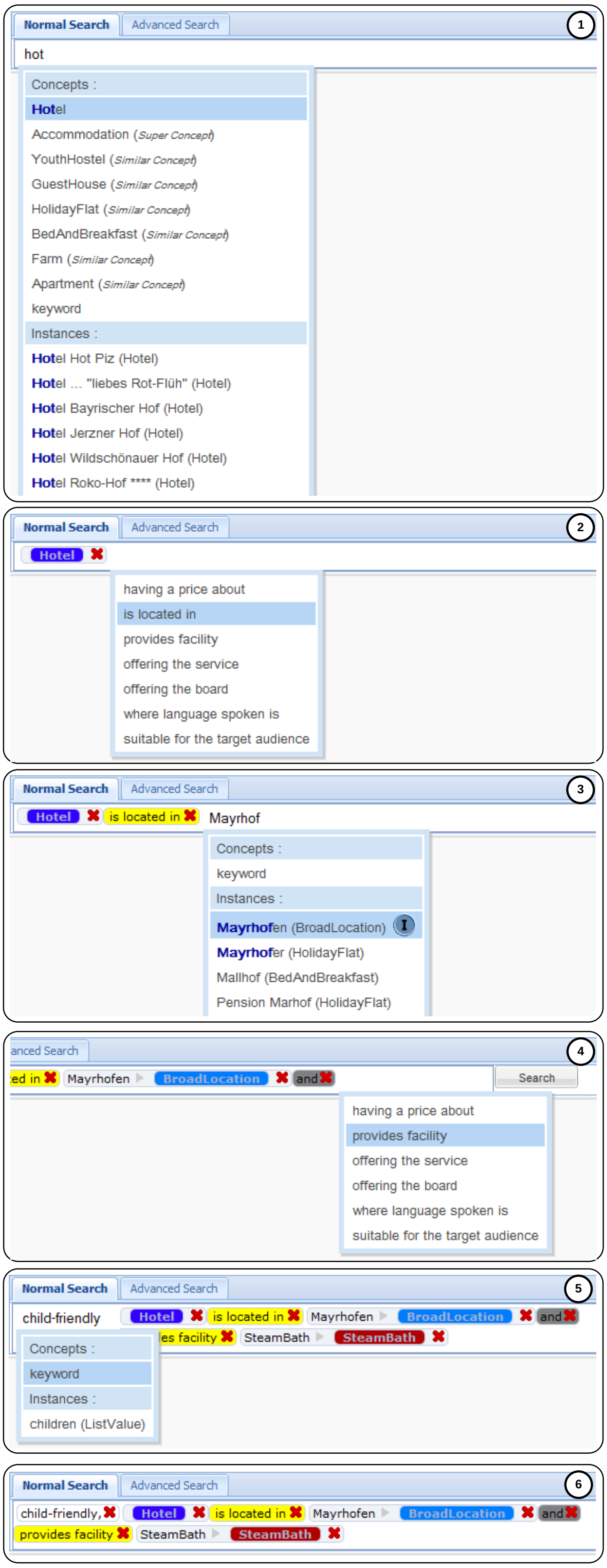

Figure 3.13: Interactive ontology-aware keyword-based input mechanism

hierarchically related but not part of the *Realm* is not displayed, because it is not of any use for the query formulation.

As soon as the user has chosen an appropriate concept or instance, in the presented showcase the concept `Hotel`, the interface shows a popup holding the properties of the concept as depicted in box 2 of Figure 3.13. It is possible to equip all properties of the ontology with a display string, which is used to display the property in the user interface. If no display string is defined, the property will not show up in the suggestion list. This mechanism ensures that properties which are not useful for query formulation or are only used for internal purposes are not propagated to the user interface. Having chosen the `located in` property with the display string “is located in”, the user starts typing “Mayrhof” as depicted in box 3 of Figure 3.13. This time the asynchronous request to the Concept & Instance Suggestor holds the property information of the query as well. This information is used to rank the instance suggestions according to the concept class that is expected by the property type. In case of `located in` a `Location` concept is expected. Since `BroadLocation` is a sub-concept of `Location` the instance “Mayrhofen” is ranked before the instance “Mayrhofer” which is of type `HoldidayFlat`. Furthermore, the instance “Mayrhofen” is annotated with a small icon that carries the letter I. When the user clicks this icon the browser opens the reference URL. This functionality helps users to distinguish between instances that have the same name but refer to different entities. For example there could be a region named Mayrhofen in Salzburg and a region named Mayrhofen in Tyrol. The relevance mechanism described in Section 3.4.1 is used for this functionality.

Having chosen the instance “Mayrhofen” the user wants to add that the hotel should be equipped with a steam bath. Therefore, she types “and” and declares the term as operator from the displayed popup list. This signals the user interface that the user is referring to the concept `Hotel` and it uses the client side ontology model to show available properties of the `Hotel` concept. However, the property `located in` is not shown anymore, since the ontology defines that the property `located in` can only be assigned to one `Location`. Even though $\mathcal{SHIF}^{(D)}$ does not make use of cardinality declarations, we incorporated this functionality for usability reasons.

Subsequently, the user chooses `provides facility` and picks the “SteamBath” instance as depicted in box 4 of Figure 3.13. Since the user is looking for child friendly accommodations, she adds the keyword “child-friendly” to the query (box 5 of Figure 3.13), because the used ontology does not offer a corresponding concept. In that case the keyword “child-friendly” describes the concept `Hotel` more specifically. Therefore, the search algorithm will rank those documents higher that contain annotations of hotels which are surrounded by the keyword "child-friendly". In case the keyword "child-friendly" would be append to the end of the query, documents that contain annotations of hotels and the keyword "child-friendly" anywhere in the document would be ranked highest. The final query is depicted in box 6 of Figure 3.13.

3.6 Putting hybrid semantic search to practice

In the following section we showcase the application of HS^3 to the tourism domain. The initial dataset for the KB was generated from trusted information that is stored in an RDBMS. The database, provided by the e-Tourism portal Tiscover[21] and henceforth called e-Tourism database, holds detailed information about 14,000 Austrian accommodations, 1,900 Austrian locations, related leisure activities and relations among them. We had to find an appropriate tourism ontology that is capable of representing the entities stored in the RDBMS. We had the three major requirements that the ontology has to be publicly available; that it needs to have concept classes for all main entities in our e-Tourism database and that it is manageable with respect to the number of concepts. Furthermore, we favored ontologies that are represented in OWL, because we intended to use OWL inference engines to merge different ontologies and extract implicit information.

Prantner et al. [78] published an overview of several tourism ontologies and suitable ontology tools. We had a deeper look at the most promising ontologies mentioned in their work and evaluated a couple others with respect to our needs. According to Prantner et al. the Mondeca Tourism Ontology[22] holds about 1000 unique tourism concepts which are derived from the tourism thesaurus maintained by the World Tourism Organization. The ontology is represented in OWL and offers concepts such as cultural objects, tourism packages and multimedia content. This ontology is too detailed and therefore not applicable for the entities of the e-Tourism database, because only a very small fraction of concepts available in the ontology are needed. The e-Tourism Ontology maintained by Deri[23] focuses on accommodations and activities which are represented in great detail via properties. This ontology represents the major concepts of e-Tourism that are also part of our e-Tourism database. Nevertheless, the level of detail of the concepts is high and most of the data is not available in this level of detail in the e-Tourism database. A promising ontology is the Harmonise ontology [45], which was created as part of the European Harmonise project aiming to support the exchange of travel and tourism information. The ontology includes accommodations, events, activities, attractions, gastronomies, lo-

[21] http://www.tiscover.com/
[22] http://www.mondeca.com/
[23] http://e-tourism.deri.at/ont/e-tourism.owl

cations and their relations. The main idea was to create an ontology that holds a common set of concepts in the tourism domain. This set of concepts in combination with a set of mapping tools should facilitate the exchange of information among different information systems without having to adapt their internal data standards. The Harmonise ontology includes about 56 concepts. All concepts are subclasses of the `Root` concept but do not have any subclasses themselves.

We decided to use the Harmonise ontology because most entities from the e-Tourism database can be transformed to instances of the ontology's concepts. Nevertheless, we had to adapt the Harmonise ontology because some data which is included in the e-Tourism database has no corresponding concept or attribute in the Harmonise ontology. We marked all adaptations so they can be distinguished from concepts and attributes of the original Harmonise ontology. For some concepts we introduced sub-concepts to reflect the level of detail of the corresponding entities in the e-Tourism database. The introduction of sub-concepts still conforms to the original ontology, because it is possible to just use the super-concept without having knowledge of the more detailed sub-concept. HS^3 can leverage the additional information during the annotation process and conduct more detailed searches. We used the Transformation Engine to create instances of the concepts `Accommodation`, `Attraction`, `Facility`, `Gastro`, `Link` and `Location`. Part of the adapted Harmonise ontology, namely the `Location`, `Attraction`, `Facility` and `Accommodation` concepts with their sub-concepts are depicted in Figure 3.14. The entire listing of all Harmonise concepts is available at the HarmoNet Web Site[24].

The sub-concept `BroadLocation` and `SpecificLocation` have been introduced to specify the `Location` concept more accurately. The `BroadLocation` concept is used to specify a wide area that includes many `SpecificLocations`. An example for a `BroadLocation` would be "Kitzbuehel". A `SpecificLoaction` is part of a `BroadLocation`. An example for a `SpecificLocation` would be "Waldhofweg 7" which is located in "Kitzbuehel". This distinction was chosen so users can conduct broader searches and due to the fact that the e-Tourism database only holds information about towns but no information about the specific location of an accommodation or attraction within the town.

To make use of the comprehensive knowledge stored in the KIM KB we incorporated the KIM Ontology and KB into our system. We mapped concepts of the Harmonise ontology to similar concepts of the KIM Ontology which includes about 250 concepts and 40 properties. The KIM KB contains about 77,500 entities with more than 110,000

[24] http://www.harmonet.org/

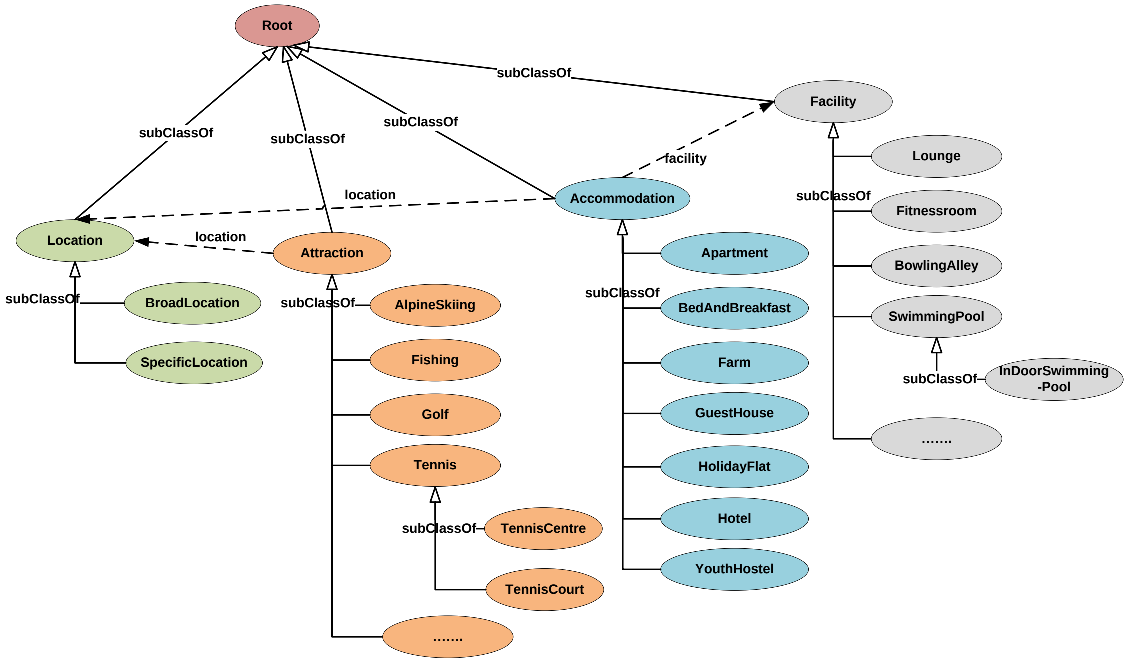

Figure 3.14: Part of the adapted Harmonise ontology

aliases. Aliases are especially useful for the NER (Named Entity Recognition) process, because these are used to find textual representations of concepts and instances in documents. Where applicable, we mapped equivalent classes of both ontologies via the `equivalentClass` property of OWL. The Transformation Engine was used to transform the data stored in the RDBMS to RDF conforming to the chosen ontology. This data was fed into a SESAME Triple Store that uses the OWLIM Sail implementation. The Data Fetchers fetched up to 10 related Web documents per accommodation and location from the World Wide Web, by using meta data provided by the Meta Data Fetchers. This resulted in approximately 150,000 documents that were annotated by the Annotators and indexed by the Indexers to generate the combined index.

To showcase HS^3's functionality consider the query depicted in box 6 of Figure 3.13 and replace the keyword "child-friendly" with the keyword "free cancellation". The adapted query is represented as complex concept description in DL as follows :

$$(\texttt{Hotel} \sqcap \exists\, offers.(\{\texttt{STEAMBATH}\} \sqcap \texttt{Facility})) \sqcup$$
$$(\texttt{Hotel} \sqcap \exists\, located.(\{\texttt{MAYRHOFEN}\} \sqcap \texttt{Location}) \sqcup$$
$$(\texttt{Hotel} \sqcap \exists\, describedAs.(\{\text{``}\texttt{FREE CANCELLATION}\text{''}\} \sqcap \texttt{Keyword})$$

Therefore the user is looking for hotels located in Mayrhofen, which are equipped with a steam bath and offer free cancellation. To provide users with additional information and increase recall the query expansion mechanism is used. In essence the query is expanded to :

$$((\texttt{Hotel} \sqcup \texttt{GuestHouse} \sqcup \texttt{Farm} \sqcup \texttt{YouthHostel} \sqcup \texttt{Appartment}) \sqcap \exists\, offers.(\{\texttt{STEAMBATH}\} \sqcap \texttt{Facility})) \sqcup ((\texttt{Hotel} \sqcup \texttt{GuestHouse} \sqcup \texttt{Farm} \sqcup \texttt{YouthHostel} \sqcup \texttt{Appartment}) \sqcap \exists\, located.((\{\texttt{MAYRHOFEN}\} \sqcap \texttt{Location}) \sqcup \texttt{Location})) \sqcup ((\texttt{Hotel} \sqcup \texttt{GuestHouse} \sqcup \texttt{Farm} \sqcup \texttt{YouthHostel} \sqcup \texttt{Appartment}) \sqcap \exists\, describedAs.(\{\text{``}\texttt{FREE CANCELLATION}\text{''}\} \sqcap \texttt{Keyword}))$$

The result is depicted in Figure 3.15. Results are ranked with respect to the user query. Therefore, the top ranked Web Sites are those that mention hotels which are equipped with a steam bath, located in Mayrhofen and offer free cancellation. These are followed by Web Sites which mention hotels that are equipped with a steam bath and are located in Mayrhofen but do not offer free cancellation, followed by Web Sites that mention other accommodations such as guesthouses, farms, apartments that match the criteria and offer free cancellation. Subsequently, Web Sites are listed that mention hotels offering free cancellation which are located in Mayrhofen and do not offer a steam bath, followed by hotels not offering free cancellation which are located anywhere and offer a steam bath etc. Therefore, the results are ranked from the most specific to the least specific.

Every result entry consists of the title of the Web page, a link to the Web page, the first two lines of the summary, a link to the cached page, a link to a list of all concepts and instances that are mentioned on the page and a drop down element that shows the complete summary. The summary shows all annotations on the page with their context. The annotations are colored in the same color as the corresponding concept in the query, or gray in case the concept is not part of the query. Therefore, the instance “Elisabethhotel” which is mentioned in one of the top ranked documents is colored in the same color as the concept `Hotel` and “Mayrhofen” is colored in the same color as `BroadLocation` in the query.

3.7 Evaluation & Deployment

In the following we will discuss our findings to the research questions defined in Section 1.2.1. Every research question is addressed in a separate subsection.

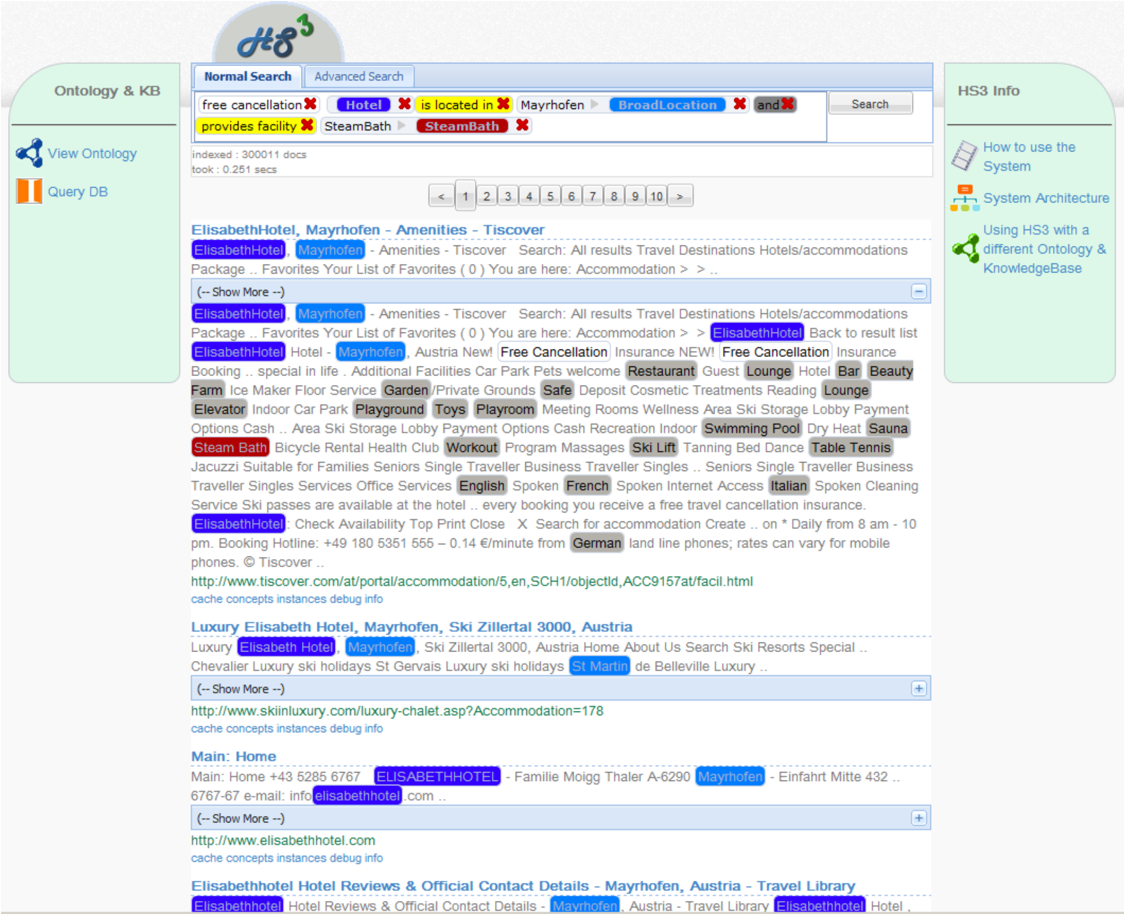

Figure 3.15: Search result

3.7.1 HS^3 performance evaluation on tourism datasets of different size

To answer ***research question 6***, asking how scalable a system that uses a hybrid search approach is with respect to the number of triples and the count of documents it can handle, we used the tourism KB to automatically generate three combined indices with HS^3, defined reference queries of different complexity and ran them against HS^3. The results are presented in the following. HS^3 was instructed to automatically fetch 3,000, 30,000 and 300,000 documents off the World Wide Web. These three different datasets were annotated and a combined index was created. Furthermore, seven queries of different complexity were defined. The queries are listed in Table 3.1.

Every query was executed 1000 times on the corresponding index and the average execution time was calculated. We used a commodity notebook with a 2 Gigahertz Dual Core Processor and 4 GB of RAM to run the performance tests. The average execution

Query type	Query
keyword only	$(\{\texttt{ELISABETHHOTEL}\} \sqcap \texttt{Keyword})$
concept simple	$\texttt{Hotel}$
concept simple & keyword	$\texttt{Hotel} \sqcap \exists$ describedAs.$(\{\texttt{CHILD-FRIENDLY}\} \sqcap \texttt{Keyword})$
concept standard	$\texttt{Hotel} \sqcap \exists$ locatedIn.$(\{\texttt{MAYRHOFEN}\} \sqcap \texttt{Location})$
concept standard & keyword	$(\texttt{Hotel} \sqcap \exists$ locatedIn.$(\{\texttt{MAYRHOFEN}\} \sqcap \texttt{Location}))$ $\sqcup (\texttt{Hotel} \sqcap \exists$ describedAs.$(\{\texttt{CHILD-FRIENDLY}\} \sqcap \texttt{Keyword}))$
concept complex	$(\texttt{Hotel} \sqcap \exists$ locatedIn.$(\{\texttt{MAYRHOFEN}\} \sqcap \texttt{Location}))$ $\sqcup (\texttt{Hotel} \sqcap \exists$ offers.$(\{\texttt{STEAMBATH}\} \sqcap \texttt{Facility}))$
concept complex & keyword	$(\texttt{Hotel} \sqcap \exists$ locatedIn.$(\{\texttt{MAYRHOFEN}\} \sqcap \texttt{Location}))$ $\sqcup (\texttt{Hotel} \sqcap \exists$ offers.$(\{\texttt{STEAMBATH}\} \sqcap \texttt{Facility}))$ $\sqcup (\texttt{Hotel} \sqcap$ describedAs.$(\{\texttt{CHILD-FRIENDLY}\} \sqcap \texttt{Keyword}))$

Table 3.1: Queries for evaluation

time in seconds for every dataset and query type is listed in Table 3.2. It is evident that the keyword only query has the least execution time regardless of the document count. Concept only search queries are more complex than keyword only queries and result in higher query times. However, they still are in the range of one second for big datasets. Furthermore, it can be observed that the execution time increases only marginally between a simple concept and complex concept query no matter how many documents have been indexed. This also holds true for the combined search, including keywords and concepts. Compared to the execution times of concept only search queries, the execution times of combined queries only increase marginally. Due to a bug in the scoring mechanism of SIREn we had to use a workaround which resulted in more complex queries and higher execution times. Hence, lower execution times can be realized when the bug gets fixed.

3.7.2 HS^3 and SESAME performance comparison

To answer ***Research Question 7***, asking how the query-performance of a system that indexes semi-structured data, relates to the one of a pure Triple Store such as SESAME [17] holding the same data, we used the reference queries defined in Section 3.7 and

Query \ Doc Count	3K	30K	300K
keyword only	0.018	0.023	0.018
concept simple	0.028	0.105	0.909
concept simple & keyword	0.036	0.131	1.046
concept standard	0.030	0.100	0.826
concept standard & keyword	0.052	0.126	0.929
concept complex	0.046	0.118	0.857
concept complex & keyword	0.066	0.153	0.995

Table 3.2: HS^3 performance in seconds per index document count and query type

ran them against SESAME. However, since SESAME does not support keyword search and combined queries, only the pure concept queries were formulated in SeRQL and ran against SESAME. The results are presented in the following.

To accurately compare the two rather different systems in terms of performance we ensured that both systems operate on approximately the same triples and that the Triple Store mimics the ranking functionality of HS^3. Unfortunately, it was not possible to leverage the `order by` construct of the RDF query language SeRQL to mimic HS^3 ranking functionality, because Sub-Queries would be needed in addition to imitate the desired functionality. Neither SeRQL nor SPARQL support Sub-Queries. The only possibility to mimic HS^3's ranking functionality is to perform explicit ranking by combining variations of the original query via UNION operators in the SeRQL query. Listing 3.1 depicts the SeRQL search query for hotels and similar accommodations by using variations of the original query and UNION operators. This query is constructed such that it mimics the ranking functionality of HS^3. Hence, it returns the triples in the same order in which HS^3 returns the documents containing references to the entities described by the triples when the query "`Hotel`" is issued. HS^3 expands the query "`Hotel`" automatically to include similar concepts that are of interest to the user and performs the ranking implicitly via weights to provide the same result ranking as the rather cumbersome SeRQL query. Hence, HS^3 returns those documents that contain information about hotels first followed by those documents that contain information about apartments, farms, youth hostels, guesthouses or bed and breakfast accommodations.

```
SELECT A FROM
  {A} rdf:type {eTourism:Hotel} LIMIT 100
  UNION
SELECT A FROM
```

```
  {A} rdf:type {eTourism:Apartment} LIMIT 100
  UNION
SELECT A FROM
  {A} rdf:type {eTourism:Farm} LIMIT 100
  UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel} LIMIT 100
  UNION
SELECT A FROM
  {A} rdf:type {eTourism:GuestHouse} LIMIT 100
  UNION
SELECT A FROM
  {A} rdf:type {eTourism:BedAndBreakfast} LIMIT 100
USING NAMESPACE
  rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
  eTourism = <http://tuwien.ac.at/rdf/itchyfeet#>
```

Listing 3.1: SeRQL Query *Concept Simple*

To compare the performance of HS^3 with the performance of a Triple Store we used only the concept based queries listed in Table 3.1 because Triple Stores are not capable of performing hybrid semantic searches. The SeRQL representation of the “concept simple” query, mimicking HS^3 ranking functionality, is depicted in Listing 3.1. The complete listings of all SeRQL queries are available in the Appendix B. The “concept standard” query represented in SeRQL contains the main query for a hotel in Mayrhofen and variations of the query to retrieve other accommodation types in Mayrhofen concatenated via the UNION operator. These queries are followed by additional queries for hotels located in any location and accommodations of other types located in any location concatenated via the UNION operator as well. The “concept complex” query extends the “concept standard” query by querying for those hotels in Mayrhofen that also offer a steam bath. This query transformed into SeRQL and mimicking HS^3 functionality by including all possible variations that may be of interest to the user results in a query with 24 UNION operators. Unfortunately, it is not possible to limit the overall result to a specific value because the LIMIT operator can only be applied on a per-SELECT basis put not to the entire SeRQL statement. We limited every query connected via the UNION operator to 100 results, because without this restriction response times of the Triple Store would have been very high. In addition, we included a second “concept complex” query, which is not listed in Table 3.1 for a more accurate performance comparison. This query asks for a hotel that is located in an arbitrary location where English is spoken and that offers a steam bath. This query transformed into SeRQL results also in a statement with

24 UNION operators.

Table 3.3 depicts the performance results of all queries executed on HS^3, a SESAME Native repository and a SESAME SwiftOWLIM repository. We used repositories of different size, whereas the small one held only 45 thousand triples, the medium one 450 thousand triples and the big one 4.5 million triples. In case of HS^3 we ensured that the corresponding amount of triples was encoded in the small, medium and big index to conduct a fair performance comparison. The SESAME Native repository makes only use of the RDF Direct Type Hierarchy whereas the SwiftOWLIM repository uses an inferencer that implements the OWL Horst dialect. Hence, the SwiftOWLIM repository can handle more advanced queries due to the inferred information. However, it was not possible to load the full 4.5 million triple set into the SwiftOWLIM repository, because the server got stuck after several hours of performing inferencing. We were only able to load half of the 4.5 million triple set into the SwiftOWLIM repository.

Query Type / System	Concept Simple	Concept Standard	Concept Complex	Concept Complex 2
HS^3 45K Triples	0.028	0.030	0.046	0.212
HS^3 450K Triples	0.105	0.100	0.118	1.032
HS^3 4.5M Triples	0.909	0.826	0.857	3.730
SESAME Native 45K Triples	0.017	0.165	0.222	0.333
SESAME Native 450K Triples	0.021	0.666	1.147	2.806
SESAME Native 4.5M Triples	0.021	0.761	3.153	13.332
SESAME SwiftOWLIM 45K Triples	0.013	0.034	0.093	0.158
SESAME SwiftOWLIM 450K Triples	0.012	0.221	0.919	1.149
SESAME SwiftOWLIM 2.25M Triples	0.007	0.215	0.987	12.412

Table 3.3: HS^3 performance in seconds per triples count and query type

It can be seen that HS^3 outperforms the Triple Store in the majority of query runs. The Triple Store is faster when it comes to simple queries where only little effort is needed to provide the user with a result ranked by relevance. The more triples and the complexer the query gets the better HS^3 is able to play its advantages off against the Triple Store. HS^3 also holds the document index in the combined index and therefore operates on a bigger dataset. This means that additional information can be leveraged for the search process resulting in higher recall (see Section 3.7 for details) compared to the Triple Store. However, the index used by SESAME is slicker and optimized for triples, leading to better performance when simple or standard queries are used and no ranking is needed. Yet, the bigger the repository becomes and the more results are retrieved the more important gets ranking. It can be observed that the "Concept Simple" query issued against the repository that uses SwiftOWLIM uses less time the bigger the amount of triples in the repository gets. This might be a special case that occurs because of caching.

In conclusion, both systems can operate with a big amount of data and provide results within milliseconds, but have different areas of application where they can prove their superiority.

3.7.3 User evaluation

To answer ***Research Question 8***, asking how the novel interactive ontology-aware keyword-based input mechanism holds up against the traditional keyword-based input mechanism in terms of usability for the average Internet user, we created a combined index based on the tourism KB and one based on the KIM Ontology & KB and conducted an usability evaluation of the new input mechanism. The results are presented in the following.

The purposes of the evaluation were to determine whether average Internet users are able to efficiently formulate semantic queries, without prior knowledge of the ontology and to determine what attitude they have toward this novel input mechanism compared to the traditional keyword-based input mechanism. We structured our evaluation into 5 sections. In the first section probands were asked to complete a Pre-Questionnaire, which was used to determine their Internet and search engine usage and to test their knowledge about the Semantic Web in general. This section was followed by a short description of the differences among concepts, instances of concepts and keywords in the context of the Semantic Web. Furthermore, a short introduction on how to use the interface was given by means of an example. The two main sections asked the probands to formulate different search queries, which were formulated in natural language, with the interactive ontology-aware keyword-based input mechanism. The first main section contained five queries from the tourism domain based on the tourism ontology and the second main section held five queries from the news and broadcast domain based on the KIM Ontology. The last question asked the users to formulate an arbitrary search query that involves an arbitrary number of concepts and instances of the concept `Person` and `Organization` or its sub-concepts. This was followed by a text box where users were asked to formulate their search intent in natural language to determine whether a user's search intent was reflected in the constructed query. In the last section probands were asked to complete a Post-Questionnaire, which was used to determine their general attitude towards this new form of search query creation. Table 3.4 depicts the queries users were asked to formulate and the percentage of users that rated the query formulation either very easy, easy, OK, hard or very hard.

Twenty-two test persons participated in the user evaluation. They were between 22 and 52 years old and moderate to regular Internet users. About 73% were male and 27% were female. They worked either in consulting, customer service, health care, management, logistics, research, software development or were enrolled as students at a university. All except two probands stated Google as their preferred search engine. Furthermore, the majority of test persons stated that they use between 3 and 4 terms for their search queries on average. About 77% of the probands have heard of the Semantic Web and about 59% knew what ontologies are. However, more than half of all probands did not know the difference between concept, instance of concept and keyword. Still, the majority of test persons was able to use the interface intuitively without knowing the difference. It can be seen from Table 3.4 that the majority of probands did not have any difficulties formulating the search queries with the interface and stated that query formulation was either very easy, easy or OK. It is noticeable that those queries, namely 5. and 6., which asked the user to use a keyword instead of an instance or concept to extend the query have been perceived as hard or very hard by a few test persons. We think that this relates to the problem that not all probands understood the difference between a keyword or term on a Web page and an annotation that refers to a concept, instance or both in the ontology and KB. Hence, the actual meaning of keyword, concept and instance in the context of the Semantic Web and World Wide Web should have been explained more comprehensively, to provide these probands with the big picture. However, asked about the difficulty to formulate queries with the interface in general, about 9.1% stated that it is very easy, about 40.9% stated that it is easy and about 50% that it is OK. The last question, which asked the probands to formulate an arbitrary search query with the given ontology showed that people had a good understanding of the ontology even though they had no prior knowledge of it. Most of the probands formulated a rather complex query involving different concepts, instances, properties and keywords. To check whether they actually understood what they have formulated, we asked them to formulate the same query in natural language. Various test persons stated that they liked the clean and tidy

No.	Query	Ontology	very easy	easy	ok	hard	very hard
1.	apartment	e-Tourism	**77.3%**	18.2%	0%	4.5%	0%
2.	hotel located in Mayrhofen	e-Tourism	**54.5%**	36.4%	9.1%	0%	0%
3.	farm located in location where English is spoken	e-Tourism	27.3%	**59.1%**	13.6%	0%	0%
4.	hotel that offers a telephone service and is located in Retz	e-Tourism	27.3%	**36.4%**	31.8%	4.5%	0%
5.	quiet farm where English is spoken that has a garden	e-Tourism	9.1%	22.7%	**40.9%**	18.2%	9.1%
6.	important person	KIM	**31.8%**	13.6%	22.7%	22.7%	9.1%
7.	person who holds the position CEO in organization Apple	KIM	22.7%	**40.9%**	31.8%	4.6%	0%
8.	bank that is owned by a person whose relative is Donald Harris	KIM	22.7%	**45.5%**	27.3%	4.6%	0%
9.	company that is a parent organization of Cisco System and partially controls the company Neogen	KIM	22.7%	**36.4%**	31.8%	9.1%	0%

Table 3.4: Query formulation with the interactive ontology-aware keyword-based input mechanism and its perceived difficulty

interface as well as the fast retrieval of suggestions. Furthermore, the probands stated that the different colors of the concepts in the graphical query were helpful during the query generation. All test persons stated that the concept, instance and property suggestion mechanism was either helpful or very helpful. A few probands criticized that it was difficult to edit the graphical query once defined, because the editing box did not appear instantly or not at all. This issue is related to the used Internet Browser and will be fixed in a future release.

3.7.4 Precision and recall evaluation

To answer ***research question 9***, asking whether precision and recall of the hybrid search approach is superior to precision and recall of the keyword-only search approach, we compared precision and recall of searches conducted on a combined index and the ones of searches conducted on a keyword-based index comprising the same documents. The results are presented in the following.

The difficulty of evaluating precision and recall for Semantic Search Systems and Hybrid Search Systems arises from the fact that no public test-datasets are available that can be used for an unbiased performance comparison among competing systems. For the evaluation of precision and recall of traditional IR systems the widely accepted TREC dataset, which includes a document collection, a set of queries and judgments representing the ground truth, can be used. However, the TREC dataset is only of limited use to compare Semantic Search Systems because a related and publicly available ontology and KB as well as a corresponding ground truth would be needed. Even though precision and recall can be evaluated by using a system's primary dataset, the precision and recall results are not directly comparable to the results of other Semantic Search Systems as long as they use different datasets. For an accurate and unbiased comparison the systems that are compared need to operate on the same document corpus, the same ontology and KB, use the same queries and need to have the same relevance judgments (ground truth) for documents. Even though there are some publicly available ontologies such as the KIM Ontology or the YAGO Ontology, either a publicly available document corpus or corresponding ground truth data is missing.

However, to be able to evaluate precision and recall of HS^3 we decided to make use of two different datasets. The first dataset, henceforth named the e-Tourism dataset, was created by using the Harmonise Ontology and the KB which was created from the Tiscover database. We instructed HS^3 to automatically fetch documents that contain information about tourism accommodation and destinations from the World Wide Web. To actually compare precision, recall and the overall performance of HS^3 to other Hybrid

Search Systems we had to ensure that we use a publicly available ontology, KB and document corpus that is used by at least one other Hybrid Search System for the second dataset. Furthermore, we had to ensure that the systems we compare are able to cope with similar type of queries. We chose the system presented by [21], because they make use of the publicly available KIM Ontology and KB and their system can cope with queries of similar type. The authors also linked to a document corpus that consists of 145,316 news documents, which they have used for the precision and recall evaluation of their approach. Unfortunately, the referenced news document corpus is not available anymore. Therefore, we instructed HS^3 to automatically build a corpus containing news documents. For this purpose we had to implement a new Metadata Fetcher plug-in, because the Yahoo BOSS Service we previously used for fetching related data is not free of charge anymore. The new Metadata Fetcher plug-in uses the Microsoft Bing news service. As Bing only returns documents that contain news within a certain date range, HS^3 were able to fetch only about 167,029 unique documents. However, the generated corpus exceeds the news corpus used by [21] in size and should be sufficient to compare the performance of both systems.

Another reason for choosing two datasets was that the ground truth for a small dataset can be determined more accurately than the ground truth for a large dataset. To generate an accurate ground truth for a large dataset a multitude of documents would need to be analyzed manually regarding their relevance for multiple queries. This is somewhat impossible for big datasets and therefore different automatic or semi-automatic algorithms need to be used to identify relevant documents in this datasets to get an approximate ground truth. However, for reasonably small datasets the ground truth can be assessed manually. Therefore, we choose a small dataset, namely the e-Tourism dataset and a big dataset, namely the news dataset, and took the average of both precision and recall evaluations to get a more accurate prediction of the overall precision and recall of HS^3. The ground truth for the small dataset was generated manually, whereas the ground truth for the large dataset was generated by using manual checking and semi-automatic approaches to identify relevant documents per query. We defined 18 queries for the e-Tourism dataset and the news dataset in total. A complete listing of all queries is available in Table 3.5. To create an approximate ground truth for the large news dataset, we implemented a plug-in for HS^3 that loads all documents into Apache Solr[25]. Subsequently, we narrowed the relevant documents per query by issuing several manually created keyword-based search queries at Apache Solr and saved the results as document sets. These documents sets were checked manually for relevance to get an approximate ground truth.

[25] http://lucene.apache.org/solr/

No.	Query	Ontology
1.	Give me information about hotel Panhans	e-Tourism
2.	Give me information about bed and breakfast accommodations where I can play squash	e-Tourism
3.	Give me information about guesthouses that have a whirlpool	e-Tourism
4.	Give me information about youth hostels that are equipped with an air condition	e-Tourism
5.	Give me information about hotels that are located in Mattersburg	e-Tourism
6.	Give me information about locations where youth hostels are located	e-Tourism
7.	Give me information about holiday flats in Mayrhofen that are equipped with a sauna	e-Tourism
8.	Give me information about hotels in Salzburg that are equipped with a whirlpool	e-Tourism
9.	Give me information about guesthouses in Mayrhofen that are located near a mountain	e-Tourism
10.	Give me information about recommended bed and breakfast accommodations in Neustift that have a garden	e-Tourism
11.	Give me news about persons that work at Walt Disney	News (KIM)
12.	Give me news about the person Oliver Kahn	News (KIM)
13.	Give me news about companies that are located in Chile	News (KIM)
14.	Give me news about companies in the retail industry that are located in Chile	News (KIM)
15.	Give me news about organizations in the retail industry that are traded on the New York Stock Exchange	News (KIM)
18.	Give me news about organizations in the financial services industry that are located in Japan and traded on the OTC Stock Exchange	News (KIM)
17.	Give me news about organizations that are located on the Cayman Islands and are traded on the NASDAQ	News (KIM)
18.	Give me news about market research reports of organizations that are located on the Bermudas	News (KIM)

Table 3.5: Evaluation queries for the e-Tourism and News (KIM) Ontology

In the following we present the precision and recall results that have been acquired by using the e-Tourism dataset and the news dataset. Furthermore, we present performance results for the queries executed on the news dataset and compare them to the performance results stated by [21]. We ensured that a commodity PC, which was commonly used back in 2007, is used for the performance evaluation, so a performance gain cannot be attributed to the higher processing power of current commodity PCs. We will discuss the two most interesting queries and results of every query set in more detail.

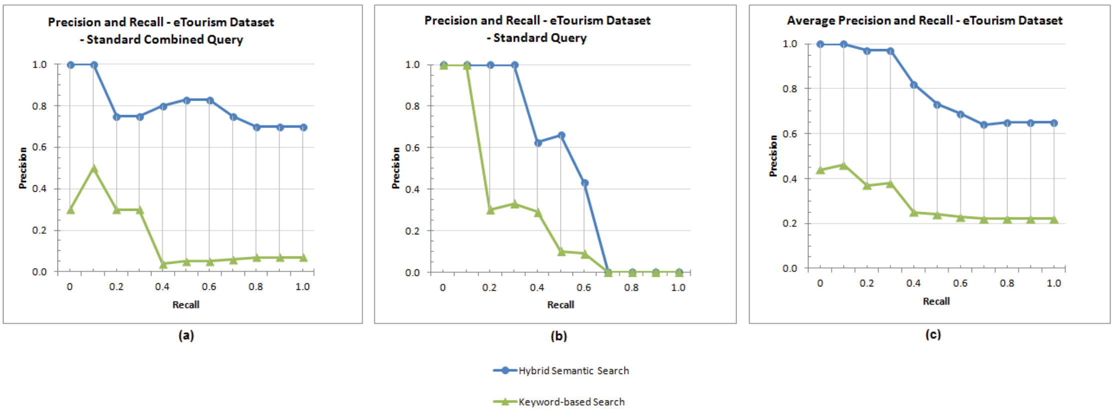

Figure 3.16: Precision and recall e-Tourism dataset

Figure 3.16 depicts the average precision and recall curve of the e-Tourism dataset (c). Furthermore, the precision and recall curves (a, b) of two queries out of the set of 10 queries are depicted and discussed in the following.

e-Tourism Dataset : Standard Combined Query (a). The standard combined query executed on the e-Tourism dataset encodes the information need "reviews about guesthouses that are located in Mayrhofen and that are nearby a mountain". Whereas the ontology does not contain the concept `Mountain` and therefore the hybrid semantic search of HS^3 leverages the contextual and full text information in the combined index as complement to retrieve documents that are relevant. The hybrid semantic search of HS^3 maintains a high precision for the given information need, because most guesthouses in Mayrhofen have been correctly annotated by the Annotator. Some have been incorrectly annotated and others were missed due to mutated vowel in their name. However, due to the combined approach the documents mentioning guesthouses which were missed by the annotator are still returned because they contain the terms "guesthouse", "mayrhofen" and "mountain" in the full text section of the combined index. The keyword-based search delivers poor results for higher recall values. The reason is that the keyword-based search is not able to retrieve most of the documents mentioning guesthouses, because they are mentioned with their actual name such as “Länderhof” or “Eckartauerhof”. However, since the query also includes the terms “guesthouse”, “mountain” and “mayrhofen” the keyword-based approach can still leverage this information to return relevant documents.

e-Tourism Dataset : Standard Query (b). The standard query executed on the e-Tourism dataset encodes the information need “reviews about guesthouses that are located in Salzburg and that have a whirlpool”. The hybrid semantic search provides not all relevant results because some guesthouses matching the criteria have not been annotated due to a mutated vowel as part of their name or encoding problems in the fetched document. However, those documents that were correctly annotated and contain guesthouses matching the criteria are returned. Even documents that are relevant but do not hold information about the facilities of a guesthouse are returned among the top ones in case the KB states that the specific guesthouse has a whirlpool. As most documents mention guesthouses with their actual name or do not contain the term whirlpool the keyword based approach returns only those documents that contain the terms “guesthouse”, “whirlpool” and “salzburg”, which explains the high precision values at low recall levels. However, in this scenario the hybrid semantic search approach cannot benefit from the full text information in the index to still return all relevant documents.

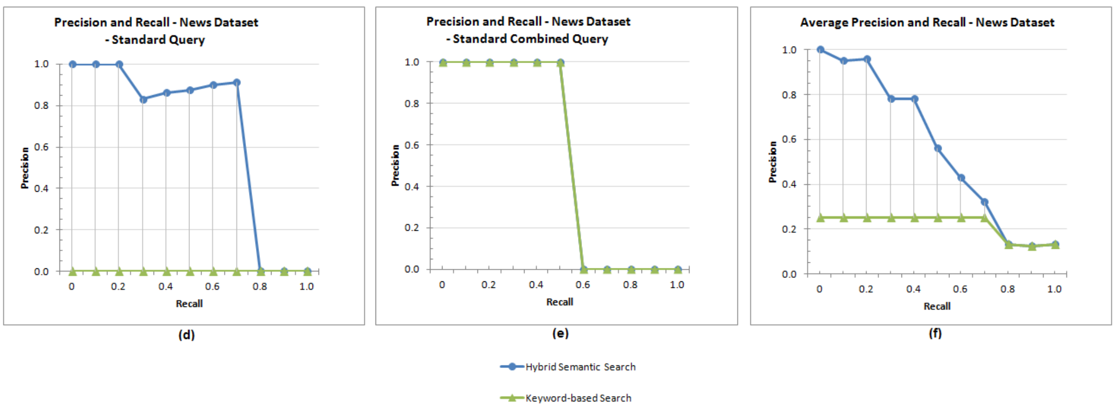

Figure 3.17: Precision and Recall News Dataset

Figure 3.17 depicts the average precision and recall curves of the news dataset (g). Furthermore, the precision and recall curves (d, e) of two queries out of a set of 8 queries are depicted and discussed in the following.

News Dataset : Standard Query (d). The standard query executed on the news dataset encodes the information need “news about organizations that are traded on NASDAQ and that are located on the Cayman Islands". The corresponding precision and recall curve shows that the hybrid semantic search approach is superior for this type of information need. The keyword-based search retrieves no relevant documents. The keyword-based approach will return documents that contain the terms "NASDAQ", "organization" or "Cayman Islands". However, these documents are not relevant. The relevant documents in the corpus do not contain any of these terms. The relevant documents just contain the name of the organization such as "Garmin Ltd". However, the keyword-based approach returns documents mentioning companies that are traded on NASDAQ because both terms occur in the document, but these companies are not located on the Cayman Islands. The hybrid semantic search approach returns relevant documents because the combined index holds information such as that the annotation of the phrase"Garmin Ltd" in a specific document refers to the public company "Garmin Ltd", which is located on the Cayman Islands and traded on NASDAQ. Still, some documents are missed because the information of some companies is not stored in the KB.

News Dataset : Standard Combined Query (e). The standard combined query executed on the news dataset contains the information need "news about market research reports from organizations that are located in Bermuda". The ontology does not contain the concept `Market Research Report` and therefore the hybrid search of HS^3 leverages the contextual and full text information in the combined index as complement to retrieve documents that are relevant. It can be seen that the hybrid search approach of HS^3 returns relevant documents at low recall. However, since not all companies that are located in Bermuda are contained in the KB the precision drops at higher recall. The keyword-based approach on the other hand benefits from the fact that several relevant documents contain the phrase "market research report" and "Bermuda". In addition, documents that contain the phrase "market research report" contain mainly information about specific organizations and eliminate the need for the term "organization" to be part of the document. Hence, also relevant documents that mention organizations with their actual names are returned. In this specific case the keyword-based approach has the same precision and recall as the hybrid semantic search approach. However, if the information of all companies located in Bermuda would be stored in the KB, the hybrid search approach of HS^3 could still maintain a high precision at higher recall rates.

The average precision and recall curve of the e-Tourism dataset (Figure 3.16 (c)) and the news dataset (Figure 3.17 (f)) shows that the hybrid semantic search approach is in general superior to the keyword-based approach. The high precision of the hybrid search approach on the e-Tourism dataset, results from the combined search approach and the sophisticated annotation and extraction rules that have been specifically tailored

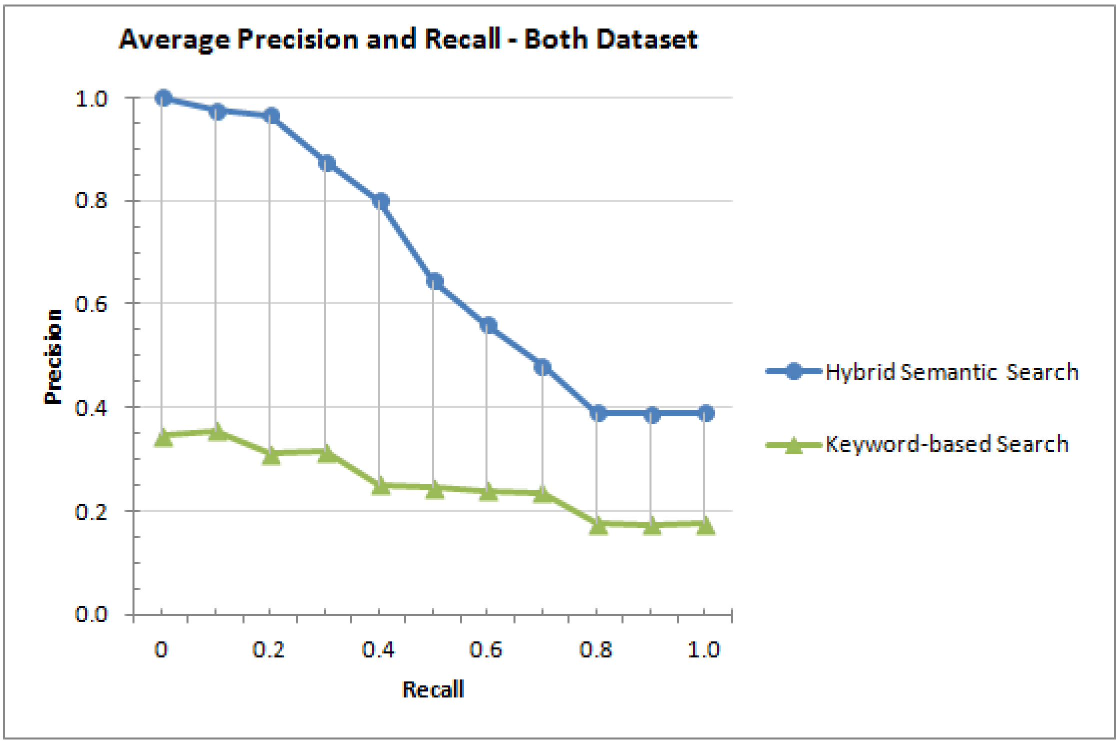

Figure 3.18: Average Precision and Recall of e-Tourism and News Dataset

to the tourism domain. Figure 3.18 depicts the average precision and recall curve that is retrieved by combining the average precision and recall curve of the e-Tourism and news dataset. Comparing the precision and recall curve of the news dataset to the average precision and recall curve presented by [21], which was also determined by using the KIM Ontology, the corresponding KB and a news document corpus as dataset, it can be seen that HS^3's approach can compete with their approach in terms of precision and recall. However, the distinction between the two systems is that HS^3 needs at most 0.618 seconds and on average only 0.218 seconds to return the result for a query on a corpus of 167,029 annotated news documents, but the approach presented in [21] needs at most about 30 seconds on a corpus of 145,316 annotated news documents. However, as discussed earlier, the precision and recall comparison should be treaded with caution, because even though the systems use the same ontology and KB, the content of the document corpus used for the evaluation is different and not exactly the same queries and relevance judgments (ground truth) were used for both systems. Furthermore, the accuracy of the ground truth of the news corpus is not as accurate as the one used for the e-Tourism corpus because semi-automatic approaches have been used to create it. For an accurate comparison the same publicly available document corpus, ground truth and queries need to be used.

3.7.5 How to apply HS^3 to a different domain

To answer ***Research Question 10***, asking what trade-offs need to be made when a hybrid search system is implemented that should be applicable for several domains (ontologies), we documented the steps that need to be carried out to apply our approach to an arbitrary domain (ontology) and the resulting trade-offs. In the following the process of adapting HS^3 to an arbitrary domain (ontology) and resulting trade-offs are described.

HS^3 has been designed to work with an arbitrary OWL-based ontology and KB. In this section we discuss the steps that need to be made to apply HS^3 to a different domain. In case the initial knowledge is not available in a data format suitable for loading it into the Triple Store, it needs to be transformed with the Transformation Engine. The Transformation Engine has an RDBMS and text / HTML file plug-in. If a different type of data source is used, a custom plug-in needs to be implemented. For the Transformation Engine to transform a custom data format into structured data, a transformation ruleset needs to be defined. Therefore, the prerequisite is that the initial data is either already stored in the KB or loaded into the KB via the Transformation Engine. Next, the Knowledge Engineer needs to define the *Realms* and the *Index Graphs* (see Section 3.4.1 for details) with respect to the target audiences' information need and the used ontology. Next, the Annotation component and Concept & Instance Suggestor need to be equipped with textual descriptions of concepts and instances. Descriptions can be either attached via the properties `prefTextualRepresentation` and `altTextualRepresentation` or by the default `rdfs:label` property to concepts and instances.

When all prerequisites are met, HS^3 can be started. On the first startup the initial suggestion index for the Concept & Instance Suggestor is build automatically. Subsequently, Metadata Fetchers and Data Fetchers are started and fetch related data off the World Wide Web and put it as work packages into the work queues. Annotators pick up the work packages, annotate the included Web pages via their annotation pipelines and put the work packages back into the work queue. Indexers pick up these work packages, analyze the annotations of the documents and update the combined index. In parallel HS^3 can already process user search requests by using the current combined index. All Fetchers, Annotators and Indexer work iteratively to keep the system's information up to date. To showcase how to apply HS^3 to a different domain we used the KIM Ontology (KIMO) [77]. A minor extension of the standard mechanism was needed, because the KIM Ontology uses the properties `hasMainAlias` and `hasAlias` to refer to the textual representations of instances and concepts.

However, to ensure HS^3 applicability to several domains (ontologies) a couple of trade-offs had to be made. First, even though it is possible to use a general approach to select triples of the Triple Store for inclusion in the combined index, the index gets very big and holds a lot of information that is of no relevance to the end-user. Hence, the better approach is to use so-called *Index Graphs*, introduced in Section 3.4, that are patterns describing relevant structures and information for a specific domain in terms of concepts and their relations. However, the trade-off to the general approach is that a domain expert is needed who defines the *Index Graphs* ones. Still, the gain in query performance due to the less crowded combined index and the higher search accuracy due to the usage of relevant patterns more than compensates the additional effort.

Second, identifying concepts and instances of concepts in documents is highly domain depended. For example to identify concepts of the tourism domain such as `Hotel`, `Location` or `Attraction` different rulesets are needed compared to the identification of concepts that belong to the news domain such as `Person`, `Company` and `Political Party`. Even though it is possible to define low-level rules that cover a great deal of concepts, these are not very accurate and miss a lot of concept and instance occurrences in documents. Hence, to accurately annotate a document corpus, more specific rulesets need to be defined.

Finally, occurrences of concepts and instances in documents do not automatically provide information about the relevancy of the document. For example there might be multiple documents that mention the instance “Hotel Elisabeth”, but 80 percent of them could be just documents containing mere listings of hotels with no additional information. It would be better to identify those documents which hold information about a specific instance. For this purpose we introduced so-called Concept Identification Graphs (CIG), described in Section 3.4, that define related concepts and instances that must or should be mentioned within the specific document to be of relevancy. According to these a Relevance Score for the instance and the corresponding document is calculated. However, the CIGs need to be defined by domain experts leading to additional effort. Hence it is a trade-off between annotation accuracy and the effort that is put into the creation of the CIGs.

Overall it took about 20 hours to apply HS^3 to the news domain. We had to create an additional plug-in for the Metadata Fetcher to use the Microsoft Bing News Service[26] and adapt the Annotator to use the properties `hasMainAlias` and `hasAlias` of the KIM Ontology to retrieve the textual representation of instances and concepts. The Annotator used the standard UIMA Concept Mapper to identify concepts and instances of concepts

[26] http://msdn.microsoft.com/en-us/library/dd250884.aspx

within documents. However, to get more accurate annotation results, additional annotation rules would need to be created, leading to a higher adaptation effort. The definition of Index Graphs, CIGs and Realms took about 4 hours.

3.8 Summary

We have proposed an approach to bridge structured and unstructured data by using a combined index to facilitate hybrid semantic searches. An overview of state-of-the art semantic and Hybrid Search Systems was given, followed by a description of their shortcomings and the advantage of a hybrid semantic search approach. We started with a formal definition of hybrid semantic search by defining the different query types via concept descriptions in DL. This was followed by a formal definition of the document structure used in the combined index and the result structure. Furthermore, we introduced a novel input mechanism for hybrid semantic search that combines the clean and concise input mechanisms of keyword-based search engines with the expressiveness of the input mechanisms provided by Semantic Search Engines. The interface amalgamates concept-based and keyword-based input. By using this interface users can formulate queries without prior knowledge of the underlying ontology. Hybrid semantic search based on the proposed combined index structure and the novel input mechanism have been implemented in the Hybrid Semantic Search System HS^3. We gave an overview of HS^3's architecture, its components and an in-detail presentation of the Search & Ranking components. In addition the mechanism used by HS^3 to automatically fetch relevant data from the World Wide Web has been described.

We discussed *Index Graphs*, which are used to encode information of the ontology and KB, that can be used during the retrieval process. Furthermore, *Concept Identification Graphs* have been presented, which are defined via RDF Reification and are used to identify and disambiguate textual representations of instances in Web documents. We applied HS^3 to the tourism and news domain, presented a showcase and performance test results. We used keyword-based, concept-based and hybrid queries of different complexity to evaluate HS^3's performance. The performance evaluation showed that response times of the system are within a second even for big datasets and complex queries consisting of a combination of concepts, instances and keywords. Furthermore, the performance tests showed that only a marginal difference between simple queries and complex queries in terms of response time exists. In addition, we conducted a precision and recall evaluation and presented the results.

The interactive ontology-aware keyword-based input mechanism has been showcased using the tourism dataset and the KIM Ontology and KB. To evaluate the usability of the novel input mechanism we conducted a user evaluation to ascertain the acceptance of the input mechanism, whether it is easy or hard to formulate queries of different complexity and if it is possible to formulate queries without prior knowledge of the used ontology. The user evaluation indicated that average Internet users are able to use the interface without any difficulties and that they deem the concept, instance and property suggestion mechanism as very helpful. The test persons liked the clean and concise interface as well as the fast retrieval of suggestions. Furthermore, the evaluation showed that the probands were able to formulate expressive queries involving concepts, instances, properties and keywords without prior knowledge of the ontology.

CHAPTER 4

Conclusion & Outlook

The work presented in this thesis focused on providing solutions to challenges that arise when offering/deploying Business Services.

First, to efficiently conduct business electronically the participating parties need suitable platforms. In our work we concentrated on Electronic Marketplaces, which are the prevalent platform type for doing business in the B2B, B2C and C2C domain. Electronic Marketplaces efficiently regulate and support the exchange of goods and services. They offer the functional and technical capabilities to facilitate trade between a multitude of participants. To participate in an Electronic Market, organizations need to adapt their internal and external business processes or create new ones. However, the more complex these processes become, the more alignment and regulation effort is needed to ensure an efficient and secure exchange of goods and information on the market. This holds especially true when an organization is competing on an Electronic Marketplace with multiple other organizations. The main drawbacks of Electronic Marketplaces are that they are either insufficiently regulated, resulting in fraud or biased trading, or that they are too restrictive, which prohibits participants from efficiently exchanging goods or services. Another issue with Electronic Marketplaces is that the non-standard interaction with the market and its participants represents a handicap to people who are used to interaction patterns from the real world. Furthermore, it is important to encourage customer's loyalty and bind them to the market, especially in the B2C and C2C domain. Hence, doing business should be a fun experience for customers rather than a tedious task.

We focused on the realization of Electronic Marketplaces by means of Electronic Institutions - a Multi-Agent System methodology. To reduce the complexity of adapting and creating e-Business processes that resemble or extend the business processes of an organization we introduced the 3D Electronic Institution Implementation Process that makes use of the 3D Electronic Institution methodology [14]. The process uses Electronic Institutions for the creation of regulated environments. This process suits two purposes. First, it offers a general approach to implement (3D) Electronic Institutions based on a formal role description of agents and a blueprint of the organization to be modeled. This approach helps to reduce the implementation complexity by using an intuitive method to relate processes of the corresponding real world organization to the components of the Electronic Institution. Second, it provides a straight forward visualization of the organization in a 3D Virtual World, by visualizing agents as avatars representing customer centric business processes and the Electronic Institution as 3D building that provides a supporting environment for the agents.

Second, even though it is possible to use a single Electronic Institution to represent an organization in an Electronic Market, this construct reaches its limits when complex processes that span across multiple cooperating organizations are needed. For this purpose we introduced the Virtual Organization construct, which enables the connection of multiple Electronic Institution that are represented as one organization on the market. Even tough multiple organizations (implemented as Electronic Institution) might form a Virtual Organization, it is represented as a single building in the 3D Virtual World, offering a single point of interaction to users. Based on the results presented in Section 2.7 the conclusion can be drawn that Virtual Organizations are suitable to regulate Electronic Marketplaces to a certain extent. By using the proposed 3D Electronic Institution Framework and a supporting implementation process, Virtual Organizations can be defined in a straight forward manner. As Virtual Organizations are based on Electronic Institutions they can make use of the same mechanisms to regulate the interaction of agents. The issues of regulating agents outside Electronic Institutions and missing inter-Electronic Institution communication has been overcome by introducing the `Ether`, which provides a regulated environment when agents move outside an Electronic Institution and enables inter-Electronic Institution communication by means of local and global agents. However, when Virtual Organizations are visualized in a 3D Virtual World certain design approaches and mechanisms such as transition constraint validation, advanced room mappings and how to deal with multiple user agent instances need to be considered. Still, it is up to the creator of the Electronic Marketplace whether to visualize its Virtual Organizations by means of a 3D Virtual World or just offer a traditional form-based Web application that enables users to delegate actions such as purchasing a product to agents participating in a Virtual Organization.

Third, having created an Electronic Marketplace by using the proposed implementation process and general framework ensures that interactions within the market are regulated and an efficient exchange of goods is possible, but it does not guarantee that participants benefit from their interaction with the market. Users need relevant information to decide what to buy, when to buy, from whom to buy, what to sell, when to sell and to whom to sell. In a nutshell, users need relevant and accurate information to base their decisions on. Even tough the Internet is a valuable source for information, it also involves unique challenges such as the sheer amount of ever expanding unstructured data, the difficulty to differentiate between relevant and irrelevant information and that information is scattered over multiple sites. In recent years the Semantic Web, which is an extension to the World Wide Web, has alleviated some of these issues by providing structured data that is processable by machines. However, the Semantic Web cannot compete with the World Wide Web in terms of pure data amount. Hence, it would be beneficial to use the strengths of one technology to compensate the weaknesses of the other.

Addressing these issues, we presented a way to bridge structured and unstructured information via hybrid semantic search by using a combined index structure. Furthermore, we introduced a novel input mechanism for hybrid semantic search that combines the clean and concise input mechanisms of keyword-based search engines with the expressiveness of the input mechanisms provided by Semantic Search Engines. This interactive input mechanism can be used to formulate ontology-aware queries without prior knowledge of the ontology. We presented the Hybrid Semantic Search System (HS^3), which uses the combined index to put hybrid semantic search into practice and implements the interactive ontology-aware keyword-based input mechanism. This system is capable of automatically enriching a given KB with relevant data from the World Wide Web. For demonstration purpose we applied HS^3 to the tourism domain by using an ontology based on the Harmonise tourism ontology and a corresponding KB, which holds facts of 14,000 accommodations and 1,900 tourism destinations. To demonstrate that HS^3 is applicable to multiple domains we applied it to the news domain by using the KIM Ontology and KB.

We presented the results of performance tests that have been conducted to determine the response times of HS^3 when search queries of different complexity are processed. In addition, we evaluated HS^3's precision and recall on two different datasets and compared it to a similar system. It was demonstrated that HS^3 needs at most 0.618 seconds and on average only 0.218 seconds to return the result for a query on a corpus of 167,029 annotated news documents, compared to the competing system which needs about 30 seconds on a corpus of 145,316 annotated news documents, reaching similar precision and recall results. Furthermore, we presented results of a user evaluation that has been conducted to evaluate the usability and user acceptance of the novel interactive ontology-aware keyword-based input mechanism. Based on the results presented in Section 3.7 the conclusion can be drawn that the hybrid semantic search approach and the combined index structure facilitate an amalgamation of keyword-based and concept-based search that provides relevant information within reasonable response times. Furthermore, the user evaluation has proven that the proposed interactive ontology-aware keyword-based input mechanism enables users to easily formulate ontology-aware queries that combine keywords, concepts, instances and properties. In addition, the evaluation showed that users are able to formulate complex ontology-aware queries without prior knowledge of the ontology due to the interactive nature of the proposed input mechanism. Users stated that the concept, instance and property suggestion mechanisms were either helpful or very helpful and that they especially liked the concise input mechanism and clean user interface.

Regarding future work, it would be interesting to research trust in the context of Virtual Organizations. One of the advantages of a Virtual Organization is that it is completely transparent to a user whether she deals with a single organization or a group of organizations that jointly conduct business. The user does not need to care, because she is always interacting with the single point of contact of the Virtual Organization, which is its visual representation as building in the 3D Virtual World. However, this advantage can turn into a disadvantage when trust is a concern, because the user cannot tell who has access to her confidential information. Since the member organizations of a Virtual Organization need to exchange confidential information, part or all of them have access to them. Consider a Virtual Organization that consists of a B2C and a B2B organization. The B2C organization implements the customer services agents, but payment processes are implemented by the B2B organization. Therefore, the customer thinks she is providing her credit card information only to the B2C organization but actually it is forwarded to the B2B organization which executes the payment process.

Another interesting research topic in the context of Virtual Organization is the usage of hybrid access methods to interact with Virtual Organizations. We incorporated a hybrid access method in the Forum Virtual Organization. It is possible to interact with the Forum via the 3D Virtual World and via an ordinary Web Browser. In this context consistency becomes important, because the state change in the Virtual Organization that is triggered in one visual representation needs to be reflected by the other visual representation. Furthermore, this setting offers the possibility to evaluate what interaction method with Virtual Organizations is preferred by users.

Regarding hybrid semantic search and the HS^3 system it would be interesting to extend the suggestion mechanism of the interactive ontology-aware interface with a context-aware suggestion mechanism. The context-aware suggestion mechanism could be used to suggest only those instances that are of relevance in the formulated query. For example, if a users starts to formulate a query with "Vienna hasAttraction" and then starts to type the string "cathe" only those cathedrals that are actually located in Vienna should show up in the suggestion list. For example, St. Stephen's Cathedral should be part of the list, but not the Temple Expiatori de la Sagrada Familia which is located in Barcelona. Furthermore, it would be interesting to check which Machine Learning techniques yield the best result when used to calculate the score value for related concepts in a *Concept Identification Graph (CIG)*. Currently the weights need to be defined manually by a Knowledge Engineer. This task could be automated by manually annotating a set of documents and the usage of Machine Learning techniques to learn the appropriate score values of related concepts. Finally, we would like to apply HS3 to a publicly available ontology, KB and related document corpus with according relevance judgments (ground truth) to do a more accurate prediction of HS3's precision and recall and compare it with other systems.

APPENDIX A

Details on the EIs forming the Itchy Feet EM

A.1 The Auction Electronic Institution

In the following the scenes of the Auction Electronic Institutions will be presented, including the finite state machine and the messages that can be uttered within the scene.

A.1.1 Clearing Scene

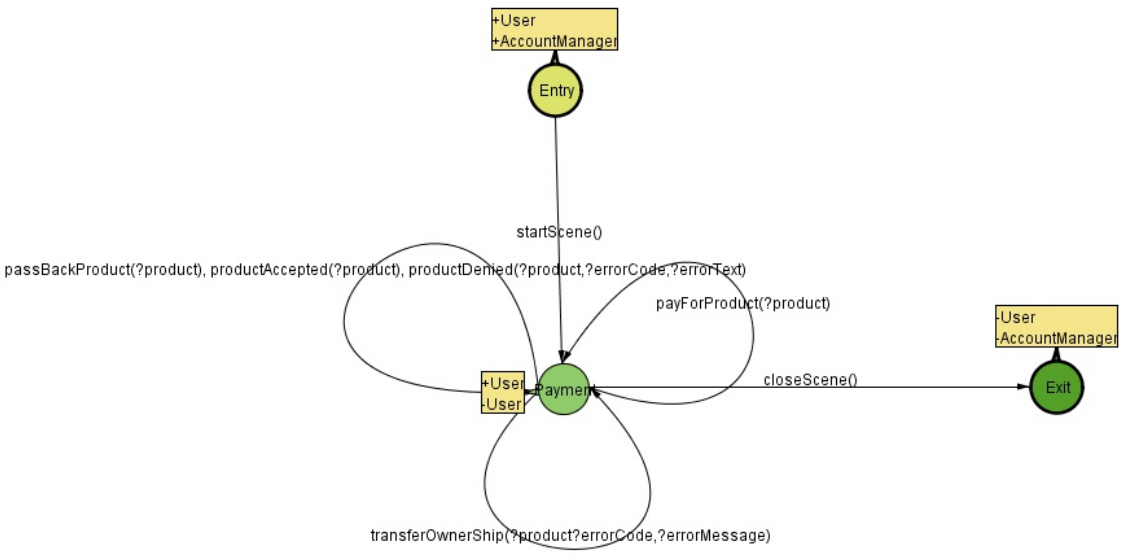

Figure A.1: Clearing Scene

Sender	Illoc.	Ontology Term	Params	Receiver
AccountManager	inform	`startScene`	{}	*all*
User	request	`passBackProduct`	product	*AccountManager*
AccountManager	response	`productAccepted`	product	*User*
AccountManager	response	`productDenied`	product,errorCode,errorText	*User*
AccountManager	response	`transferOwnerShip`	product,errorCode,errorMessage	*User*
AccountManager	inform	`closeScene`	{}	*all*

Table A.1: Ontological terms and messages used in the Clearing Scene

A.1.2 Auction Scene

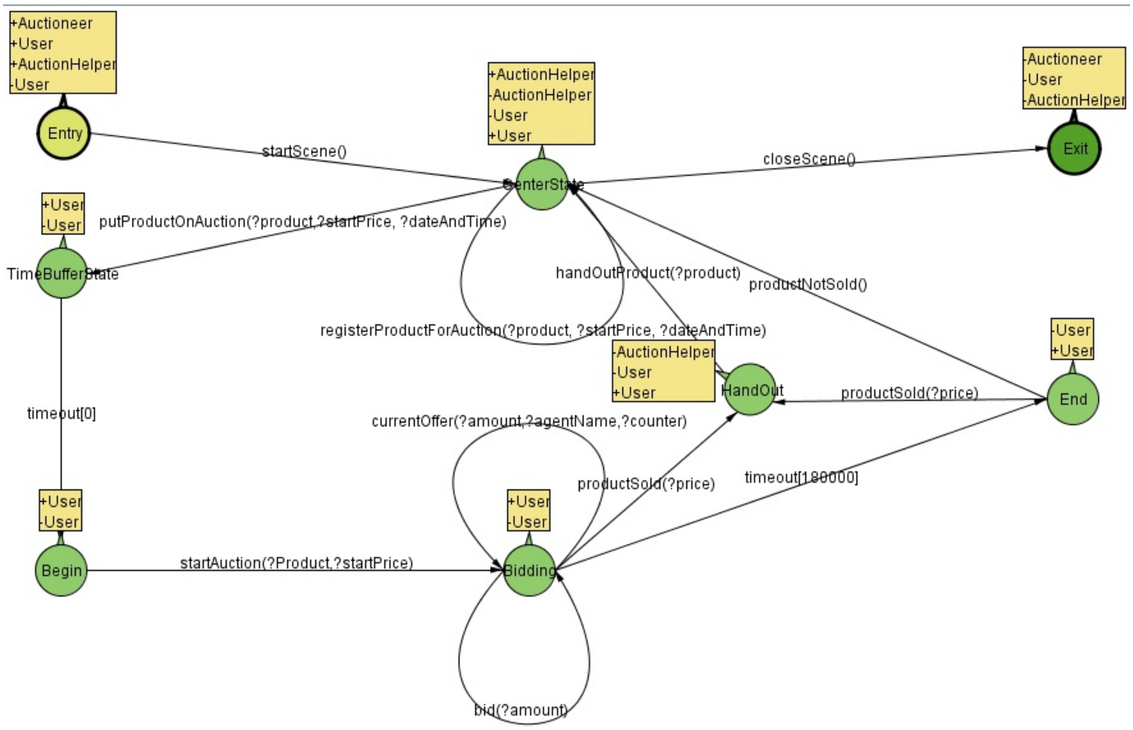

Figure A.2: Auction Scene

Sender	Illoc.	Ontology Term	Params	Receiver
Auctioneer	inform	`startScene`	{}	*all*
AuctionHelper	request	`registerProductForAuction`	product,startPrice,dateAndTime	*Auctioneer*
Auctioneer	inform	`putProductOnAuction`	product,startPrice,dateAndTime	*all*
Auctioneer	inform	`startAuction`	product,startPrice	*all*

User	inform	`bid`	amount	*Auctioneer*
Auctioneer	inform	`currentOffer`	amount,agentName,counter	*all*
Auctioneer	inform	`productSold`	price	*all*
Auctioneer	inform	`productNotSold`	{}	*all*
Auctioneer	inform	`handOutProduct`	product	*User*
Auctioneer	inform	`closeScene`	{}	*all*

Table A.2: Ontological terms and messages used in the Auction Scene

A.1.3 Offering Scene

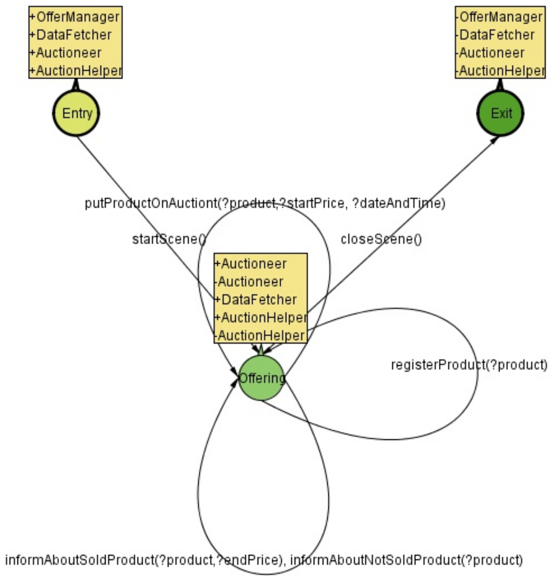

Figure A.3: Offering Scene

Sender	Illoc.	Ontology Term	Params	Receiver
OfferManager	inform	`startScene`	{}	*all*
OfferManager	request	`putProductOnAuction`	product,startPrice,dateAndTime	*AuctionHelper*
AuctionHelper	inform	`informAboutSoldProduct`	soldProduct,endPrice	*OfferManager*
AuctionHelper	inform	`informAboutNotSoldProduct`	notSoldProduct	*OfferManager*
DataFetcher	request	`registerProduct`	product	*OfferManager*
OfferManager	inform	`closeScene`	{}	*all*

Table A.3: Ontological terms and messages used in the Offering Scene

A.1.4 Information Scene

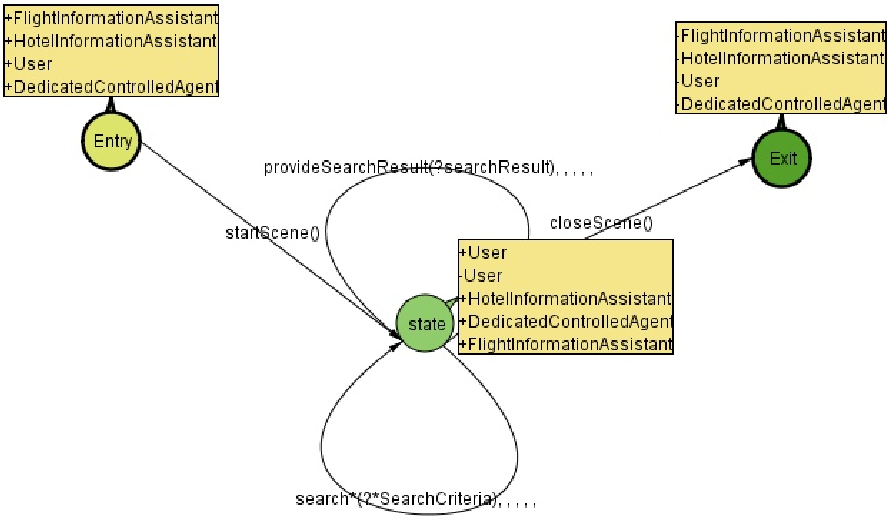

Figure A.4: Information Scene

Sender	Illoc.	Ontology Term	Params	Receiver
FIA (FlightInformationAssistant)	inform	`startScene`	{}	*all*
User	request	`searchForFlights`	searchCriteria	*FIA*
DCA (DedicatedControlledAgent)	request	`searchForFlights`	searchCriteria	*FIA*
FIA	response	`provideSearchResult`	searchResult	*User*
FIA	response	`provideSearchResult`	searchResult	*DCA*
User	request	`searchForHotels`	searchCriteria	*HIA*
DCA	request	`searchForHotels`	searchCriteria	*HIA*
HIA (HotelInformationAssistant)	response	`provideSearchResult`	searchResult	*User*
HIA	response	`provideSearchResult`	searchResult	*DCA*
FIA	inform	`closeScene`	{}	*all*

Table A.4: Ontological terms and messages used in the Information Scene

A.2 The Travel Agency Electronic Institution

In the following the scenes of the Travel Agency Electronic Institutions will be presented, including the finite state machine and the messages that can be uttered within the scene.

A.2.1 Booking Scene

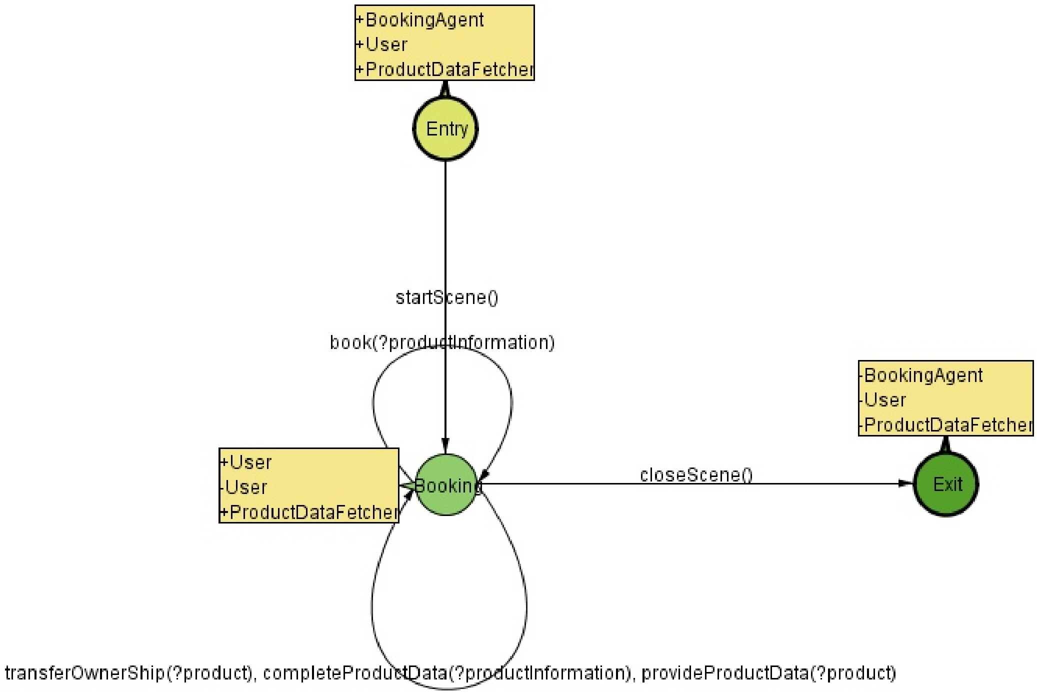

Figure A.5: Booking Scene

Sender	Illoc.	Ontology Term	Params	Receiver
BookingAgent)	inform	`startScene`	{ }	*all*
User	request	`book`	product	*BookingAgent*
BookingAgent	response	`transferOwnerShip`	product,errorCode,errorMessage	*User*
BookingAgent	request	`completeProductData`	productInformation	*ProductDataFetcher*
ProductDataFetcher	response	`provideProductData`	product	*User*
BookingAgent	inform	`closeScene`	{ }	*all*

Table A.5: Ontological terms and messages used in the Booking Scene

A.2.2 Clearing Scene

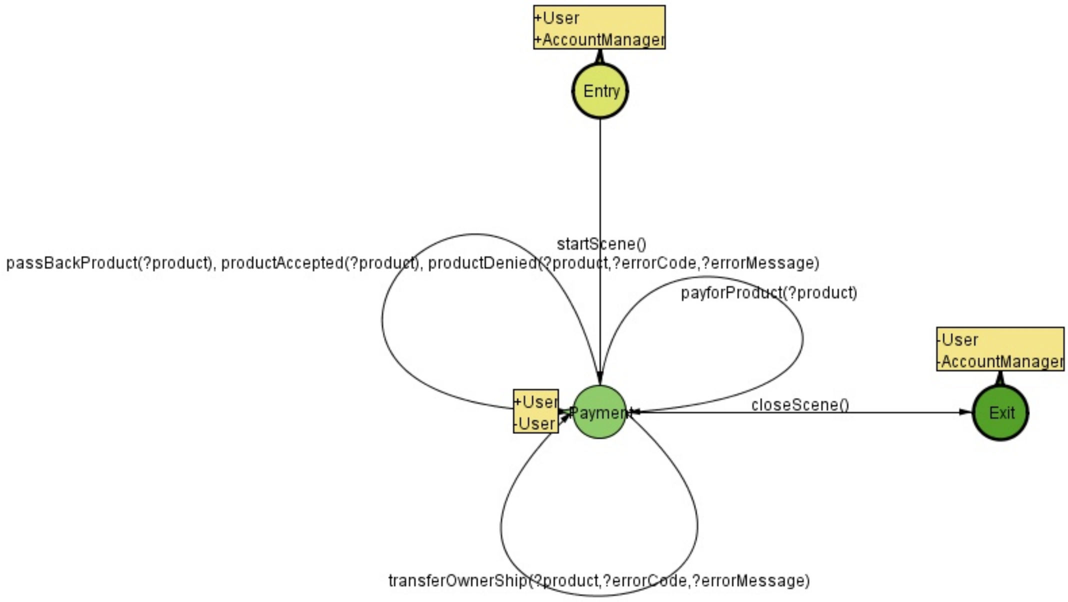

Figure A.6: Clearing Scene

Sender	Illoc.	Ontology Term	Params	Receiver
AccountManager	inform	`startScene`	{ }	*all*
User	request	`passBackProduct`	product	*AccountManager*
AccountManager	response	`productAccepted`	product	*User*
AccountManager	response	`productDenied`	product,errorCode,errorText	*User*
AccountManager	response	`transferOwnerShip`	product,errorCode,errorMessage	*User*
AccountManager	inform	`closeScene`	{ }	*all*

Table A.6: Ontological terms and messages used in the Clearing Scene

A.2.3 Information Scene

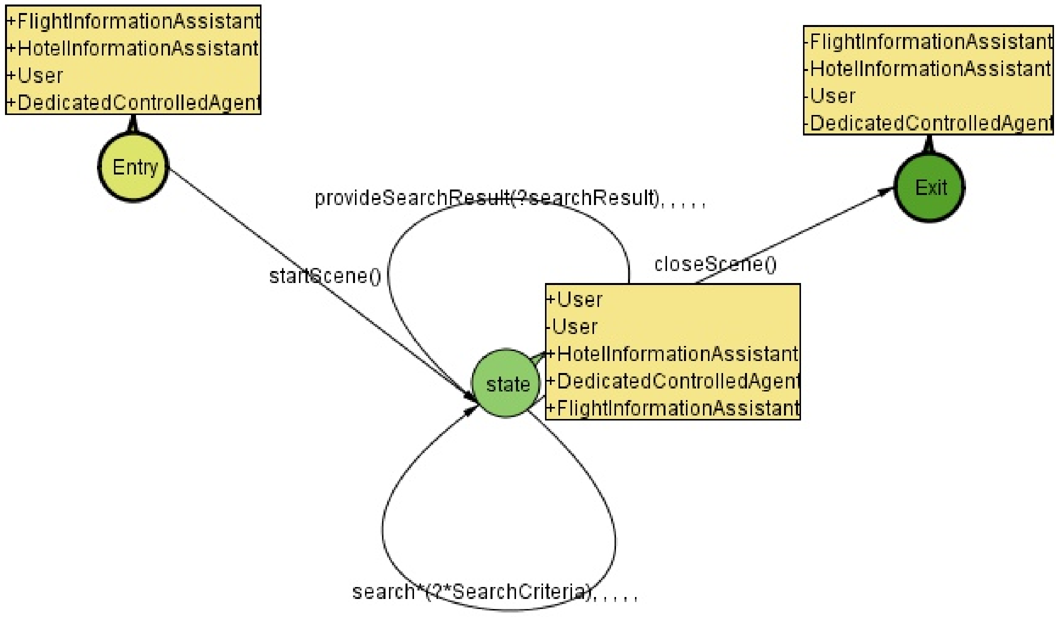

Figure A.7: Information Scene

Sender	Illoc.	Ontology Term	Params	Receiver
FIA (FlightInformationAssistant)	inform	`startScene`	{}	*all*
User	request	`searchForFlights`	searchCriteria	*FIA*
DCA (DedicatedControlledAgent)	request	`searchForFlights`	searchCriteria	*FIA*
FIA	response	`provideSearchResult`	searchResult	*User*
FIA	response	`provideSearchResult`	searchResult	*DCA*
User	request	`searchForHotels`	searchCriteria	*HIA*
DCA	request	`searchForHotels`	searchCriteria	*HIA*
HIA (HotelInformationAssistant)	response	`provideSearchResult`	searchResult	*User*
HIA	response	`provideSearchResult`	searchResult	*DCA*
FIA	inform	`closeScene`	{}	*all*

Table A.7: Ontological terms and messages used in the Information Scene

A.2.4 Sale Scene

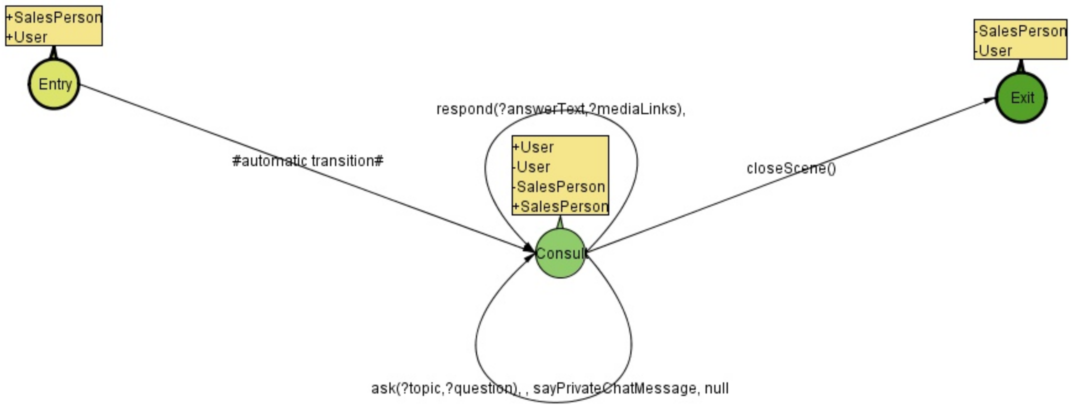

Figure A.8: Sale Scene

Sender	Illoc.	Ontology Term	Params	Receiver
User	request	`ask`	topic,question	*all*
User	response	`answer`	answer	*all*
SalesPerson	request	`ask`	topic,question	*all*
SalesPerson	response	`answer`	answer	*all*
User	request	`sayPrivateChatMessage`	message	*User*
SalesPerson	request	`sayPrivateChatMessage`	message	*User*
SalesPerson	inform	`closeScene`	{}	*all*

Table A.8: Ontological terms and messages used in the Sale Scene

A.3 The Ether Electronic Institution

In the following the scenes of the Ether Electronic Institutions will be presented, including the finite state machine and the messages that can be uttered within the scene.

A.3.1 Ether Scene

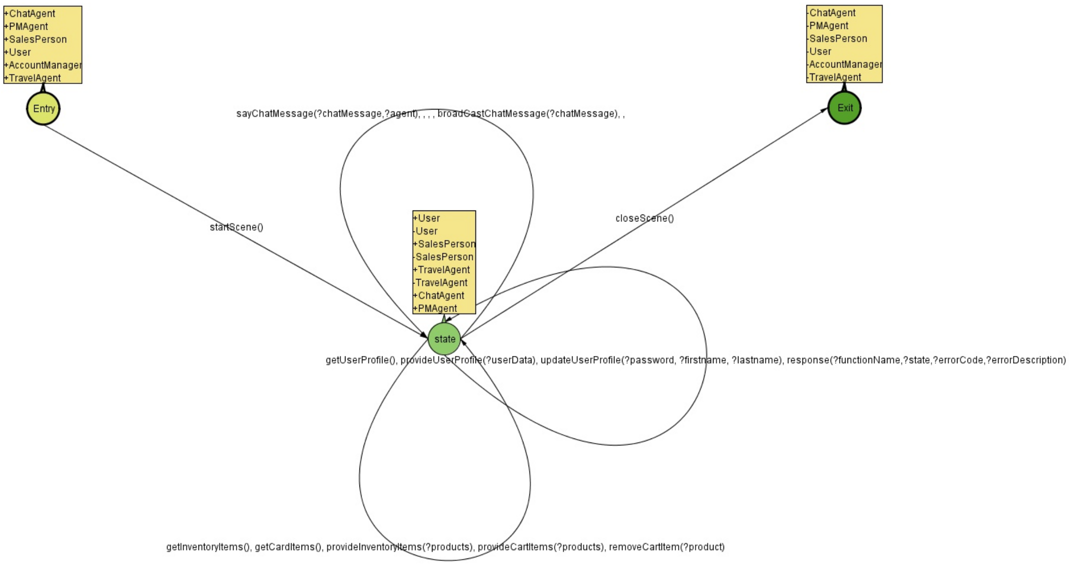

Figure A.9: Ether Scene

Sender	Illoc.	Ontology Term	Params	Receiver
AccountManager	inform	`startScene`	{}	*all*
User	request	`getInventoryItems`	{}	*AccountManager*
User	request	`getCartItems`	{}	*AccountManager*
AccountManager	response	`provideInventoryItems`	products	*User*
AccountManager	response	`provideCartItems`	products	*User*
User	request	`removeCartItem`	product	*AccountManager*
User	inform	`sayChatMessage`	chatMessage,agent	*ChatAgent*
ChatAgent	inform	`sayChatMessage`	chatMessage,agent	*User*
all	inform	`broadCastChatMessage`	chatMessage,sender	*ChatAgent*
ChatAgent	inform	`broadCastChatMessage`	chatMessage,sender	*all*
User	request	`getUserProfile`	{}	*AccountManager*
AccountManager	request	`provideUserProfile`	UserData	*User*
User	request	`updateUserProfile`	password,firstname,lastname	*AccountManager*
AccountManager	inform	`closeScene`	{}	*all*

Table A.9: Ontological terms and messages used in the Ether Scene

A.3.2 Registration Scene

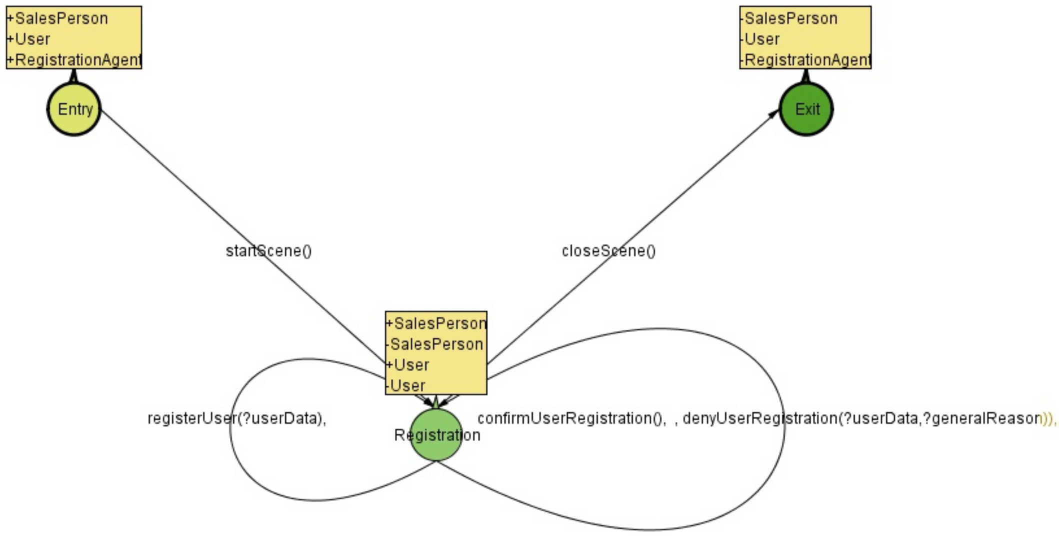

Figure A.10: Registration Scene

Sender	Illoc.	Ontology Term	Params	Receiver
RegistrationAgent	inform	`startScene`	{}	*all*
User	request	`registerUser`	userData	*RegistrationAgent*
SalesPerson	request	`registerUser`	userData	*RegistrationAgent*
RegistrationAgent	response	`confirmUserRegistration`	{}	*User*
RegistrationAgent	response	`confirmUserRegistration`	{}	*SalesPerson*
RegistrationAgent	response	`denyUserRegistration`	userData,reason	*User*
RegistrationAgent	response	`denyUserRegistration`	userData,reason	*SalesPerson*
RegistrationAgent	inform	`closeScene`	{}	*all*

Table A.10: Ontological terms and messages used in the Registration Scene

A.3.3 Validation Scene

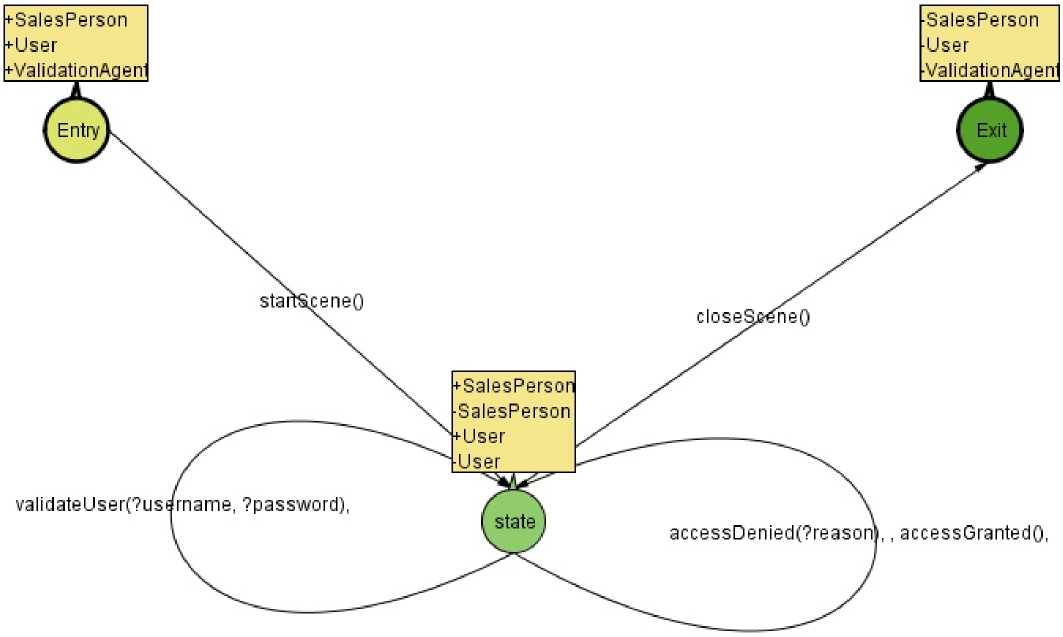

Figure A.11: Validation Scene

Sender	Illoc.	Ontology Term	Params	Receiver
ValidationAgent	inform	`startScene`	{}	*all*
User	request	`validateUser`	username,password	*ValidationAgent*
SalesPerson	request	`validateUser`	username,password	*ValidationPerson*
ValidationAgent	response	`accessDenied`	reason	*User*
ValidationAgent	response	`accessDenied`	reason	*SalesPerson*
ValidationAgent	response	`accessGranted`	{{	*User*
ValidationAgent	response	`accessGranted`	{{	*SalesPerson*
ValidationAgent	inform	`closeScene`	{}	*all*

Table A.11: Ontological terms and messages used in the Validation Scene

A.4 The Forum Electronic Institution

In the following the scenes of the Forum Electronic Institutions will be presented, including the finite state machine and the messages that can be uttered within the scene.

A.4.1 Forum Scene

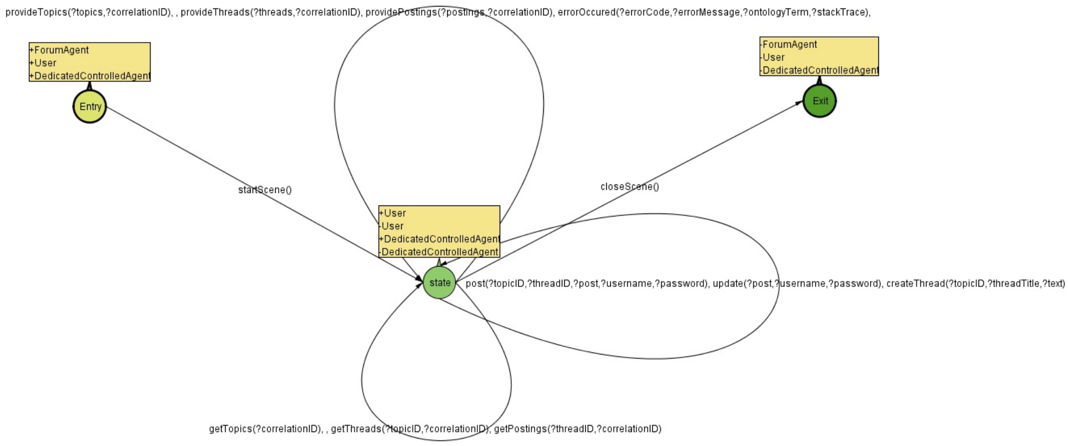

Figure A.12: Forum Scene

Sender	Illoc.	Ontology Term	Params	Receiver
ForumAgent	inform	`startScene`	{ }	*all*
User	request	`getTopics`	correlationID	*ForumAgent*
DCA	request	`getTopics`	correlationID	*ForumAgent*
User	request	`getThreads`	topicID,correlationID	*ForumAgent*
User	request	`getPostings`	threadID,correlationID	*ForumAgent*
ForumAgent	response	`provideTopics`	topics,correlationID	*User*
ForumAgent	response	`provideThreads`	threads,correlationID	*User*
ForumAgent	response	`provideTopics`	topics,correlationID	*DCA*
ForumAgent	response	`providePostings`	postings,correlationID	*User*
ForumAgent	error	`errorOccured`	errorCode,errorMessage, ontologyTerm,stackTrace	*User*
ForumAgent	error	`errorOccured`	errorCode,errorMessage, ontologyTerm,stackTrace	*DCA*
User	request	`post`	topicID,threadID,post,username,password	*ForumAgent*
User	request	`update`	post,username,password	*ForumAgent*
User	request	`createThread`	topicID,threadTitle,post,username,password	*ForumAgent*
ForumAgent	inform	`closeScene`	{ }	*all*

Table A.12: Ontological terms and messages used in the Forum Scene

APPENDIX B

SerQL Queries for the Performance Evaluation

In the following the SerQL queries that have been used to mimic part of HS3 functionality, to compare HS3's performance with the performance of a Triple Store, are listed.

```
SELECT A FROM
  {A} rdf:type {eTourism:Hotel},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Apartment},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Farm},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:GuestHouse},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:BedAndBreakfast},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Hotel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Apartment},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
```

```
SELECT A FROM
  {A} rdf:type {eTourism:Farm},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:GuestHouse},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:BedAndBreakfast},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100
USING NAMESPACE
  rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
  eTourism = <http://tuwien.ac.at/rdf/itchyfeet#>,
  eTourismCity = <http://www.itchy-feet.org/initialload/city/>
```

Listing B.1: SerQL Query *Concept Standard*

```
SELECT A FROM
  {A} rdf:type {eTourism:Hotel},
  {A} eTourism:accommodationLocation {eTourismCity:16190},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Apartment},
  {A} eTourism:accommodationLocation {eTourismCity:16190},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Farm},
  {A} eTourism:accommodationLocation {eTourismCity:16190},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel},
  {A} eTourism:accommodationLocation {eTourismCity:16190},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:GuestHouse},
  {A} eTourism:accommodationLocation {eTourismCity:16190},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:BedAndBreakfast},
  {A} eTourism:accommodationLocation {eTourismCity:16190},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
```

```
  {A} rdf:type {eTourism:Hotel},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Apartment},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Farm},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:GuestHouse},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:BedAndBreakfast},
  {A} eTourism:accommodationLocation {eTourismCity:16190} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Hotel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Apartment},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Farm},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:GuestHouse},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:BedAndBreakfast},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Hotel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
```

```
SELECT A FROM
  {A} rdf:type {eTourism:Apartment},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Farm},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:GuestHouse},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:BedAndBreakfast},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100
USING NAMESPACE
  rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
  eTourism = <http://tuwien.ac.at/rdf/itchyfeet#>,
  eTourismCity = <http://www.itchy-feet.org/initialload/city/>
```

Listing B.2: SerQL Query *Concept Complex*

```
SELECT A FROM
  {A} rdf:type {eTourism:Hotel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {B} eTourism:languageSpoken {eTourism:Class29},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Apartment},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {B} eTourism:languageSpoken {eTourism:Class29},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Farm},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {B} eTourism:languageSpoken {eTourism:Class29},
  {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel},
  {A} eTourism:accommodationLocation {B},
```

```
    {B} rdf:type {eTourism:BroadLocation},
    {B} eTourism:languageSpoken {eTourism:Class29},
    {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
    {A} rdf:type {eTourism:GuestHouse},
    {A} eTourism:accommodationLocation {B},
    {B} rdf:type {eTourism:BroadLocation},
    {B} eTourism:languageSpoken {eTourism:Class29},
    {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
    {A} rdf:type {eTourism:BedAndBreakfast},
    {A} eTourism:accommodationLocation {B},
    {B} rdf:type {eTourism:BroadLocation},
    {B} eTourism:languageSpoken {eTourism:Class29},
    {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
    {A} rdf:type {eTourism:Hotel},
    {A} eTourism:accommodationLocation {B},
    {B} rdf:type {eTourism:BroadLocation},
    {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
    {A} rdf:type {eTourism:Apartment},
    {A} eTourism:accommodationLocation {B},
    {B} rdf:type {eTourism:BroadLocation},
    {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
    {A} rdf:type {eTourism:Farm},
    {A} eTourism:accommodationLocation {B},
    {B} rdf:type {eTourism:BroadLocation},
    {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
    {A} rdf:type {eTourism:YouthHostel},
    {A} eTourism:accommodationLocation {B},
    {B} rdf:type {eTourism:BroadLocation},
    {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
    {A} rdf:type {eTourism:GuestHouse},
    {A} eTourism:accommodationLocation {B},
    {B} rdf:type {eTourism:BroadLocation},
    {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
    {A} rdf:type {eTourism:BedAndBreakfast},
    {A} eTourism:accommodationLocation {B},
    {B} rdf:type {eTourism:BroadLocation},
    {A} eTourism:facility {eTourism:SteamBath_1} LIMIT 100 UNION
SELECT A FROM
    {A} rdf:type {eTourism:Hotel},
    {A} eTourism:accommodationLocation {B},
    {B} rdf:type {eTourism:BroadLocation},
    {B} eTourism:languageSpoken {eTourism:Class29}
    LIMIT 100 UNION
```

```
SELECT A FROM
  {A} rdf:type {eTourism:Apartment},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {B} eTourism:languageSpoken {eTourism:Class29}
  LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Farm},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {B} eTourism:languageSpoken {eTourism:Class29}
  LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {B} eTourism:languageSpoken {eTourism:Class29}
  LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:GuestHouse},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {B} eTourism:languageSpoken {eTourism:Class29}
  LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:BedAndBreakfast},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation},
  {B} eTourism:languageSpoken {eTourism:Class29}
  LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Hotel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Apartment},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:Farm},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:YouthHostel},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
  {A} rdf:type {eTourism:GuestHouse},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100 UNION
SELECT A FROM
```

```
  {A} rdf:type {eTourism:BedAndBreakfast},
  {A} eTourism:accommodationLocation {B},
  {B} rdf:type {eTourism:BroadLocation} LIMIT 100
USING NAMESPACE
  rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
  eTourism = <http://tuwien.ac.at/rdf/itchyfeet#>,
  TourismCity = <http://www.itchy-feet.org/initialload/city/>
```

Listing B.3: SerQL Query *Concept Complex 2*

Bibliography

[1] H. Aldewereld, F. Dignum, A. García-Camino, P. Noriega, J. A. Rodríguez-Aguilar, and C. Sierra. Operationalisation of norms for usage in electronic institutions. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems*, AAMAS '06, pages 223–225, New York, NY, USA, 2006. ACM.

[2] R. Alt, M. A. Cäsar, and J. U. Grau. Collaboration in the consumer product goods industry - analysis of marketplaces. In *Proceedings of the 10th European Conference on Information Systems, Information Systems and the Future of the Digital Economy (ECIS '02)*, pages 582–595, Gdansk, Poland, 2002.

[3] G. Aranda, V. Botti, and C. Carrascosa. Mmog based on mas: the mmog layer. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1149–1150, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[4] E. Argente, V. Julian, and V. Botti. Multi-agent system development based on organizations. *Electronic Notes in Theoretical Computer Science*, 150(3):55–71, 2006.

[5] F. Baader, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[6] C. Badica, G.-G. Popa, M. G. Mihnea Scafes, M. Gawinecki, P. Kobzdej, and M. Paprzycki. Design considerations for a negotiation component in a model e-commerce agent system. In *Proceedings of the Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 205–208, Washington, DC, USA, 2006. IEEE Computer Society.

[7] H. Bast, A. Chitea, F. Suchanek, and I. Weber. Ester: Efficient search on text, entities, and relations. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 671–678, New York, NY, USA, 2007. ACM.

[8] H. Berger, M. Dittenbach, D. Merkl, A. Bogdanovych, S. Simoff, and C. Sierra. Playing the e-business game in 3d virtual worlds. In *Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*, OZCHI '06, pages 333–336, New York, NY, USA, 2006. ACM.

[9] H. Berger, M. Dittenbach, D. Merkl, A. Bogdanovych, S. Simoff, and C. Sierra. Opening new dimensions for e-tourism. *Virtual Reality*, 11:75–87, June 2007.

[10] R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi, and D. Petrelli. Hybrid search: Effectively combining keywords and semantic searches. In *Proceedings of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC'08, pages 554–568, Berlin, Heidelberg, 2008. Springer-Verlag.

[11] M. Bichler. *The Future of e-Markets: Multidimensional Market Mechanisms*. Cambridge University Press, 2001.

[12] N. Bikakis, G. Giannopoulos, T. Dalamagas, and T. Sellis. Integrating keywords and semantics on document annotation and search. In R. Meersman, T. Dillon, and P. Herrero, editors, *On the Move to Meaningful Internet Systems, OTM 2010*, volume 6427 of *Lecture Notes in Computer Science*, pages 921–938. Springer-Verlag, 2010.

[13] A. Bogdanovych, H. Berger, S. Simo, and C. Sierra. Narrowing the gap between humans and agents in e-commerce: 3d electronic institutions. In *In Proceedings of the 6th International Conference on Electronic Commerce and Web Technologies (EC-Web '05), LNCS 3590*, pages 128–137, Copenhagen, Denmark, 2005. Springer-Verlag.

[14] A. Bogdanovych, M. Esteva, S. Simoff, C. Sierra, and H. Berger. A methodology for 3d electronic institutions. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, pages 57:1–57:3, New York, NY, USA, 2007. ACM.

[15] A. Bogdanovych, S. Simoff, and M. Esteva. Virtual institutions prototype. In *In proceedings of the Eights International Joint Conference on Autonomous Agents*

and Multiagent Systems (AAMAS 2009), pages 1373–1374, Budapest, Hungary, 2009. ACM.

[16] J. Broekstra and A. Kampman. Serql: A second generation rdf query language. In *Proceedings of the 2003 SWAD-Europe Workshop on Semantic Web Storage and Retrieval*, 2003.

[17] J. Broekstra, A. Kampman, and F. v. Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, ISWC '02, pages 54–68, London, UK, 2002. Springer-Verlag.

[18] S. Bromuri, V. Urovi, P. Contreras, and K. Stathis. A virtual e-Retailing environment in GOLEM. In *Proceedings of the 4th International Conference on Intelligent Environments (IET '08)*, pages 1–8, Seattle, WA, USA, July 2008.

[19] H. L. Cardoso and E. Oliveira. Virtual enterprise normative framework within Electronic Institutions. In *Engineering Societies in the Agents World V*, Lecture Notes in Computer Science, pages 14–32, Berlin/Heidelberg, Germany, 2005. Springer-Verlag.

[20] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, WWW Alt. '04, pages 74–83. ACM, 2004.

[21] P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19:261–272, February 2007.

[22] D. Chao. Doom as an interface for process management. In *Proceedings of the SIGCHI Conference: Human Factors in Computing Systems (CHI '01)*, pages 152–157, New York, NY, USA, 2001. ACM.

[23] D. K. W. Chiu and H. fung Leung. Towards ubiquitous tourist service coordination and integration: A multi-agent and semantic web approach. In *Proceedings of the 7th international conference on Electronic commerce*, ICEC '05, pages 574–581, New York, NY, USA, 2005. ACM.

[24] G. Cuní, M. Esteva, P. García, E. Puertas, C. Sierra, and T. Solchaga. Masfit: Multi-agent system for fish trading. In L. S. Ramon López de Mántaras, editor,

In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), pages 710–714. IOS Press, IOS Press, 2004.

[25] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 168–175. Association for Computational Linguistics, 2002.

[26] J. Debenham and S. Simoff. An e-market framework for informed trading. In *Proceedings of the 15th international conference on World Wide Web (WWW '06)*, pages 651–658, New York, NY, USA, 2006. ACM Press.

[27] J. K. Debenham. Bargaining with information. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '04, pages 663–670, Washington, DC, USA, 2004. IEEE Computer Society.

[28] J. K. Debenham. Automated trading: Making it happen. In *Proceedings of the 24th IASTED international conference on Artificial intelligence and applications*, pages 1–6, Anaheim, CA, USA, 2006. ACTA Press.

[29] R. Delbru, N. Toupikov, M. Catasta, and G. Tummarello. A node indexing scheme for web entity retrieval. In L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, editors, *Proceedings of the Extended Semantic Web Conference (ESWC 2010)*, volume 6089 of *Lecture Notes in Computer Science*, pages 240–256. Springer-Verlag, 2010.

[30] F. Dignum. Abstract norms and electronic institutions. In *Proceedings of the International Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA '02)*, pages 93–104, 2002.

[31] V. Dignum and F. Dignum. Modelling agent societies: co-ordination frameworks and institutions. In *Proceedings of the 10th Portuguese Conference on Artificial Intelligence : Progress in Artificial Intelligence Knowledge Extraction, Multi-Agent Systems, Logic Programming, and Constraint Solving*, EPIA '01, pages 191–204, London, UK, December 2001. Springer-Verlag.

[32] V. Dignum, H. Weigand, and L. Xu. Agent societies: Towards frameworks-based design. In M. Wooldridge, G. Weiß, and P. Ciancarini, editors, *Agent-Oriented Software Engineering II*, volume 2222 of *Lecture Notes in Computer Science*, pages 33–49. Springer-Verlag, 2002.

[33] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 652–659, New York, NY, USA, 2004. ACM.

[34] S. Drago, A. Bogdanovych, M. Ancona, S. J. Simoff, and C. Sierra. From graphs to euclidean virtual worlds: Visualization of 3d electronic institutions. In *Proceedings of the thirtieth Australasian conference on Computer science - Volume 62*, ACSC '07, pages 25–33, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.

[35] M. Esteva. *Electronic Institutions: From Specification to Development*. PhD thesis, Institut d'Investigació en Intelligència Artificial (IIIA), Barcelona, Spain, 2003.

[36] M. Esteva, D. de la Cruz, and C. Sierra. Islander: an electronic institutions editor. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*, AAMAS '02, pages 1045–1052, New York, NY, USA, 2002. ACM.

[37] M. Esteva, J. Rodriguez-Aguilar, C. Sierra, P. Garcia, and J. Arcos. On the formal specifications of electronic institutions. In *Agent Mediated Electronic Commerce, The European AgentLink Perspective*, pages 126–147, London, UK, 2001. Springer-Verlag.

[38] M. Esteva, J. A. Rodriguez-Aguilar, J. L. Arcos, C. Sierra, P. Noriega, B. Rosell, and D. de la Cruz. Electronic institutions development environment. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: demo papers*, AAMAS '08, pages 1657–1658. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[39] M. Esteva, B. Rosell, J. A. Rodriguez-Aguilar, and J. L. Arcos. Ameli: An agent-based middleware for electronic institutions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '04, pages 236–243, Washington, DC, USA, 2004. IEEE Computer Society.

[40] M. Fasli and M. Michalakopoulos. e-Game: a platform for developing auction-based market simulations. *Decision Support Systems*, 44:469–481, 2008.

[41] S. S. Fatima, M. Wooldridge, and N. R. Jennings. A comparative study of game theoretic and evolutionary models of bargaining for software agents. *Artificial Intelligence Review*, 23:187–205, April 2005.

[42] M. Fernandez, V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta, and P. Castells. Semantic search meets the web. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 253–260, Washington, DC, USA, 2008. IEEE Computer Society.

[43] D. Ferrucci and A. Lally. Uima: An architectural approach to unstructured information. processing in the corporate research environmen. *Journal of Natural Language Engineering*, 10:327–348, September 2004.

[44] J. Fisher and A. Craig. Developing business community portals for SMEs - issues of design, development and sustainability. *Electronic Markets*, 15(2):136–145, 2005.

[45] O. Fodor and H. Werthner. Harmonise: A step toward an interoperable e-tourism marketplace. *International Journal of Electronic Commerce*, 9:11–39, January 2005.

[46] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2)*, volume 500-215 of *NIST Special Publication*, pages 243–252. NIST, 1994.

[47] J. Froschauer, I. Seidel, M. Gärtner, H. Berger, and W. Merkl. New directions in science communication: A virtual research & experience landscape. In *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education (E-Learn 2009)*, pages 2590–2597. AACE, 2009.

[48] J. Froschauer, I. Seidel, M. Gärtner, H. Berger, and W. Merkl. Design and evaluation of a serious game for immersive cultural training. In *International Conference on Virtual Systems and Multimedia*, pages 253–260. IEEE CS Press, 2010.

[49] M. Gärtner, A. Rauber, and H. Berger. Bridging structured and unstructured data via hybrid semantic search and interactive ontology-enhanced query formulation. *in review*.

[50] M. Gärtner, I. Seidel, and H. Berger. Agent mediated trading in a 3d e-tourism environment. In *Proceedings of the 10th International Conference on Electronic Commerce (ICEC '08)*, pages 1–10, New York, NY, USA, 2008. ACM.

[51] M. Gärtner, I. Seidel, J. Froschauer, and H. Berger. The formation of virtual organizations by means of electronic institutions in a 3d e-tourism environment. *Information Sciences*, 180:3157–3169, September 2010.

[52] M. Gärtner, I. Seidel, J. Froschauer, H. Berger, and A. Rauber. A hybrid semantic search system for the tourism domain. *in review*.

[53] F. Giunchiglia, U. Kharkevich, and I. Zaihrayeu. Concept search. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC 2009 Heraklion, pages 429–444, Berlin, Heidelberg, 2009. Springer-Verlag.

[54] M. Grieger. Electronic marketplaces: A literature review and a call for supply chain management research. *European Journal of Operational Research*, 144:280–294, 2003.

[55] R. Guha, R. McCool, and E. Miller. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 700–709, New York, NY, USA, 2003. ACM.

[56] M. A. Hearst. *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[57] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7–38, January 1998.

[58] T. Juan, A. Pearce, and L. Sterling. Roadmap: Extending the gaia methodology for complex open systems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS '02)*, pages 3–10, New York, NY, USA, 2002. ACM.

[59] E. Kandogan, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu. Avatar semantic search: A database approach to information retrieval. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 790–792, New York, NY, USA, 2006. ACM.

[60] N. Karacapilidis and P. Moraïtis. Intelligent agents for an artificial market system. In *Proceedings of the 5th International Conference on Autonomous Agents*, AGENTS '01, pages 592–599, New York, NY, USA, 2001. ACM.

[61] G. Kasneci, F. M. Suchanek, G. Ifrim, S. Elbassuoni, M. Ramanath, and G. Weikum. Naga: Harvesting, searching and ranking knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1285–1288, New York, NY, USA, 2008. ACM.

[62] E. Kaufmann and A. Bernstein. How useful are natural language interfaces to the semantic web for casual end-users? In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 281–294, Berlin, Heidelberg, 2007. Springer-Verlag.

[63] E. Kaufmann and A. Bernstein. Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. *Journal of Web Semantics*, 8:377–393, November 2010.

[64] A. Kiryakov, D. Ognyanov, and D. Manov. Owlim - a pragmatic semantic repository for owl. In M. Dean, Y. Guo, W. Jun, R. Kaschek, S. Krishnaswamy, Z. Pan, and Q. Z. Sheng, editors, *WISE Workshops*, volume 3807 of *Lecture Notes in Computer Science*, pages 182–192. Springer-Verlag, 2005.

[65] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, and K. M. Goranov. Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2:49–79, December 2004.

[66] B. Kot, B. Wuensche, J. Grundy, and J. Hosking. Information visualisation utilising 3d. computer game engines case study: A source code comprehension tool. In *Proceedings of the 6th ACM SIGCHI New Zealand Chapter's International Conference on Computer-Human Interaction : Making CHI Natural (CHINZ '05)*, CHINZ '05, pages 53–60, New York, NY, USA, 2005. ACM.

[67] J. Manojlovich, P. Prasithsangaree, S. Hughes, J. Chen, and M. Lewis. UTSAF: A multiagent-based framework for supporting military-based distributed interactive simulations in 3D virtual environments. In *Proceedings of the 35th Conference on Winter Simulation (WSC '03)*, pages 960–968, New Orleans, LA, USA, 2003. Winter Simulation Conference.

[68] C. Martin, D. Schreckenghost, P. Bonasso, D. Kortenkamp, T. Milam, and C. Thronesbery. An environment for distributed collaboration among humans and software agents. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03)*, pages 1062–1063, New York, NY, USA, 2003. ACM.

[69] N. Mavetera and A. Kadyamatimba. A comprehensive agent: Mediated e-Market framework. In *Proceedings of the 5th International Conference on Electronic Commerce (ICEC '03)*, pages 158–164, New York, NY, USA, 2003. ACM.

[70] J. Mayfield and T. Finin. Information retrieval on the semantic web: Integrating inference and retrieval. In *Proceedings of the SIGIR Workshop on the Semantic Web*, 2003.

[71] G. Navarro. A guided tour to approximate string matching. *ACM Computer Surveys*, 33:31–88, March 2001.

[72] J. Nielsen. *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, Thousand Oaks, CA, USA, 1999.

[73] E. Oliveira and A. P. Rocha. Agents advanced features for negotiation in electronic commerce and virtual organisations formation process. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce*, volume 1991 of *Lecture Notes in Computer Science*, pages 78–97, London, UK, 2001. Springer-Verlag.

[74] P. O'Reilly and P. Finnegan. Performance in electronic marketplaces: Theory in practice. *Electronic Markets*, 15(1):23–37, 2005.

[75] C. Pao-Hua, C. Kuang-Ku, and W. Menq-Jiun. Use of an intelligent agent for an e-Commerce bargaining system. *Lecture Notes in Computer Science, Knowledge-Based Intelligent Information and Engineering Systems*, 4251:300–309, 2006.

[76] T. R. Payne, K. Sycara, and M. Lewis. Varying the user interaction within Multi-Agent Systems. In *Proceedings of the 4th International Conference on Autonomous Agents (AGENTS '00)*, pages 412–418, New York, NY, USA, 2000. ACM.

[77] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov. Kim - a semantic platform for information extraction and retrieval. *Journal of Natural Language Engineering*, 10:375–392, September 2004.

[78] K. Prantner, Y. Ding, M. Luger, Z. Yan, and C. Herzog. Tourism ontology and semantic management system: State-of-the-arts analysis. In *Proceedings of the IADIS International Conference WWW/Internet 2007*, 2007.

[79] E. Prud'hommeaux and A. Seaborne. Sparql query language for rdf. Technical report, W3C, October 2004.

[80] L. Reeve and H. Han. Survey of semantic annotation platforms. In *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC 2005)*, SAC '05, pages 1634–1638, New York, NY, USA, 2005. ACM.

[81] J. A. Rodríguez‐Aguilar, F. J. Martín, P. Noriega, P. Garcia, and C. Sierra. Towards a test-bed for trading agents in electronic auction markets. *AI Communications*, 11:5–19, January 1998.

[82] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[83] G. Schreiber, A. Amin, L. Aroyo, M. van Assem, V. de Boer, L. Hardman, M. Hildebrand, B. Omelayenko, J. van Osenbruggen, A. Tordai, J. Wielemaker, and B. Wielinga. Semantic annotation and search of cultural-heritage collections: The multimedian e-culture demonstrator. *Journal of Web Semantics*, 6:243–249, November 2008.

[84] I. Seidel. *Engineering 3D Virtual World Applications : Design, Realization and Evaluation of a 3D e-Tourism Environment*. PhD thesis, Technical University of Vienna, Department of Software Technology and Interactive Systems, Vienna, Austria, 2010.

[85] I. Seidel and H. Berger. Integrating Electronic Institutions with 3D Virtual Worlds. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07)*, pages 481–484, Washington, DC, USA, 2007. IEEE Computer Society.

[86] I. Seidel, M. Gärtner, J. Froschauer, H. Berger, and D. Merkl. An agent-based centralized e-Marketplace in a virtual environment. In *Proceedings of the 21st International Conference on Software Engineering & Knowledge Engineering (SEKE'09)*, pages 218–221, Skokie, IL, USA, 2009. Knowledge Systems Institute Graduate School.

[87] I. Seidel, M. Gärtner, M. Pöttler, H. Berger, M. Dittenbach, and D. Merkl. *Tourism Informatics: Visual Travel Recommender Systems, Social Communities, and User Interface Design*, chapter Itchy Feet: A 3D e-Tourism Environment, pages 209–242. Information Science Reference, Hershey, PA, USA, 2009b.

[88] G. Smith, M. L. Maher, and J. S. Gero. Designing 3D Virtual Worlds as a society of agents. In *Proceedings of the 10th International Conference on Com-*

puter Aided Architectural Design Futures (CAADFutures '03), pages 105–114, Dordrecht, Netherlands, 2003. Kluwer Academic Publishers.

[89] G. Stumme, A. Hotho, and B. Berendt. Semantic web mining - state of the art and future directions. *Journal of Web Semantics*, 4(2):124–143, 2006.

[90] F. Suchanek, G. Kasneci, and G. Weikum. YAGO - a large ontology from Wikipedia and WordNet. *Journal of Web Semantics*, 6(3):203–217, September 2008.

[91] V. Tablan, D. Damljanovic, and K. Bontcheva. A natural language query interface to structured information. In *Proceedings of the 5th European Semantic Web Conference on the Semantic Web: Research and Applications*, ESWC'08, pages 361–375. Springer-Verlag, 2008.

[92] D. Traum and J. Rickel. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '02)*, pages 766–773, New York, NY, USA, 2002. ACM.

[93] B. Travica. Virtual organization and electronic commerce. *SIGMIS Database*, 36(3):45–68, 2005.

[94] H. Wang, T. Tran, and C. Liu. Ce2 - towards a large scale hybrid search engine with integrated rankingsupport. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1323–1324, New York, NY, USA, 2008. ACM.

[95] S. Wang, S. Zheng, L. Xu, D. Li, and H. Meng. A literature review of electronic marketplace research: Themes, theories and an integrative framework. *Information Systems Frontiers*, 10(5):555–571, 2008.

[96] C. Wartena, R. Brussee, L. Gazendam, and W.-O. Huijsen. Apolda: A practical tool for semantic annotation. In *DEXA Workshops*, pages 288–292. IEEE Computer Society, 2007.

[97] D. C. Wimalasuriya and D. Dou. Using multiple ontologies in information extraction. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 235–244. ACM, 2009.

[98] F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing multiagent systems: The gaia methodology. *ACM Transactions on Software Engineering Methodologies*, 12(3):317–370, 2003.

[99] Z. Zeng and L. Zhang. An integrated approach for developing e-Commerce system. In *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WICOM '07)*, pages 3596–3599, September 2007.

[100] G. Zenz, X. Zhou, E. Minack, W. Siberski, and W. Nejdl. From keywords to semantic queries - incremental query construction on the semantic web. *Journal of Web Semantics*, 7:166–176, September 2009.

[101] Z.-J. Zha, L. Yang, T. Mei, M. Wang, and Z. Wang. Visual query suggestion. In *Proceedings of the 17th ACM International Conference on Multimedia (MM '09)*, MM '09, pages 15–24, New York, NY, USA, 2009. ACM.

Curriculum Vitae

Markus Gärtner

Address:	Kienmayergasse 4/8 1140 Vienna Austria
E-Mail:	gmax@chello.at
Birthdate:	22. March 1980
Birthplace:	Vienna
Citizenship:	Austria

Education

Jun. 2007 – present	PhD Studies in Technical Sciences at the Vienna University of Technology
March 2007	Master in Business Informatics (MSc.) (*passed with distinction*)
2006 – present	Master Studies in Media Informatics at the Vienna University of Technology
2005 – 2007	Master Studies in Business Informatics at the Vienna University of Technology
Jan. 2005	Bachelor in Business Informatics (BSc.) (*passed with distinction*)
2002 – 2005	Bachelor Studies in Business Informatics at the Vienna University of Technology
1998 – 2000	Advanced training course for Business Data Processing Specialist
1995 – 1998	Business and Informatics School
1994 – 1995	Higher Technical Education Institute
1990 – 1994	Primary School Mater Salvatoris
1986 – 1990	Elementary School Josefinum Breitensee

Professional Experience

Oct. 2011 – present	Project Manager & Solution Designer at T-Mobile Austria (40 h / week)
Jun. 2007 – 2010	Project member in the “Itchy Feet” Research Project (Austrian Science Fund (FWF)) (10 h / week)

Jun. 2007 – Aug. 2011	Project Manager at T-Mobile Austria (30 h / week)
2006 – Jun. 2007	Senior Software Architect at T-Mobile Austria (40 h / week)
2005 – 2006	Software Developer & Architect at iTEC Informationssystems AG and T-Mobile Austria (25 h / week)
2002 – 2005	Software Developer & Architect at BEKO Engineering & Informatics AG and T-Mobile Austria (25 h / week)
2001 – 2002	Software Developer & Architect at BEKO Engineering & Informatics AG and T-Mobile Austria (40 h / week)
2000 – 2001	Software Developer at BEKO Engineering & Informatics AG