D I S S E R T A T I O N

Telecommunications and the Next Generation Web: Introducing ENUM to the Semantic Web

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

unter der Leitung von

o. Univ.-Prof. Dipl.-Ing. Dr. Georg Gottlob
Institut für Informationssysteme [E184]

und

Univ.-Ass. Mag. Dr. Robert Baumgartner
Institut für Informationssysteme [E184]

eingereicht an der Technischen Universität Wien
Fakultät für technische Naturwissenschaften und Informatik

von

Dipl.-Ing. Kurt Reichinger
Ettenreichgasse 50 / 4 / 53
1100 Wien

Wien, am 12. Dezember 2006

Kurzfassung

Die vorliegende Dissertation beschäftigt sich mit der grundlegenden Fragestellung, inwieweit eine Verknüpfung von innovativen Ansätzen, in der Telekommunikation einerseits, und dem World Wide Web andererseits, in der Lage ist, signifikanten Mehrwert für einen oder beide dieser Bereiche zu generieren. Telephone Number Mapping (ENUM) und das Semantic Web Projekt Friend-Of-A-Friend (FOAF) werden herangezogen, um exemplarisch zu untersuchen, inwieweit eine Annäherung dieser beiden Gebiete Synergien erzeugen kann und welcher Aufwand damit potentiell verbunden ist.

ENUM ist eine Entwicklung der letzten Jahre, die das Ziel einer verstärkten Konvergenz von klassischer Telekommunikation und dem Internet verfolgt. Konkret handelt es sich um die Abbildung von herkömmlichen Telefonnummern auf Internet-Domain-Namen, die in Kombination mit Mechanismen des Domain Name Systems eine Verknüpfung mit weiteren Kommunikations- und sonstigen Parametern erlauben. Darauf aufsetzend wird die Einführung einer Vielzahl innovativer Dienste möglich, wobei das Hauptanwendungsgebiet aktuell auf Voice-over-IP (VoIP) liegt.
Das zeitgleich entstandene Semantic Web ist eine zukunftsweisende Weiterentwicklung des heutigen World Wide Web mit der Zielsetzung eines zunehmend maschinenlesbaren Webs. Hierzu werden die im Web verfügbaren Inhalte mit Informationen zur Bedeutung (Semantik) dieser Inhalte versehen, die es Maschinen ihrerseits ermöglichen, den Inhalt nunmehr in seinem Kontext zu erfassen. Die auf den unterschiedlichen Ebenen des Semantic Web stattfindende Innovation ermöglicht die Kreation einer Vielzahl neuer Web-basierter Dienste und Applikationen. FOAF ist eine der ersten Anwendungen des Semantic Web und dient der maschinenlesbaren Modellierung sozialer Netzwerke.

Der erste Teil der Dissertation widmet sich der theoretischen Aufarbeitung der grundlegenden Entwicklungen und Funktionalitäten von ENUM und des Semantic Web vor dem Hintergrund einer beabsichtigten Annäherung der beiden Forschungsgebiete. Basierend auf einleitenden Analysen werden in der Folge Schnittstellen und Anknüpfungspunkte zwischen den beiden Themenbereichen aus einem formalen, theoretisch geprägten Blickwinkel erörtert und Überlappungsbereiche aufgezeigt.

Der zweite Teil der Dissertation beschäftigt sich im Sinne eines Ingenieurs-Ansatzes mit möglichen Anwendungsgebieten und der konkreten Umsetzung der Überlegungen aus dem ersten Teil. In diesem Abschnitt wird der PHOAF Prototyp vorgestellt, der im Zuge der Arbeit an der Dissertation zur Evaluierung unterschiedlicher Applikationsszenarien entwickelt und implementiert wurde. PHOAF integriert ENUM- und Semantic Web-Technologien und bildet die technologische Basis für eine Reihe von Anwendungsbeispielen. Dies sind allesamt konvergente Applikationen, die im Kontext von VoIP, sozialen Netzwerken sowie Vertrauensüberlegungen (Trust) angesiedelt sind und im Detail vorgestellt werden.

Acknowledgements

First and foremost, I would like to thank my supervisors at the Institute of Information Systems in the Computer Science Department at Vienna University of Technology: Georg Gottlob, now with the University of Oxford, for his open-minded approach offering the possibility of doing this thesis at DBAI Group, in the first place; and Robert Baumgartner, my personal supervisor, who has been influential and encouraging throughout the work on this thesis. His outstanding commitment is best reflected in his continued support when his contract with DBAI expired in August 2006.

A big thank-you goes to my brother and Java guru Gerd Reichinger, who was a particularly invaluable help with regard to Java coding and PHOAF implementation issues.

Furthermore, I want to thank the various members of the international scientific community I had the privilege to meet for all the interesting discussions and valuable comments on my work. This includes the anonymous peer reviewers who did not make me happy all the time, but nevertheless contributed to the quality of the work.

This section would be incomplete without mentioning the Austrian ENUM community. Alexander Mayrhofer, wizard of the Austrian ENUM domain, was always there for a chat, an advice on IETF issues and for one or the other alteration to my NAPTR records. Richard Stastny and Michael Haberler, masterminds of the Austrian ENUM success story, with their tireless efforts for the development of ENUM have been an ever-inspiring example for my own work

More thanks go to Georg Serentschy and Ernst Langmantel at my employer, the Austrian Regulatory Authority for Broadcasting and Telecommunications, both providing continuous support for my research activities.

Finally it should be mentioned that this work has been partially supported by REWERSE, a research "Network of Excellence" (NoE) on "Reasoning on the Web" within the 6th European Framework Program of the European Commission.

Contents

1 Introduction and Motivation

In recent years, the world of telecommunication has undergone major changes, with the migration from analogue to digital technology in the eighties and the massive success of mobile telephony in the nineties of the last century. The occurrence of the Internet (and the World Wide Web as one of its most successful services) meant another huge challenge for traditional telecommunication providers as core networks had to be upgraded for the transport of rapidly increasing amounts of data traffic; and access networks had to be prepared for the customers growing demand for broadband Internet. Innovative start-up companies brought new services to the market with astonishing speed, venture capitalists where quick at hand to finance the new ideas; everything seemed to go perfect - until the bubble burst. Today, more than six years after the collapse, the industry has gained momentum again, and new challenges lie ahead with convergence between established telecommunication and new Internet technology being one of the key issues.

This thesis puts the spotlight on two representatives of new developments in the area of telecommunication and the Internet: Telephone Number Mapping (ENUM), designed for enabling interoperability between the worlds of traditional telephony and the Internet; and the Semantic Web, adding computer-understandable meaning to documents on the World Wide Web for creation of the Next Generation Web. Both are based on innovative, forward-thinking concepts offering a broad range of opportunities for the development of new services and applications. The focus of this thesis is on identifying synergies the integration of ENUM and Semantic Web technology is able to bring. On the one hand, this is done by a theoretic evaluation identifying possible links and connections; on the other hand, practical application examples are introduced to show the possible benefits with real-world use cases.

While there is extensive research in various Semantic Web areas presented at conferences and published in proceedings and books, there is only a very limited amount of research papers being published regarding ENUM. This does not mean there is no development in ENUM; however, related work is rather done in standardisation organisations or entities directly working with ENUM technology, e.g. domain registries or VoIP providers. A similar situation applies to the main issue of this thesis, the integration of ENUM and the Semantic Web, where the lack of available research is due to the novelty of the work with the first related papers submitted by the author (together with co-authors). This thesis in particular aims to put the issue on the research map in the first place.

The research methodology chosen for this thesis is a blend of theoretic and application-oriented work complemented by discussions with experts from both fields of interest, i.e. from the ENUM and Semantic Web community. The research process started with an extensive study of literature available from both areas. As the author has been working on

ENUM issues for the last years[1], the main focus of this first research cycle was on the Semantic Web. In a next step, similarities and links between ENUM and the Semantic Web were identified based on the theoretic analysis performed initially. This process led to first ideas on possible synergies and first visions regarding interesting applications. In parallel, the first theoretic considerations were presented to the scientific community at the WTAS'2005 conference in Calgary for an early test of the work's significance [59]. Although people showed immediate interest and feedback on the ideas highlighted was encouraging, the discussions with experts from both areas showed that there is very limited knowledge about the other topic, i.e. ENUM experts often are not informed about Semantic Web issues, and vice versa. In order to evaluate the theoretic results and considerations, a prototype allowing for the implementation of an application test bed was developed and implemented. The prototype again was presented to the scientific community at the ISWC'2005 conference in Galway for immediate feedback to be considered in the ongoing research [62]. Once more, the work raised interest and stimulated fruitful discussions with experts from the Semantic Web community.
At the same time, the author submitted a proposal for a new Internet RFC to the IETF aiming to have the developments reflected in an IETF RFC [63]. That Internet draft dealing with the registration of a new ENUMservice related to the thesis' work was tabled and presented at the IETF'65 meeting in Dallas. Although the input was not adopted as an immediate IETF ENUM Working Group item, it further stimulated the creation of another Internet Draft defining the general registration process for new ENUMservices [40]. As the IETF is approached with a growing number of proposals regarding ENUMservices, and the actual registration process being deemed to be rather time-consuming, a guide and template document was created for more efficient handling.
The thesis work was continued with more ideas regarding new applications evaluated and the results once more presented to the scientific community at the IRMA'2006 conference in Washington, D.C. [60]. Finally, the prototype was further upgraded now supporting applications in the context of the emerging Web 2.0. Recapitulating the whole research process, introducing ENUM to the Semantic Web is sort of entering virgin soil with some questions remaining unanswered. The thesis, however, shows that connecting the two areas is possible, reasonable and potentially brings a lot of opportunities. Further research and commitment to the issue according to the results of this thesis definitely should be in the interest of both communities.

Figure 1 illustrates the research process described above in a chronological way from the first literature studies to the final steps of finishing this thesis. The illustration differentiates the work in both areas and shows the major scientific feedback loops with presentations at various conferences.

[1] The author has been involved in the successful introduction of ENUM in Austria in his role at the Austrian Regulatory Authority for Broadcasting and Telecommunications. This ranged from conducting public consultations, participating in working groups and workshops, organising the delegation of the Austrian ENUM domain to setting up the administrative framework for the commercial start of ENUM in Austria.

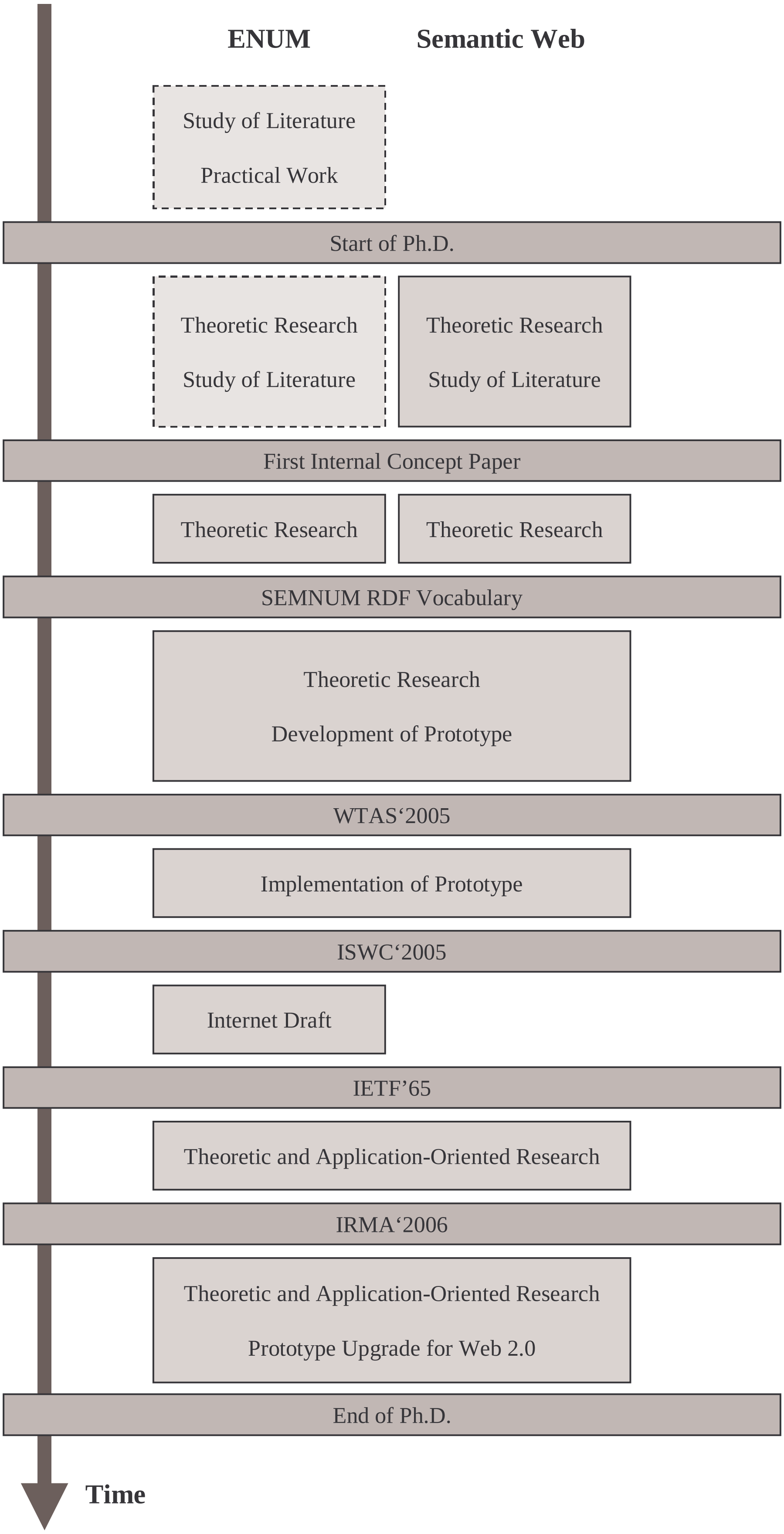

Figure 1: Research Process

Section 1 of this thesis introduces the reader to the overall context, explains the research methodology chosen and the research path followed, and gives a short overview on each section of the document.

Section 2 gives a basic overview on ENUM, starting with the generic principle of the ENUM mapping function and illustrative ENUM use cases. This is complemented by technical details on the embedding of ENUM in the Internet Domain Name System, the Uniform Resource Identifiers used with ENUM, and the various ENUMservices responsible for the broad range of possible applications supported. A comparison of User ENUM and Infrastructure ENUM closes the section.

Section 3 provides an introduction to the Semantic Web, explaining the horizontal layers of the Semantic Web layer cake. This starts with Uniform Resource Identifiers and Unicode (layer 1) and moves on to Extensible Markup Language documents, schema and query language (layer 2), and the Resource Description Framework with associated query languages (layer 3). This is followed by a description of schemas and ontologies (layer 4), and the logic layer providing the possibility to make logical statements (layer 5). Finally, the top layers introducing proof (layer 6) and trust (layer 7) are discussed.

Section 4 deals with a new phenomenon called Web 2.0. This term refers to a new generation of Internet-based services, ranging from social networking sites to wikis, communication tools, and folksonomies. Generally, Web 2.0 services have in common that people are enabled to collaborate and share information online. One of the key aspects is easy tagging of content, which makes the introduction of Web 2.0 services an important step towards the Semantic Web.

Section 5 explains the foundations of the Friend-of-A-Friend (FOAF) project, which is a Semantic Web application particularly well suited for a combined usage with ENUM. FOAF defines a vocabulary based on RDF for expressing metadata about people and their interests, activities and relationships. That way, FOAF facilitates the creation of the Semantic Web version of a typical personal homepage. This section introduces the FOAF basics, from available RDF classes and properties to practical implementation issues. FOAF plays a major part with regard to the PHOAF prototype and related application examples presented in later sections of this thesis.

Section 6 discusses issues connecting the Semantic Web and ENUM from a formal and theoretic point of view. Taking the Semantic Web's layered architecture as guidance, it is analysed from the Semantic Web's layer 1 up to layer 7 whether similarities or relations are to be found in ENUM. Furthermore, possible convergent applications taking use of ENUM and Semantic Web technology in general will be pointed out, building a basis for practical application examples.

Section 7 introduces the PHOAF prototype implemented during the course of the work on this thesis in order to effectively evaluate the basic concept of combining ENUM and the Semantic Web's FOAF project, and to build a base for the implementation of application examples. The PHOAF main functionalities are looking up the ENUM DNS database, retrieving the ENUM data, detecting the location of FOAF RDF data, and finally parsing a FOAF file for requested data. Concretely, this section explains the basic functionalities, the architectural principle being based on Java, implementation characteristics, and workflow issues. The section is complemented with screenshots from the PHOAF prototype.

Section 8 presents applications representative for the combined usage of ENUM and Semantic Web technology. The examples all take advantage of the PHOAF prototype introduced in section 7. Exemplarily, the following applications are presented: VoIP called party information presentation; VoIP call forwarding on called party affiliation; a phonebook contact network; an ENUM/FOAF to RDF transcoding application; a trust calculator for corresponding data in ENUM and FOAF; and finally a unique key to distributed Web 2.0 application content is introduced.

Section 9 gives a detailed introduction to the SEMNUM RDF vocabulary created by the author for describing information about (communication) identifiers stored in ENUM. The purpose of SEMNUM is to have all terms contained in ENUM described in a single vocabulary. The translation of information found in ENUM into RDF terms, enables users or (more precisely) agents to further process the data found and to create new documents based on RDF. In this section, SEMNUM is described by declaring a namespace, presenting related classes and properties, and an example code fragment for illustration.

Section 10 draws final conclusions regarding the benefits of introducing ENUM to the Semantic Web as proposed in this thesis. As work in this area is right at the beginning possible directions for future research are identified.

As indicated above, some results of this thesis have previously been published: this applies to the basic considerations proposing a combination of ENUM and Semantic Web technology [59]; the SEMNUM RDF vocabulary for describing ENUM data [61]; the introduction of the PHOAF prototype [62]; the presentation of application examples being based on PHOAF [60]; and the proposal for registering a new ENUMservice [63].

2 ENUM

2.1 Introduction

ENUM is the acronym for Telephone Number Mapping[2] and describes a protocol for mapping (translating) an ordinary telephone number to an Internet Domain Name. In combination with specific Domain Name System (DNS) records that functionality is used for the detection of services a user – the holder of the telephone number – previously has subscribed to. ENUM therefore can be described as a database comparable to directory assistance. ENUM is based on work of the IETF originally specified in RFC 2916, meanwhile superseded by RFC 3761 [23].

Figure 2: ENUM – one number for many purposes[3]

ENUM allows one to use a telephone number as a unique key pointing to service identifiers associated with a certain person (or organisation). In Internet terminology, these service identifiers are Uniform Resource Identifiers (URIs) as defined by the IETF [10]. Typical examples of URIs used with ENUM are mail URI for e-mail service, Web URL for WWW service, SIP AoR for Voice over IP (VoIP) service, URI for Instant Messaging (IM) service, or telephone URI for pointing to a further telephone number. However, other identifiers as Geographic Positioning System (GPS) location coordinates or a public key for encryption are possible as well.

[2] Other explanations of the abbreviation "ENUM" include E.164 Number to Uniform Resource Identifier Mapping, Electronic Number Mapping, and Electronic Numbering.

[3] Figure: © SWITCH - The Swiss Education & Research Network.

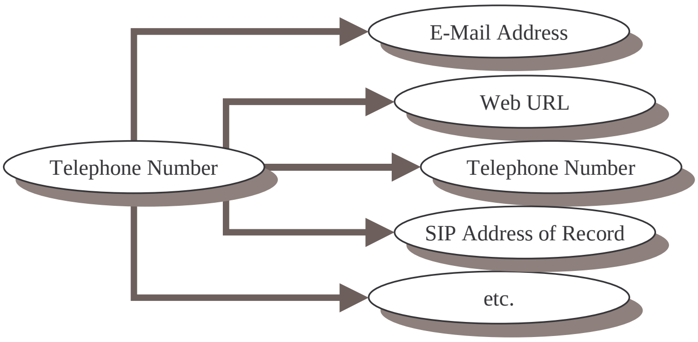

Figure 3: ENUM generic principle – pointing from a telephone number to other identifiers

In a typical usage scenario the unique identifier will stay the same over time, while the telephone number owner's details may change from time to time and subsequently be altered by the subscriber to the ENUM service. The identifier used with ENUM is the well known E.164 telephone number used in the global telephone numbering plan [34].[4] Taking into account today's high penetration of mobile phones[5] which can be seen as personal devices with associated "personal" numbers (as well as the possibility of number portability) the vision of the E.164 telephone number becoming the universal key to a full range of services is tempting.

In order to make a telephone number accessible by (Web) clients or (Web) services, it has to be mapped onto the Internet. This task is performed by ENUM, translating an E.164 telephone number into an ENUM Domain Name (see section 2.3).

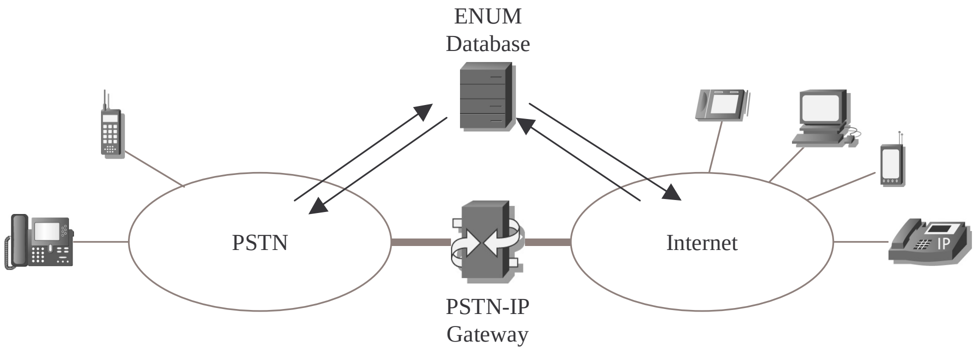

Figure 4: Achieving convergence between PSTN and the Internet using ENUM (simplified overview)

Although the example with multiple parameters to be found behind an ENUM domain may sound striking with an "electronic business card" scenario coming to one's mind, the commercial focal point of the ENUM mapping currently is about IP telephony. For the convergence of voice telephony services, i.e. for reaching a subscriber on the traditional Public Switched Telephone Network (PSTN) from the Internet and vice versa, it is necessary to implement some sort of translation from E.164 telephone numbers (used on the PSTN) to

[4] ITU-T Recommendation E.164 specifies that the entire telephone number should be 15 digits or shorter, and begin with a country prefix. In most countries, this is followed by an area or city code and the subscriber number. Take as an example the Austrian number +43-1-58801, where 43 is the country code (Austria), 1 is the city code (Vienna) and 58801 the subscriber number (Vienna University of Technology).

[5] Statistically, Austria, as an example, already has a SIM card penetration larger than 100% of population.

IP addresses (used on the Internet). ENUM is perfectly equipped to do exactly this. Figure 4 explains the converged scenario with devices from the telephony world (e.g. a fixed-network telephone or a mobile phone) as well as the Internet world (e.g. an IP phone, a personal digital assistant, a personal computer or a tablet PC) doing queries in the ENUM database to extract the communication identifier of the "other world" device.

2.2 E.164 Numbering Plan

Telephone numbers are used for different purposes. They may be interpreted as a user identification (i.e. for identification of the user to the service provider); as a (network) address for mapping to an access point or user device; and as a name for mapping to the current (network) address, where the user has logged in with an appropriate user identification. Identifiers usually are provider-specific, addresses are network-specific, and names are mostly service-specific.[6] In the case of different services or networks wanted to be interoperable, either the same naming conventions must be used, or a mapping from names and/or addresses must be provided.
There are many flavours of telephone numbers reflected in different numbering plans. It has to be differentiated between private, local, national and international numbering plans (see Figure 5). However, there is only one international numbering plan for public telephony numbers: the so-called E.164 Recommendation as defined by ITU-T [34].
Furthermore, a meaning is (or may be) attached to telephone numbers or telephone number ranges as explained and illustrated below.

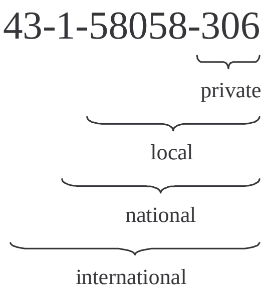

Figure 5: Private, local, national, and international numbering plan

[34] defines the structure and functionality for a couple of categories of numbers used for international public telecommunication: Geographic Areas, Global Services and Networks. For each of the categories the components of the numbering structure and the digit analysis needed to route calls are explained. In addition, the standard comprises regulations regarding Country Code of Trials and Groups of Countries ("GoCs"). [34] defines a maximum length of 15 digits for E.164 numbers and the structure of the country codes, with geographic country codes consisting of 1 to 3 digits, other country codes always of 3 digits.

Regarding the intrinsic meaning of a telephone number, a couple of possible categories are listed as follows. It should be mentioned however, that this meaning may be specific in one or the other way, e.g. depending on country, service, operator or the specific situation.

- Country Code: Giving information on the country administering the number.

[6] Note that this terminology is not used very strictly, as an e-mail *address* definitely is a *name* according to the convention introduced above.

- Number Range: Giving information on the principal usage of a number, e.g. geographical number, mobile number, toll free number, premium rate number, etc.
- Service: Giving information on the service being accessible using a number, e.g. mobile service, emergency services, or time information service.
- Operator: Giving information on the network operator allocating the numbers to its customers.
- Company / Individual: Giving information on a company or an individual using (owning) the number. This is utilised with the Calling Line Identification (CLI) enabling the presentation of the calling party's number on the called party's terminal.
- Status: Giving information of status of called party, e.g. extension 100 for the boss, and extension 101 for the secretary.
- Cost of Call: Giving information on the cost involved when using a number, e.g. toll free numbers, premium rate numbers, or on-net calls.
- Vanity Numbers: Often used in combination with premium rate or toll free numbers, e.g. 0900-DONATE or 0800-HELPLINE.[7]

It is up to future work to evaluate whether the creation of a vocabulary or ontology (see section 3.5) describing the possible meanings of a telephone number can be advantageous for the development of new applications and services.

2.3 ENUM Use Cases

ENUM is used to detect service identifiers associated with a person (or organisation) being the holder (owner) of a telephone number. In this section simple examples for session set-up situations are given in order to explain the usage principles of ENUM.

2.3.1 PSTN-to-Internet Session

A first use case presented is utilising ENUM for setting up sessions initiated from end points on the PSTN. As conventional PSTN devices typically support an addressing scheme based on numbers, those devices and associated systems in most cases do not support the Internet addressing scheme using names; often it is even impossible (or rather complicated, at least) to enter names containing special characters as "@" or "//:", e.g. on traditional telephone sets. In order to set up sessions from traditional PSTN endpoints to Internet endpoints the necessary mapping from numbers to names can be performed using ENUM.
Considering a user A on the PSTN wanting to set up a VoIP call to a user B on the Internet, the call is initiated by user A entering user B's telephone number on a PSTN device. As the traditional telephone network and the Internet are distinct networks with their own protocols and addressing schemes, an appropriate protocol and addressing scheme conversion has to be performed. Typically, this is done at a Gateway located on the edge of PSTN and Internet. Without ENUM, the call would be routed on the PSTN from the source to the destination endpoint. With ENUM implemented, a device (located somewhere between user A and user B) can perform an ENUM query translating user B's phone number into a VoIP identifier to be used for subsequent call routing on the Internet.

[7] Vanity numbers are telephone numbers where a customer can dial letters on the dialpad of an enduser device, i.e. "ABC" is on the "2" button, DEF on the "3" button, and so on. The vanity number 0900-DONATE therefore corresponds to the number 0900-366283, while 0800-HELPLINE corresponds to 0800-43575463.

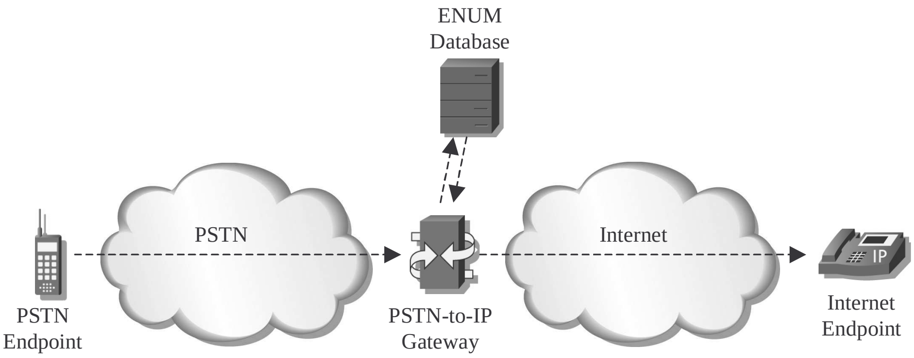

Figure 6: PSTN-to-Internet session initiated using ENUM query

Figure 6 shows an example PSTN-to-Internet voice session utilising an ENUM query initiated by the PSTN-to-IP Gateway to translate the telephone number dialled at the PSTN endpoint (and transmitted with the signalling message to the Gateway) into an Internet name addressing the Internet endpoint.

2.3.2 ENUM-enabled PBX

A Public Branch Exchange (PBX) enables an organisation to implement its own private numbering plan within a self-managed and self-controlled environment. Typically, this results in the organisation having a public central office number (or main number) with self-defined extensions for departments and individuals.[8]

In the standard use case of a PBX every call is checked for whether the called number is to be found within the own organisation numbering plan, or not. If the called number belongs to the organisation's own numbering plan, the call is routed using the own network (incurring no additional costs). If the called number is located in a different numbering plan, the call gets routed to the PSTN (incurring additional costs for the organisation).

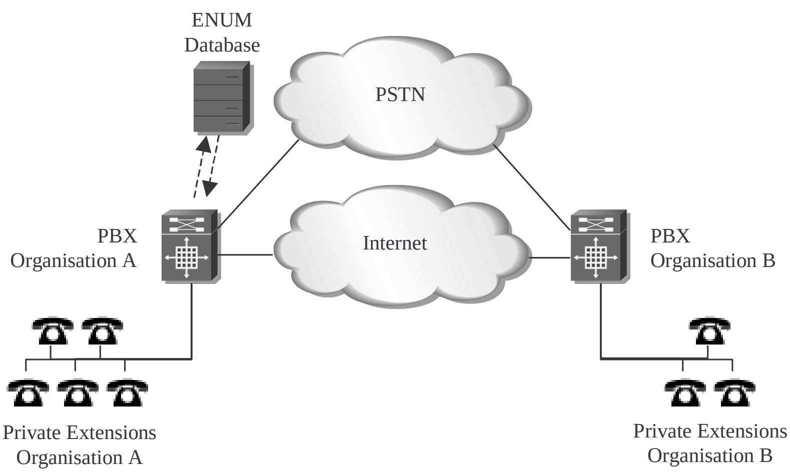

Figure 7: ENUM-enabled PBX

With an ENUM-enabled PBX (see example from Figure 7 explaining the connection between two organisations' PBXs both double-homed on the PSTN and the Internet), there is a further option for routing the call. Every call to the PSTN is checked by the ENUM-enabled PBX for

[8] Cp. the Vienna University of Technology having +43-1-58801 as central office number with extension 18403 for the DBAI secretary, for instance.

an ENUM entry of the respective called number (all-call query). If an ENUM entry indicates the called party to be reachable by VoIP, the PBX will route the call onto the Internet (again avoiding additional costs on the PSTN).

2.3.3 Internet-to-Internet Session

Another use case is supporting the users' demand for seamless migration from traditional telephony to innovative VoIP services. VoIP users want to be able to connect to other VoIP users and – due to the high penetration figures much more important – to users on the traditional PSTN. Therefore VoIP providers and their customers demand classical telephone numbers to be used with VoIP services. As VoIP services have to be addressed using Internet names (e.g. a H.323 address or a SIP Address-of-Record), the telephone number has to be translated into that type of identifier. Again, this can be done utilising ENUM (see Figure 8).

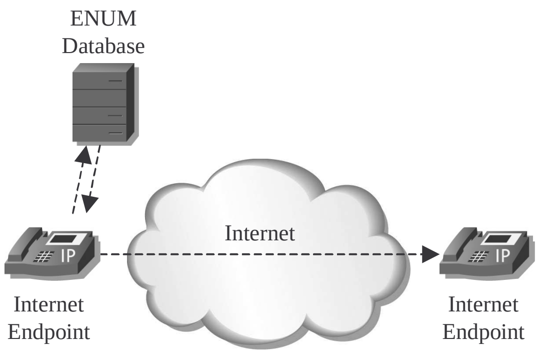

Figure 8: Internet-to-Internet session initiated using ENUM query

2.3.4 Internet (VoIP) Interconnection

A third use case regards the interconnection of VoIP providers, where ENUM can be used to detect a network's ingress point for VoIP calls. This use case utilises a specific form of ENUM, called Infrastructure ENUM (see section 2.10).

2.4 ENUM Mapping Function

The essential function of ENUM is the translation of a telephone number in the international format (so-called E.164 number) to an associated domain name. This mapping is done as shown in Figure 9.

The reasons for the rather strange looking translation are manifold. Going for a straight mapping of telephone number to domain name some could end up with a DNS string of 431580580.example.com, for example. While simple enough for human users this method would cause problems if it were to be used for a sizable number of E.164 numbers, as there are DNS performance constraints associated with the size of the DNS zone file, the frequency of updates and the cache characteristics. Another problem would be the administrative responsibility, as a large number of entities surely would be interested to be the authority for such a large zone [33].

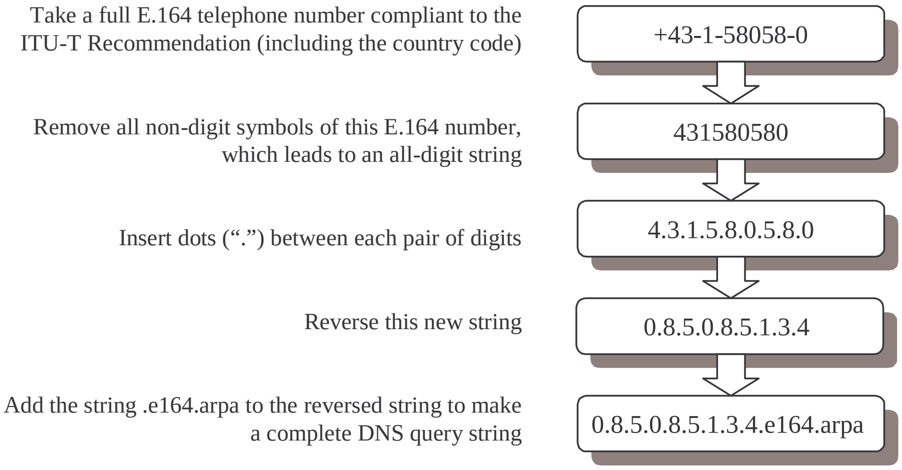

Figure 9: ENUM mapping function

In order to avoid the issues mentioned, the inherent structure of E.164 telephone numbers is used when defining the mapping algorithm. E.164 numbers have the most specific parts to the left (i.e. the +43 in +43-1-58058-0) while Domain Names and IP addresses have the most significant parts to the right (i.e. the .at in dbai.tuwien.ac.at, or the 3 in 128.131.111.3), the order of the E.164 number simply is reversed, resulting in a structure similar to that of domain names. The most significant part of the telephone number (+43) becomes the most significant part of the ENUM domain name (3.4.e164.arpa). In the resulting string 0.8.5.0.8.5.1.3.4.e164.arpa every digit can be treated as a node on the DNS name hierarchy. This has the advantage that each country code as issued by the ITU [35] corresponds to a delegatable DNS domain: +43, the international country code for Austria, can have a corresponding DNS delegation for the zone 3.4.e164.arpa. Within each country code the DNS can be further delegated, as it is possible in the PSTN world of the E.164 numbers.[9]

Figure 10: Hierarchical names, addresses and numbers

As a string like 0.8.5.0.8.5.1.3.4.e164.arpa is not easy to handle for human users, the ENUM translation typically will be done by internal processes of the applications or services, working with the familiar E.164 telephone number in the user interface, if necessary at all.

[9] Cp. specific number ranges in the Austrian telephone numbering plan, e.g. for mobile services (0)6xx or toll-free services (0)800.

2.5 ENUM and the DNS

The structure of ENUM is embedded into the hierarchical structure of the Internet DNS itself. Pursuant to the ITU/IAB ENUM Liaison [36] the ENUM database is to be administered in a hierarchical model with a single international database pointing to single national databases for each telephone country code, that in turn point to authorised service providers. This model is broken down into tiers, with tier 0 being the international level, tier 1 being the national level and tier 2 being the competitive service provider level. With specific ENUM implementations it is possible to also have a tier 3, which is the case when a user (e.g. a company) wants to privately maintain its own ENUM name server.

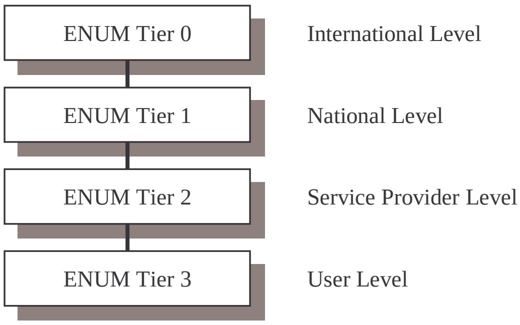

Figure 11: ENUM tiered architecture

The ENUM top-level tier 0 is located in the DNS at a second-level domain of the .arpa domain, named e164.arpa. The .arpa domain as defined by the IETF [32] has been selected to host the e164 subdomain for the ENUM use case by the Internet Architecture Board (IAB) as that domain is dedicated to infrastructure issues and deemed to be well managed, state and secure [66]. The IAB acts as administrative contact for the e164.arpa domain, i.e. being responsible for all administrative issues regarding that domain. In this function IAB has chosen Réseaux IP Européens (RIPE) as its technical contact responsible for the technical operation of the ENUM tier 0. With ITU TSB a telecommunication organisation is involved in the management of the ENUM tier 0 as it has to indicate to RIPE's Network Control Centre (NCC) who the authorised ENUM domain name holder of a member state is. In the case of Austria, the Federal Ministry of Transport, Innovation and Technology (at the time of application for delegation of the ENUM domain[10]) was the responsible authority for the Austrian telephone number space and rightfully informed the ITU TSB on its agreement to delegate the Austrian ENUM domain 3.4.e164.arpa to the Austrian Regulatory Authority for Broadcasting and Telecommunication (RTR-GmbH).
From ENUM tier 1 (e.g. 3.4.e164.arpa in the case of Austria holding the +43 telephone country code) downwards the administration has been defined as national matter. This means that all administrative, legal, technical and commercial issues from ENUM Tier 1 downwards could be implemented differently in different countries. Figure 12 shows the location of the ENUM branch within the global DNS tree structure. The .arpa domain is a generic top-level domain located below the DNS root with three subdomains (.in-addr.arpa, .ip6.arpa and .e164.arpa) defined. Below the ENUM tier 0 .e164.arpa domain the country code specific ENUM domains are to be found. In Figure 12 the ENUM domains for country codes "+1" (North-American numbering plan comprising 19 countries, incl. the United States of America

[10] Since that time, with the new Austrian Telecommunications Act 2003 (TKG 2003) the responsibility for numbering issues largely has moved to the Austrian Regulatory Authority for Broadcasting and Telecommunications (RTR-GmbH).

and Canada), "+43" (Austria), "+44" (United Kingdom) and "+971" (United Arab Emirates) are shown as examples.
Below that level the name servers of competing ENUM service providers may be located (depending on the national ENUM implementation strategies).

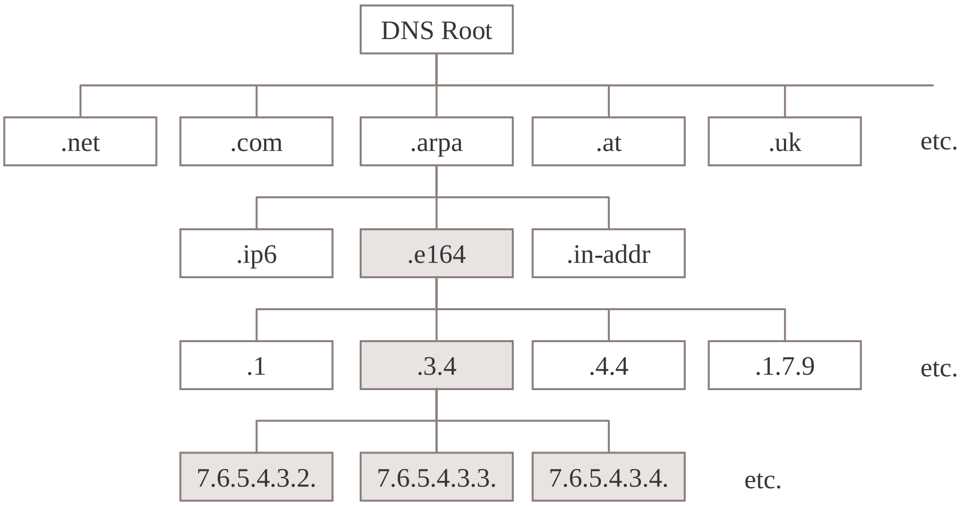

Figure 12: Location of ENUM branch in the DNS tree

It has to be noted that the ENUM branch in principle could be located anywhere on the DNS. However, locating it in a well-defined domain has advantages regarding international coordination and domain management issues as well as stability considerations.[11]

A further aspect are private ENUM implementations. As ENUM simply defines a mapping algorithm from telephone numbers to domain names, private ENUM implementations in any domain are possible, in principle. These may be public or private domains. However, only ENUM deployments as described above guarantee the international interoperability.

2.6 DNS URI Resource Records

The DNS of the Internet [1] allows a collection of URI resource records to be associated with a single DNS name. A set of such records (or individual records) will be returned as answer to a qualified DNS query. DNS resource records in general are described in IETF RFC 1035 [53].

Berkley Internet Name Domain (BIND) [1] is the most popular program used for the operation of name servers with about 80% of name servers worldwide using it.[12] In general, BIND is used to control the name server functionalities. For running a name server with BIND it is necessary to have a configuration file, and files with zone data for the respective domains.

- The configuration file ("named.conf") specifies the attributes of the associated name server. This comprises the working principle (recursive or non-recursive) and possible restrictions on queries and zone transfers. Furthermore it contains those domains the

[11] It has to be mentioned that [23] explicitely defines ENUM to be deployed using the e164.arpa domain.
[12] Cp: http://en.wikipedia.org/wiki/BIND, 05.09.2006.

name server is authoritative for, and the name and location of the files, containing the sets of resource records for the respective domains (called zone files).

- The zone file of one domain lists all hosts or subdomains that are in or below that domain. That is done by means of resource records (RR). It has to be noted that one host or domain name can have multiple RRs, e.g. one RR for the mail server and another one for the Web server. Currently, there are more than 40 types of RRs defined. A resource record by definition has the format `[name] [TTL] [class] Type RR_Data` where the values in brackets, however, do not have to be stated necessarily. `[name]` is the domain or host name, `[TTL]` (time to live) is the maximum time (in seconds) a caching name server is allowed to store the RR set, and for Internet applications the standard class is "IN".

Important categories of resource records used with BIND in the DNS include the following:

- A record or address: maps a hostname to a 32-bit IPv4 address. This is currently the most used resource record on the DNS.
- AAAA record or IPv6 address record: maps a hostname to a 128-bit IPv6 address. This is the IPv6 version of the A record.
- CNAME record or canonical name record: makes one domain name an alias of another. The aliased domain gets all the subdomains and DNS records of the original. The CNAME record is also used for IP address reverse mapping in Classless Interdomain Routing (CDIR).[13]
- MX record or mail exchange record: maps a domain name to a list of mail exchange servers for that domain. MX record contains a FQDN defining the MX host and a 16-bit preference field.
- PTR record or pointer record: maps an IPv4 address to the canonical name for that host. Setting up a PTR record for a hostname in the in-addr.arpa domain that corresponds to an IP address implements reverse DNS lookup for that address. For example (at the time of writing), www.icann.net has the IP address 192.0.34.164, but a PTR record maps 164.34.0.192.in-addr.arpa to its canonical name, referrals.icann.org.
- NS record or name server record: maps a domain name to a list of DNS servers authoritative for that domain. NS record contains a FQDN defining the name server.
- SOA record or start of authority record: specifies the DNS server providing authoritative information about an Internet domain, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone.
- SRV record: is a generalized service location record.
- TXT record: allows an administrator to insert arbitrary text into a DNS record. For example, this record is used to implement the Sender Policy Framework specification.

If an ENUM DNS query is initiated from an e-mail client, most likely a DNS resource record containing an e-mail-address will be chosen while a Voice-over-IP client could look for a SIP address, for instance. It is left to the individual application to determine which particular record (or URI) to search for, to retrieve and how to interpret it.

[13] CDIR is a way to prevent a massive increase in the size of Internet routing tables.

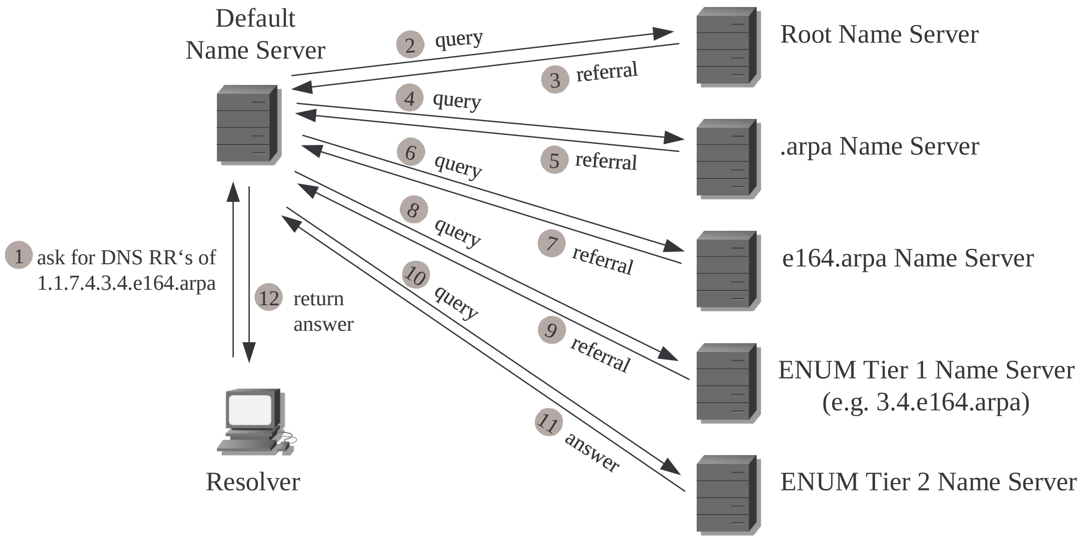

Figure 13: ENUM DNS resolution process

Figure 13 shows a (simplified) retrieval of information stored on an ENUM Name Server. Typically a Resolver (Client) starts a query by questioning its associated Name Server. If the information requested is not to be found in the Name Server's own cache, the Name Server initiates a DNS query in the Public Internet. In our example from Figure 13 the Resolver's default Name Server has no information on how to reach the target server and therefore on its part launches a query questioning the top-level Root Name Server. The Root Name Server (in fact, there are 13 worldwide distributed Root Name Servers) knows about the IP addresses of the top-level Name Servers authoritative for the generic top-level domains (e.g. .com, .net, .biz or .arpa) as well as the country code top-level domains (e.g. .at, .be, .cn or .uk). In our example the Root Name Server returns the address of the .arpa Name Server to the querying Name Server. Next, the .arpa Name Server gets queried and so on. As soon as the requested domain name is found to be hosted on some ENUM Name Server, the requested information (i.e. one or more DNS URI resource records) is retrieved and forwarded to the requesting Resolver. The application (e.g. the e-mail or Voice-over-IP client) then has to choose the appropriate URI and establish the service in the usual way, e.g. by initiating a further resolution of the chosen URI using the DNS again.

2.7 NAPTR Resource Records

The URI resource records used with ENUM are the Naming Authority Pointer (NAPTR) DNS resource records (RR) explained within the Dynamic Delegation Discovery System (DDDS) in a couple of IETF RFCs [44][45][46][47][48]. A resource record is a unit of data in the DNS, defining some attribute for a domain name such as an IP address, a string of text or a mail route. The NAPTR RR [46] was originally created as a way to encode rule-sets in DNS. As a result, delegated sections of a Uniform Resource Identifier (URI) [10] could be decomposed in such a way that they could be changed and re-delegated over time, leading to a resource record including a regular expression to be used by a client program to rewrite a string into a domain name. Regular expressions have the advantage of a high compactness-to-expressivity ratio, i.e. allow encoding a high amount of information in a rather small DNS packet.

The structure of a NAPTR RR is shown in Figure 14 below.

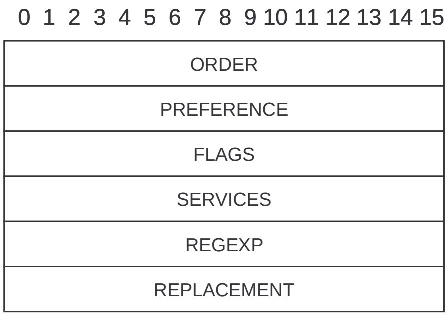

Figure 14: Packet format for NAPTR resource record

The following components are contained in the NAPTR fields. It has to be noted that not all fields defined for NAPTR RRs are applicable for the ENUM use case. This is reflected in the following descriptions:

- **Order field:** specifies the order in which multiple NAPTR records must be processed. As no ordered lists are used within the ENUM field trials, this field is not applicable.
- **Preference field:** determines the processing order when multiple NAPTR records have the same order value.
- **Flags:** to modify the actions of further DNS lookups. It is up to the application (ENUM, for instance) to specify, how this flags should be defined, which ones are terminal and which are not. [23] specifies just one flag for ENUM, named “U”, meaning that this rule is the last one and that the output of the rule is a URI.
- **Services field:** specifies the resolution protocol and service. Again it is up to the application to specify the values found in this field. The service parameters for ENUM are defined in [23] as follows:

```
service_field = “E2U” 1*(enumservice)
enumservice = “+” type 0*(subtype)
type = 1*32(ALPHA/DIGIT)
subtype = “:“ 1*32(ALPHA/DIGIT)
```

  See also list of ENUMservices in section 2.8.

- **Regular expression (REGEXP) field:** allows query client to rephrase the original request in a DNS format. In the case of ENUM the REGEXP always produces a URI.
- **Replacement field:** defines the next DNS query object. This field is not used within ENUM.

An example of a NAPTR RR for a given ENUM domain may look as follows.

Listing 1: Example of NAPTR RRs

```
$ORIGIN 6.0.3.8.5.0.8.5.1.3.4.e164.arpa.
IN NAPTR 10 10 “u” “E2U+voice”“!ˆ^.*$!sip:reichinger@sip.rtr.at!”.
IN NAPTR 10 10 “u” “E2U+email”“!ˆ^.*$!mailto:kurt@hotmail.com!”.
IN NAPTR 10 20 “u” “E2U+voice”“!ˆ^(.*$)$!tel:+436643504516!”.
```

The NAPTR RR from Listing 1 contains information associated with the ENUM domain 6.0.3.8.5.0.8.5.1.3.4.e164.arpa, i.e. the E.164 telephone number +43-1-58058-306, and three referrals to other communication identifiers. Those are

- a SIP URI for voice communication (VoIP),
- an E-Mail URI for e-mail communication, and
- a further telephone number for voice communication (PSTN).

Two of those communication identifiers are to be used for a voice service indicated by "E2U+voice" in the service field. The preferred method for voice communication in the example above is to use the SIP URI indicated by a "10" in the preference field of the associated NAPTR RR, while the telephone number is only second choice for voice communication indicated by a "20" in the preference field. This means that the owner of the telephone number +43-1-58058-306 (i.e. the author) has stated in ENUM that for voice communication he prefers to be contacted using Voice over IP, using the SIP protocol and the given SIP URI. Furthermore, an e-mail URI is provided for e-mail communication.

A query for DNS resource records associated with a certain domain may be performed using appropriate tools as DIG or NSLOOKUP. Alternatively online tools with similar capabilities may be used. Listing 2 shows a typical example from a DIG services website[14] when the domain queried is used for ENUM purposes.

Listing 2: Result of a DIG query for the ENUM domain associated with the author's mobile telephone number

```
; <<>> DiG 9.3.1 <<>> 6.1.5.4.0.5.3.4.6.6.3.4.e164.arpa ANY
 ;; global options:  printcmd
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40796
 ;; flags: qr rd ra; QUERY: 1, ANSWER: 8, AUTHORITY: 0, ADDITIONAL: 0

 ;; QUESTION SECTION:
 ;6.1.5.4.0.5.3.4.6.6.3.4.e164.arpa. IN ANY

 ;; ANSWER SECTION:
 6.1.5.4.0.5.3.4.6.6.3.4.e164.arpa. 1200      IN SOA dns1.my-enum.at. hostmaster.my-
enum.at. 502081017 1200 3600 604800 600
6.1.5.4.0.5.3.4.6.6.3.4.e164.arpa. 1200 IN NAPTR 100 10 "u" "E2U+web:http"
"!^.*$!http://members.chello.at/reichinger/!" .
 6.1.5.4.0.5.3.4.6.6.3.4.e164.arpa. 1200      IN NAPTR 100 10 "u" "E2U+email:mailto"
"!^.*$!mailto:kurt.reichinger@rtr.at!" .
 6.1.5.4.0.5.3.4.6.6.3.4.e164.arpa. 1200      IN NAPTR 100 10 "u" "E2U+email:mailto"
"!^.*$!mailto:kurt.reichinger@chello.at!" .
 6.1.5.4.0.5.3.4.6.6.3.4.e164.arpa. 1200      IN NS  dns1.my-enum.at.
 6.1.5.4.0.5.3.4.6.6.3.4.e164.arpa. 1200      IN NS  dns2.my-enum.at.

 ;; Query time: 2095 msec
 ;; SERVER: 127.0.0.1#53(127.0.0.1)
 ;; WHEN: Thu Jul  6 19:22:50 2006
 ;; MSG SIZE  rcvd: 508
```

Besides the specific records used for ENUM, that domain also contains SOA and NS records (see section 2.6).

[14] Cp. http://www.kloth.net/services/dig.php, 06.07.2006.

2.8 ENUMservices

The data stored in the NAPTR RRs are basically simple rewrite rules, with the ENUM domain name being the unique key to request these rules. The range of services that potentially could be made available behind an ENUM domain name is vast and open for developers. The services currently envisaged for use within ENUM are named ENUMservices. ENUMservices can be distinguished in proposed and already registered ENUMservices.

2.8.1 Originally proposed ENUMservices

The services originally proposed for ENUM implementations are to be found in a Technical Specification (TS) [22] published by the European Telecommunications Standardisation Institute (ETSI). These services are listed below as an overview (see Table 1). However, as already mentioned, this list is extendible with additional ENUMservices should demand occur. An example will be explained later with the new ENUMservice "foaf" tabled as an Internet-Draft by the author (see Section 8.6.1 and Annex 1).

ENUMservices for interactive media-stream exchange		
voice:sip	voice:h323	voice:tel
video:sip	video:h323	video:tel
ENUMservices for discrete (non-session related) messages		
email:mailto	fax:tel	ifax:mailto
sms:tel	sms:sip	sms:sips
sms:mailto		
ems:tel	ems:sip	ems:sips
ems:mailto		
mms:tel	mms:sip	mms:sips
mms:mailto		
ENUMservices for information source		
web:http	web:https	ft:ftp
ENUMservices for service resolution services		
sip	h323	pres
ENUMservices for session-oriented message-exchanges		
tp:tel	im:sip	
ENUMservices for instant information display – announcement		
ann:sip	ann:h323	ann:tel
ann:http	ann:ftp	
ENUMservice for redirection		
Enum		
ENUMservices for location information		
loc:http		
ENUMservices for public key information		
key:ldap	key:http	

Table 1: ENUMservices proposed by ETSI

ETSI suggests that all ENUMservice fields should be in the format `type:subtype` as defined in [23] (see above: services field), with the exception of the ENUMservices sip, h323 and pres, which have only a `type`.

Each of the groups of ENUMservices listed above indicates how an associated URI should be interpreted by the application initiating the ENUM DNS query. This means ENUM is providing information on how a URI is to be used by the client or service that initially requested the ENUM data. This is an important feature of ENUM, as a URI normally does not imply the usage with a specific service, e.g. the URI tel:+43-664-3504516 can be used for e.g. voice telephony, sending of facsimile (fax) or sending of short messages (SMS). ENUM tells a querying party what the ENUM data maintainer’s (i.e. called party’s) preferred usage of some URI is, e.g. voice telephony in the example above.

The following list describes the ENUMservices from Table 1 according to ETSI [22]:

- **ENUMservices for interactive media-stream exchange** indicate that the resource identified by the associated URI is capable of being contacted to provide a communication session during which interactive media streams carrying voice or video data can be exchanged.
- **ENUMservices for discrete (non-session related) messages** is intended to indicate that the associated resource is capable of receiving a discrete (non-session related) message or document. This group may be selected by a querying client if they want to deliver a message (such as a fax) to a correspondent.
- **ENUMservices for information source** indicates that the associated resource can act as a source for information. It acts as the "opposite" of the ENUMservices associated with message sending, in that the latter indicates a source for data whilst the former indicates a sink for data.
- **ENUMservices for service resolution services** is used where the ENUM subscriber wants to use a specialized "Service Resolution Service" above and beyond ENUM. It can be used where the services available depend on factors that cannot be covered in the global ENUM system; for example, the services "advertised" may depend on the person asking, and so requires authentication to be performed before any detailed information is returned.
- **ENUMservices for session-oriented message-exchanges** indicates that the remote resource is capable of engaging in chat sessions (i.e. session-oriented message exchanges). It differs from the ENUMservices for discrete messages in that the latter group implies a session-oriented message exchange, whilst the former group implies a discrete message can be sent to the resource at this contact address.
- **ENUMservices for instant information display – announcement** indicates that an item of instant information from the ENUM Subscriber is to be presented by the ENUM Client to the ENUM User. Those ENUMservices shall point to recorded announcements or to a web page containing at least text and voice data.
- **ENUMservice for redirection** is a special case in that it includes all other ENUMservices within. The goal is to provide a default ENUMservice. This may be used to indicate that the ENUMservices supported with a NAPTR are not specified in any more detail. In effect, the ENUM Subscriber is asserting that this NAPTR supports ALL ENUMservices. However, in practice it means that further processing (by evaluating the regexp and so constructing the associated URI) is needed before the end system can be sure whether to discard the record or not.

- **ENUMservices for location information** indicates that the associated resource can act as a source for location information. It is proposed to provide this information in Geography Markup Language (GML).
- **ENUMservices for public key information** indicates that the associated resource can act as a source for public key information.

2.8.2 Registered ENUMservices

As defined in [23], ENUMservices have to be registered with the Internet Assigned Names and Numbers Authority (IANA) in order to be in conformance with the according usage conditions. Provided an ENUMservice has obtained the necessary approval of the IETF, and the according RFC is published, IANA will register the ENUMservice and make the ENUMservice registration available to the community in addition to the RFC publication itself. The list of registered ENUMservices is published by IANA on its Web site.[15]

In general, ENUMservice specifications contain the functional specification (i.e. what it can be used for), the valid protocols, and the URI schemes that may be returned. Typically, an ENUMservice registration contains the sections illustrated in the following registration template:

- ENUMservice Type
- ENUMservice Subtype(s)
- URI Scheme(s)
- Functional Specification
- Security considerations
- Intended usage: (One of COMMON, LIMITED USE or OBSOLETE)
- Author
- Any other information that the author deems interesting

Table 2 shows the ENUMservices having been successfully registered with IANA up to May 26, 2006.

ENUMservices for interactive media-stream exchange		
voice:tel		
ENUMservices for discrete (non-session related) messages		
email:mailto	fax:tel	ifax:mailto
sms:tel	ems:tel	mms:tel
sms:mailto	ems:mailto	mms:mailto
ENUMservices for information source		
web:http	web:https	ft:ftp
ENUMservices for service resolution services		
Sip	h323	pres
ENUMservices for voice profile for Internet mail		
vpim:mailto	vpim:ldap	

Table 2: ENUMservices registered with IANA

[15] Cp. http://www.iana.org/assignments/enum-services, as of 26.05.2006.

2.8.3 Experimental ENUMservices

As outlined in section 2.8.2 ENUMservices have to be registered with IANA in compliance with a number of registration rules. The only exception to that rules is for ENUMservices used for experimental purposes. The according types and subtypes have to start with the facet “X-“. These elements are unregistered, experimental, and should be used only with the active agreement of the parties exchanging them, according to [23]. Recent developments in the IETF ENUM working group, however, show efforts to ease the procedure for registering innovative ENUMservices [40].

It will be referred to experimental ENUMservices later in this thesis, when a new ENUMservice will be introduced (see section 8.6.1). In order to use the new ENUMservice in application tests prior to an official registration with IANA it is implemented using the “X-“-facet, in the meantime.

2.9 Privacy Issues

2.9.1 General Considerations

Populating the ENUM database means storing data in the DNS. With the DNS being publicly accessible by nature and data within the DNS being freely retrievable, privacy concerns regarding that data are becoming an issue. Comparable to data published on a website on the WWW, data in the DNS can be harvested, aggregated, stored and re-used by other parties, e.g. for generating lists of targets for sending of unrequested information. This could result in being targeted with SPAM (unsolicited e-mail), SPIT (unsolicited VoIP calls), junk fax, junk SMS or other unwanted information. Even after removing the DNS entry itself, copies of the information could still be available on some other place being stored there by a third party. Therefore it is generally not recommendable to populate an ENUM domain with data that should not become publicly available.

2.9.2 Identification, Authentication and Validation in ENUM

In order to tackle privacy concerns and to maintain the integrity of the E.164 telephone numbering plan, identification, authentication and validation are major issues in any ENUM implementation. First, it must not happen that an ENUM domain is under the control of a person that is not the holder of the associated E.164 telephone number. Second, it must not happen that an unauthorised person can alter the data within another person’s ENUM domain.

- **Identification** of an ENUM subscriber means, that some individual interested in having his (communication) details included in ENUM, must be identified by the ENUM service provider. This means that the subscriber has to show, that he is the one he pretends to be. This can be done by e.g. manually checking the subscriber’s identity card or passport, by means of a certificate issued by a trusted third party or other measures.
- **Authentication** of an already known ENUM subscriber is necessary, when the subscriber wants to update data stored in his NAPTR RRs. Typically this is performed by checking a set of subscriber credentials (e.g. username and password) using a web interface.
- **Validation** of someone’s right to use a specific E.164 number is another extremely important (and possibly most difficult) task, as it has to be assured, that only persons with the right to use a specific E.164 number are allowed to have the associated ENUM domain delegated. This task has to be repeated from time to time in order to

assure that the ENUM subscriber still is the holder of the E.164 telephone number. As soon as the ENUM subscriber looses the right to use the E.164 number (i.e. by cancellation of contract with telephony service provider), he also looses the right to use the associated ENUM domain and subsequently the NAPTR resource records have to be deleted.

2.10 User ENUM and Infrastructure ENUM

The original idea and the first implementations of ENUM had a strong focus on end users autonomously configuring their respective entries in the ENUM database. In consequence, ENUM often has been compared with the electronic pendant to a personal business card, enabling users to flexibly alter their ENUM entries according to their personal communication needs and preferences. The user's E.164 telephone number was designated to be the unique key to access that "business card data" electronically. Logically consistent, that flavour of ENUM was named User ENUM. Other end users as well as operators of communication networks or services have the opportunity to retrieve and use that data.

From the very beginning of ENUM there was discussion regarding usage for inter-carrier issues. In the last 12 to 18 months, Infrastructure ENUM [39] [41] has emerged as a second option for the use of ENUM.[16] It allows a carrier to populate the Infrastructure ENUM DNS database with the telephone numbers of all of its own customers independently of User ENUM.

In contrast to User ENUM, the NAPTR RRs in Infrastructure ENUM typically will not contain end customer data (e.g. a customer's SIP AoR or E-mail address), but operator information to be used for e.g. routing or charging (VoIP federation[17] information). Consequently, these data is not controlled and maintained by the end user anymore but by the operator. Infrastructure ENUM has a number of potential use cases, such as routing of IP-based traffic for MMS and SMS (in mobile environments) or for interconnection purposes between IP-based services. In the latter case an operator would populate ENUM with all of its numbers pointing to the operator's network ingress point serving the given E.164 telehone number, i.e. directing all traffic to a defined gateway for better control, accounting and charging of sessions.

For operators ENUM offers a financially interesting option to terminate calls: Calls on the classic PSTN usually have to be handed over to the called party's access provider, thus incurring termination fees. Should a provider determine (by means of an ENUM query) that the called party is (also) reachable on the Internet with a VoIP URI, a call can be routed onto the Internet (using a PSTN-to-IP Gateway) avoiding the above mentioned termination fees in the PSTN world. This of course has the reverse effect on access network operators financially relying on termination fees, as an increasing number of calls may be routed onto the Internet (without the incurrence of such fees). However, with today's low penetration of User ENUM and accordingly low success rates of ENUM queries only a few operators so far implemented all-call ENUM query[18] in their systems.

The introduction of Infrastructure ENUM potentially leads to a large amount of E.164 telephone numbers being available in the Infrastructure ENUM database within a very short time frame, therefore increasing the value of ENUM. As Metcalfe's law states, the value of a

---

[16] In Austria, one of the world's leading countries regarding ENUM implementation, Infrastructure ENUM is already in operation since May 2006. Cp. http://www.enum.at/index.php?id=515&L=9, 20.11.2006.

[17] A VoIP federation is a group of VoIP providers having agreed to exchange traffic.

[18] All-call ENUM query refers to the method of performing an ENUM query for each and every call originated in a network or passing through some network equipment.

network equals approximately the square of the number of users of the system, i.e. n^2. Since a user cannot connect to itself, the actual calculation is the number of diagonals and sides in an n-gon:

$$\text{Value} = n\,(n-1)/2$$

The networking effect of communication technologies and networks described by Metcalfe's law is also valid for the ENUM use case. The more ENUM domains delegated (i.e. telephone numbers stored in the ENUM DNS), the more ENUM queries will be successful and the higher the value of ENUM will become (see Figure 15).

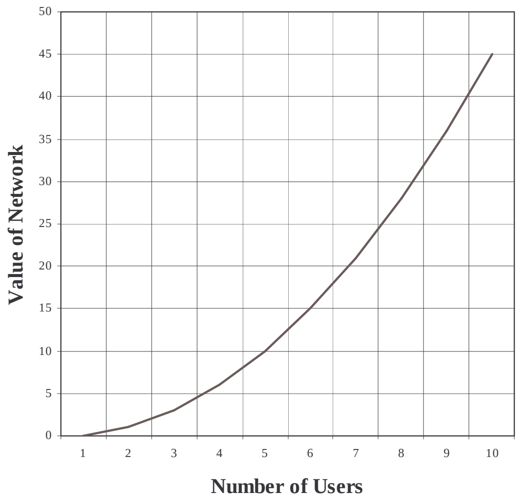

Figure 15: Value of network according to Metcalfe's law

Infrastructure ENUM is planned to be implemented in a different DNS tree than User ENUM, allowing the usage of Infrastructure and User ENUM in parallel. The IETF proposes to have Infrastructure ENUM implemented in a subtree of the .arpa domain, .ie164.arpa, but other trees in principle are possible as well. Especially for an intermediate period, until all administrative and political hurdles for the delegation of the new subdomain are taken, the use of alternative domains or trees is well possible. Respective work is going on in IETF's ENUM[19] and SPEERMINT[20] working groups. Despite these standardisation efforts commercial solutions have emerged in parallel with X-Connect[21] or the AMS-IX SIP Peering[22] being popular examples already in operation.

---

19 Cp. http://www.ietf.org/html.charters/enum-charter.html, 07.10.2006.
20 Cp. http://www.ietf.org/html.charters/speermint-charter.html, 07.10.2006.
21 Cp. http://www.xconnect.net/, 07.10.2006.
22 Cp. http://www.ams-ix.net/services/cug/sip.html, 08.10.2006.

3 Semantic Web

3.1 Introduction

The term “semantic” comes from the Greek expression σημαντικός, *semantikós* (i.e. “significant meaning”), derived from *sema* (i.e. “sign”). In general, *semantics* is the study of meanings. In the case of the Semantic Web, this can be interpreted as adding computer-understandable meaning (semantics) to the content of documents on the World Wide Web (WWW).

The Semantic Web, envisioned by Tim Berners-Lee in a frequently cited article [8] from 2001, has been in a steady process of being further defined and developed since. Today, the Semantic Web standardisation activities are bundled at the World Wide Web Consortium (W3C)[23] with Berners-Lee currently heading that institution.

The Semantic Web represents a concept in which machine-readable content builds the base for a whole set of new applications on the Internet. The Semantic Web is explicitly designed to enable adding of computer-understandable meaning to documents on the WWW. The core element of the Semantic Web is a system of structured semantic knowledge called metadata. In order to animate this system, new languages for ontologies, rules and proofs are needed.

Today’s WWW is focused on document presentation (primarily coded in Hyper Text Markup Language; HTML[24]) aimed at human users, i.e. Web pages being designed to be read by human users. The Semantic Web is a project aiming to make Web pages understandable by computers, so that they can search websites and perform actions in a standardised way. The potential benefit is that computers can harness the enormous network of information and services on the Web much better than a human user. A computer could, for example, automatically find a hotel and a flight for a given conference in line with the user’s preferences (e.g. 4-star hotel, close to conference venue, non-smoking room, economy class flight, aisle seat, vegetarian menu, etc.), and book both fitting in with the user’s schedule (e.g. derived from Microsoft™ Outlook).

The Semantic Web’s architecture also known as the Semantic Web “layer cake” or Semantic Web stack, is shown in Figure 16 (original version, 2001[25]), Figure 17 (W3C updated version, 2006[26]) and Figure 18 (further elaborated version presented by T. Berners-Lee at AAAI’06, 2006[27]) all three visualising how the major building blocks are expected to sit above each other.

[23] http://www.w3.org/2001/sw/Activity
[24] http://www.w3.org/TR/html401/
[25] Cp. http://www.w3c.tut.fi/talks/2002/0923sw-vtt-on/slide15-0.html, 10.10.2006.
[26] Cp. http://www.w3.org/2004/Talks/0209-Helsinki-IH/105.html, 10.10.2006.
[27] Cp. http://www.w3.org/2006/Talks/0718-aaai-tbl/Overview.html, 10.10.2006.

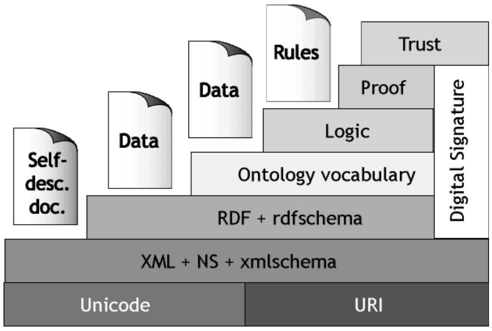

Figure 16: Semantic Web Architecture (original version, 2001)

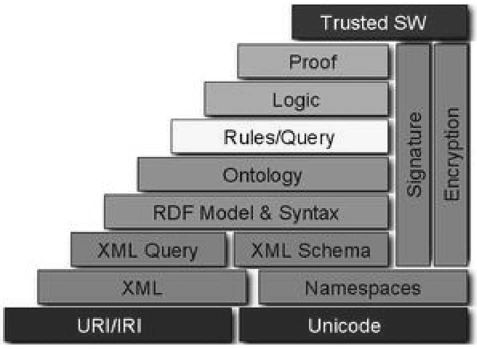

Figure 17: Semantic Web architecture (W3C's updated version, 2006)

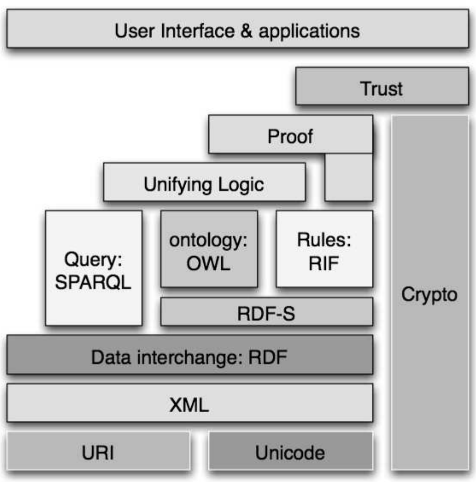

Figure 18: Semantic Web architecture (version by Tim Berners-Lee, AAAI'06, 2006)

In the following a closer look will be taken on each of the layers and associated functionalities, describing the full picture of the Semantic Web's architecture. For the purpose of this description the original Semantic Web architecture (see Figure 16) will be used as normative reference, however without missing out on new considerations from the further elaborated versions.

3.2 Layer 1: Identifiers

The basement layer of the Semantic Web is built by the Uniform Resource Identifier[28] (URI) and Unicode (cp. Figure 16 to Figure 18).

3.2.1 Uniform Resource Identifier

In general, a Uniform Resource Identifier (URI) [10] is used to give "something" on the Web a name. Anything that has a URI therefore can be said to be "on the Web" and the possibilities of assigning names are vast: a person, a document, an object, a trip to Rome and anything else one can think of, all can have a URI.
URIs are highly decentralised identifiers without any control of who creates them and how they are used. This leads to a situation where everybody can create and use URIs inevitably ending up with multiple URIs representing the same thing as well as different things being described with the same URI.

It is typical for the World Wide Web to create a Web page and to use a URI to name it. The Web page usually describes the object and explains that the URL of that Web page represents this object. For example one could create the URI http://www.example.com/mythesis which represents my copy of the thesis. Therefore the URI can be understood as not describing a Web page anymore, but the object ("my thesis") itself. So the URI http://www.example.com/mythesis becomes the name of the object "my thesis".

A well-known example of a URI is the Uniform Resource Locator (URL) or http: URI [9][24], which is an address enabling one to visit a web page or – more exactly – that lets a computer locate a specific resource (the Web site) on the Internet. Other familiar URIs are the mailto: URI to encode e-mail addresses [31], the ftp: URI for file transfer [57] or the sip: URI for usage with the Session Initiation Protocol [64].
Another URI is the Uniform Resource Name (URN) [52], a URI that identifies a resource by name in a particular namespace. A URN can be used to talk about a resource without implying its location or how to dereference it. For example, the URN urn:ISBN:0-262-01210-3 is a URI that allows one to talk about a book (in the case of the ISBN example from above it is the title from Grigoris Antoniou and Frank van Harmelen, A Semantic Web Primer, The MIT Press). However, a URN does not provide any information on where and how to obtain a copy of that book, for instance.

3.2.2 Unicode

The second building block of the Semantic Web's base layer is Unicode. Unicode is an industry standard designed to allow text and symbols from all natural languages to be consistently represented and manipulated by computers. Unicode characters can be encoded using any of several schemes termed Unicode Transformation Formats (UTF). The Unicode Consortium[29] aims to replace existing character encoding schemes with Unicode in order to reach compatibility with multilingual environments. Today, the Unicode standard is implemented in many technologies, including XML, the Java programming language, and modern operating systems.[30]

---

[28] The W3C's Semantic Web stack consists URI and IRI. IRI stands for Internationalised Resource Identifiers, enabling people to name resources in their own languague. With few exceptions, the natural scripts of the world's languages use characters other than A-Z. By expanding allowed characters from a subset of US-ASCII to the Universal Character Set (Unicode/ISO 10646), IRIs allow content developers and users to identify resources in their own languages.
[29] Cp. http://www.unicode.org/, 20.08.2006.
[30] Cp. http://en.wikipedia.org/wiki/Unicode, 20.08.2006.

3.3 Layer 2: Documents

3.3.1 Extensible Markup Language

The Extensible Markup Language (XML) [73][74] was originally designed to send documents across the Web in a simple way.[31] Anyone can design its own document format and write documents using that format. XML enables the user to include markup to enhance the meaning of a document. Depending on the client program the markup and its associated meaning will be used (or not) accordingly. Each program or application is free to choose which markups of an XML document to use and how to interpret them. Thus, sharing an XML document adds meaning to the content; however, assuming that both parties know and understand the element names. For example, if some person labels something a `<name>Reichinger</name>` and another person labels that field `<nom>Reichinger</nom>`, there is no way a machine will know those two mean the same thing unless Semantic Web technologies like ontologies are added (see section 3.5).

As introduced above, the task of including additional meanings to a document is performed by attaching descriptive "tags" to certain portions of the document. A full set of tags (opening tag + content + closing tag) is called an element. The start tag consists of a name written between angle brackets, like `<tag>`; the end tag consists of the same name written between angle brackets, but with a slash preceding the name, like `</tag>`. In addition to content, an element can also contain attributes. Attributes are name-value pairs included in the start tag after the element name, like `<tag attribute_1="5" attribute_2="unit">Content</tag>`. Attribute values must always be quoted, using single or double quotes. Each attribute name should appear only once in any element.

The basic syntax for one element in XML is `<name attribute="value">content</name>`. An example XML code fragment illustrates the usage of XML terms in Listing 3 below.

Listing 3: XML code fragment illustrating basic XML concept

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
   <title>Telecommunications and the Next Generation Web: Introducing ENUM to the
Semantic Web</title>
   <author>K. Reichinger</author>
   <supervisor>R. Baumgartner</supervisor>
   <supervisor>G. Gottlob</supervisor>
   <type>Ph.D. Thesis</type>
   <organisation>Vienna University of Technology</organisation>
   <year>2006</year>
</document>
```

The first line is the XML declaration stating what version of XML (i.e. version 1.0) is used. In addition it contains information about character encoding (i.e. UTF-8); information about external dependencies may be found there as well. The rest of the code fragment from Listing 3 above consists of nested elements (with start tag, content and end tag) as described above.

3.3.2 XML Namespaces

XML Namespaces (XML NS) enable one to give a URI to each element and attribute. This opens the opportunity for everyone to create his/her own tags and mix them with tags made by others. XML namespaces enable the same document to contain XML elements and

[31] Cp. http://en.wikipedia.org/wiki/XML, 20.08.2006.

attributes taken from different vocabularies, without any naming collisions occurring. XML Namespaces therefore help to secure the semantic interoperability among metadata vocabularies.
A namespace is declared using the reserved XML attribute xmlns, the value of which must be a URI (Uniform Resource Identifier) reference, e.g. xmlns=http://www.w3.org/1999/xhtml defining the Extensible Hypertext Markup Language. It has to be noted, however, that the URI is not actually read. Instead, the URI is simply treated by an XML parser as a string. For example, http://www.w3.org/1999/xhtml itself does not contain any code; it just describes the xhtml namespace to human readers.

3.3.3 XML Schemas

XML schemas are used to describe a type of XML document. Typically, this is expressed in terms of constraints on the structure and content of documents beyond the basic XML syntax constraints. Examples of languages developed specifically to express XML schemas are the Document Type Definition (DTD) language (with rather limited capability), or the more expressive XML Schema and RELAX NG. Documents are only considered valid if they satisfy the requirements of the schema they have been associated with. These requirements typically include such constraints as:

- Elements and attributes that must/may be included, and their permitted structure
- How character data is to be interpreted, e.g. as a number, a date, a URL, a Boolean, etc

An example of a simple Schema to describe a country is illustrated in Listing 4 below.

Listing 4: Simple Schema to describe a country

```
<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="country" type="Country"/>
 <xs:complexType name="Country">
  <xs:sequence>
   <xs:element name="name" type="xs:string"/>
   <xs:element name="population" type="xs:decimal"/>
  </xs:sequence>
 </xs:complexType>
</xs:schema>
```

An example of an XML document that conforms to this schema is given below in Listing 5. The code fragment simply says that there is a country named Austria with a population of 8.22 million.

Listing 5: XML code fragment conforming to Schema from Listing 4

```
<country
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="country.xsd">
  <name>Austria</name>
  <population>8220000</population>
</country>
```

3.3.4 XML Query

As explained in previous sections, XML is a markup language capable of describing information of data sources by labelling the respective content. In order to perform queries on those data sources (e.g. documents, databases or repositories) a specific XML query language is needed. Such a query language can express queries across different kinds of data, may it be directly stored in XML or viewed as XML via some middleware. One example of an XML query language has been specified by the W3C XML Query Working Group: XQuery 1.0[32] is designed to be broadly applicable across many types of XML data sources.

XQuery allows one to extract and manipulate data from XML documents or any other data source that can be viewed as XML. XQuery uses XPath expression syntax to address specific parts of an XML document. It works with a SQL-like "FLWOR expression" for performing joins. Such a FLWOR expression is constructed from the five clauses providing the name: FOR, LET, WHERE, ORDER BY, and RETURN. The language further provides syntax allowing new XML documents to be constructed. Where the element and attribute names are known in advance, an XML-like syntax can be used; in other cases, expressions referred to as dynamic node constructors are available.

XQuery 1.0 however, does not include features for updating XML documents or databases, as is done by XUpdate. It also lacks full text search capability. Both these features may be included in a subsequent version of the language.

3.4 Layer 3: Statements

3.4.1 Resource Description Framework

The Resource Description Framework (RDF) as defined by the W3C [71] brings the possibility to make statements that are machine-processable.[33] Of course a computer still does not “understand” what someone actually said, but it knows how to deal with it. For example, someone (a machine or agent) could search the web for all movie ratings and provide an average rating for each film being put back onto the Internet. Another website could take that information and create a “Worst Movies Ever” page. In Semantic Web speaking this means that the agents involved later can make logical inferences, based on metadata, to perform tasks.

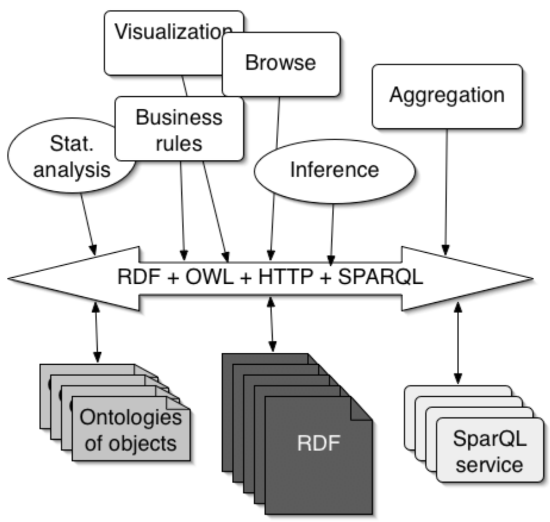

Figure 19: RDF data bus – from http://www.w3.org/2006/Talks/0718-aaai-tbl/Overview.html#(6)

[32] Cp. http://www.w3.org/TR/xquery/, 23.09.2006.
[33] Cp. e.g. http://www.w3schools.com/rdf/default.asp, 07.09.2006.

Contrary to the impression possibly created by Figure 19 above, RDF itself is quite simple. Each RDF statement consists of three parts: a subject, a predicate and an object. The structure therefore is very much like a sentence, except that almost all the words are URIs. An example of such an RDF statement is illustrated in Listing 6 below.

Listing 6: RDF statement in Notation3

```
<http://kurt-reichinger.at/>
<http://like.example.com/terms/reallylikes>
<http://holidays.example.com/destinations/Greece>
```

The example says that a person named Kurt Reichinger really likes holidays in Greece. For simplification this statement is written in a language called Notation 3 [11] focusing on human readability, but the official RDF specification defining an XML representation of RDF is just a little more complex. An example of an XML RDF statement is listed in Listing 7 below.

Listing 7: XML RDF code fragment

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/05/contact#">
   <contact:Company
   rdf:about="http://www.w3.org/Organization/contact#DBAI-Group">
     <contact:name>DBAI-Group</contact:name>
     <contact:mailbox rdf:resource="mailto:sek@dbai.tuwien.ac.at"/>
     <contact:phone>+43 (1) 58801 18403</contact:phone>
   </contact:Company>
</rdf:RDF>
```

The code fragment in the example from Listing 7 basically makes a statement about a resource, which in this case is the organisation http://www.w3.org/Organization/contact#DBAI-Group. First, it is stated that the code has to be interpreted as XML. Furthermore the code uses RDF elements and attributes from two namespaces. The organisation can be identified by the URI http://www.w3.org/Organization/contact#DBAI-Group; its name is DBAI-Group, its e-mail address is sek@dbai.tuwien.ac.at, and its phone number is +43 (1) 58801 18403.

RDF identifies resources using URIs, and describes these resources with properties and property values. The combination of resource, property and property value forms an RDF statement as described above, i.e. some sort of sentence consisting of subject, predicate and object.

- A resource in this context is anything that can have a URI, e.g. http://www.thesis.at.
- A property on the other hand is a resource that has a name, e.g. "author".
- A property value is the value of a property, such as "Kurt Reichinger" or “http://www.mythesis.at", i.e. a property value can be another resource.

An example statement using resource, property and property value from above could be “The author of http://www.thesis.at is Kurt Reichinger”.

- The subject of that statement above is http://www.thesis.at.
- The predicate is: author.
- The object is: Kurt Reichinger.

RDF is based on a directed graph model, i.e. an RDF statement can be described by means of a syntax-neutral graph (see Figure 20).[34] To better illustrate coherences, RDF statements are often additionally represented by such graphs ranging from rather simple (as in Figure 20 below) to very complex figures.

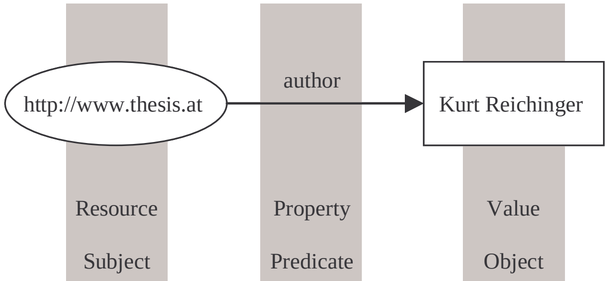

Figure 20: Syntax-neutral graph describing the statement from above

Although we can assume RDF not to be a highly complex language, not too many people can be expected to learn and use this language anytime soon. If the Semantic Web is wanted to be up and running within a decent timeframe, the semantic annotation of content for the Semantic Web must be provided by other sources, i.e. databases already deployed. RDF is perfectly suited to publish these databases' information on the Web. As discussed before, every single item of the databases will get a URI, enabling intelligent systems to fit the available data together. For easy publication of legacy database content appropriate translation tools creating RDF data have to be implemented. An application example will be shown in section 8.4.

RDF started as framework for metadata providing interoperability between applications exchanging machine-understandable information on the Web. RDF emphasizes facilities to enable automated processing of Web resources and as such provides the basic building blocks for supporting the Semantic Web. RDF metadata can be used in a variety of application areas; for example: in resource discovery to provide better search engine capabilities; in cataloguing for describing the content and content relationships available at a particular Web site, page, or digital library; by intelligent software agents to facilitate knowledge sharing and exchange; in content rating; in describing collections of pages that represent a single logical "document"; for describing intellectual property rights of Web pages, and in many others. RDF with digital signatures will be one of the keys to building the "Web of Trust" for electronic commerce, collaboration, and other applications.

3.4.2 RDF Query

RDF as described in Section 3.4.1 needs a query language in order to exploit its full potential. SPARQL as defined by the W3C [81][80] stands (recursively) for SPARQL Protocol and RDF Query Language.[35] SPARQL is the successor to other RDF query languages, e.g. RDF Query Language, RDQL [79].

---

[34] Cp. e.g. http://www.dbis.informatik.uni-frankfurt.de/~tolle/RDF/DBISResources/RDFIntro.html, 05.09.2006.
[35] Cp. http://www.w3.org/TR/rdf-sparql-query/, 23.09.2006.

The following example shows how to find the names of all European capitals using SPARQL (see Listing 8).

Listing 8: SPARQL code fragment

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital.
  ?y abc:countryname ?country.
  ?x abc:isCapitalOf ?y.
  ?y abc:isInContinent abc:europe.
}
```

In SPARQL, variables are labeled with a "?" prefix. In the example from Listing 8 all those variable data for ?capital und ?country is returned that matches the four RDF triples given. The prefix "abc" defined in line 1 of the SPARQL code fragment from above is used instead of "http://example.com/exampleOntology#" for better readability.

3.5 Layer 4: Schemas and Ontologies

The idea of using already existing data stored in already existing databases seems striking, but it has to be considered that – apart from privacy and data protection concerns – today's database systems are far from being optimized for the Web. Any agent optimised to understand a certain database structure may fail at a system with a different structure. Similar problems arise if the internal structure (e.g. the way of rating movies) of the database is altered. An example of a system depending on well defined structures and contents of databases is the Austrian website Geizhals[36] that provides price comparison from a multitude of electronic equipment shops. All shops participating have to provide their data in a structure defined by Geizhals, the operator of the price comparison portal. Although this is not what Semantic Web proponents envisage, it is showing the potential of seamless machine-readable data on the Next Generation Web.

An even bigger problem is the fact that computers (as well as humans, by the way) can face severe problems when trying to figure out what a specific term found in a database could mean or how it should be used. In fact all the URIs are rather useless, as long as their meaning remains undescribed.

Schemas and ontologies are ways to describe the meaning and relationships of terms. Such a description (again implemented in RDF) enables computer systems to use terms more easily and to convert between them. The schema/ontology layer builds a central unit of the Semantic Web Architecture as simple descriptive to complex classificatory schemas are created and registered there, enabling agents to interpret data, make inferences and perform tasks.[37]

[36] http://www.geizhals.at

[37] Cp. http://www-128.ibm.com/developerworks/web/library/wa-semweb/, 20.08.2006.

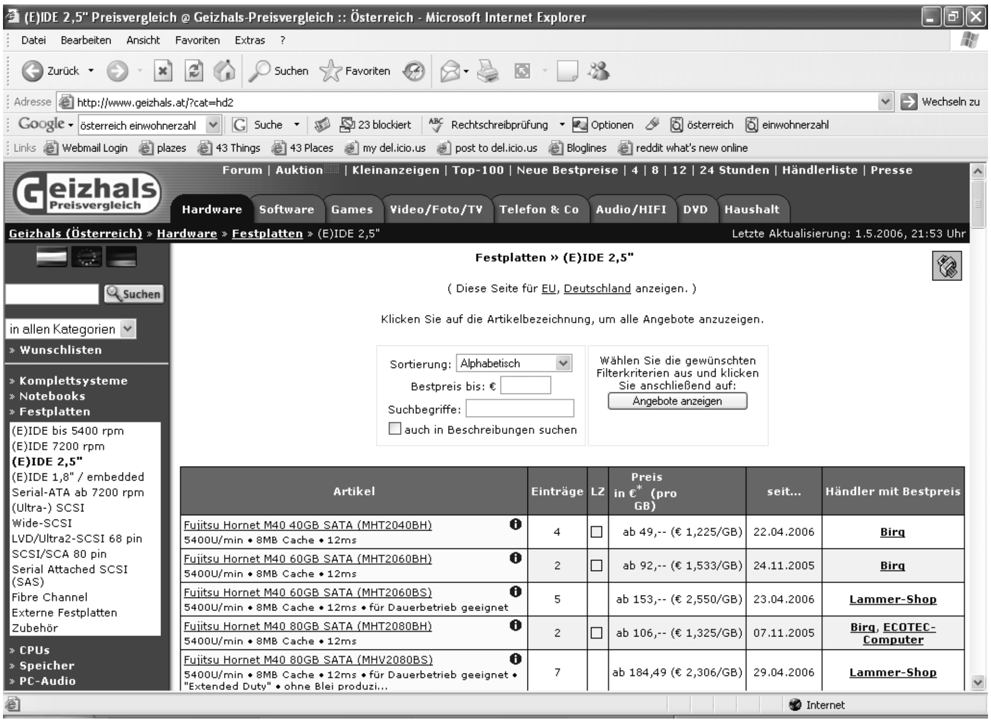

Figure 21: Screenshot from Geizhals Web site, from 01.05.2006

3.5.1 Schemas

In computer science, a Schema can be described as a model enabling one to define structure, content and even semantics of documents. A Schema can also be used for describing the structure of a database, for instance.[38]

One example used on the Semantic Web is RDF Schema [78], which is a semantic extension of RDF.[39] It provides mechanisms to describe groups of related resources and the relationships between those resources. The RDF Schema class and property system is similar to the type systems of object-oriented programming languages such as the Java language. RDF differs from many such systems. Rather than define a class in terms of the properties of its instances, the RDF vocabulary description language describes properties in terms of which classes of resource the properties apply to. Both RDF and RDF Schema can be serialised using XML and XML Schema. The existence of standards for describing data (RDF) and data attributes (RDF Schema) enables the development of a set of readily available tools to read and exploit data from multiple sources. The degree to which different applications can share and exploit data is sometimes called syntactic interoperability. The more standardised and widespread these data manipulation tools are, the higher the degree of syntactic interoperability, and the easier and more attractive it becomes to use the Semantic Web approach as opposed to a point-to-point integrated solution.[40]
Syntactic interoperability is all about parsing data correctly. It requires mapping between terms, which in turn requires content analysis. Content analysis requires formal and explicit specifications of domain models defining terms used and their relationships. Such formal domain models are sometimes called ontologies.

---

[38] Cp. http://en.wikipedia.org/wiki/Schema, 05.09.2006.
[39] Cp. http://en.wikipedia.org/wiki/Rdf_schema, 05.09.2006.
[40] Cp. http://www.hpl.hp.com/semweb/sw-technology.htm, 05.09.2006.

3.5.2 Ontologies

Ontologies are often described in rather abstract terms used by Artificial Intelligence experts[41], but nevertheless have established themselves in recent years on the WWW for the description of specific domains. Web Ontologies meanwhile range from large complex taxonomies e.g. categorising Web Sites to brief descriptions of e.g. products or people.[42] Various disciplines have been and/or are currently in the process of developing ontologies for their specific needs.
The main purpose of such an ontology is to define a common vocabulary in order to better share information in a domain. Ontologies define machine-interpretable data models in terms of classes, subclasses and properties. An ontology therefore can also be described as a network of concepts, relationships and constraints that provide context for data and information as well as processes. Some further reasons for creating a domain ontology are to be found in [54] and listed below.

- Share common understanding of the structure of information among people or software agents
- Enable reuse of domain knowledge
- Make domain assumptions explicit
- Separate domain knowledge from the operational knowledge
- Analyse domain knowledge

Typically, the creation of a new domain-specific ontology requires a tight cooperation between specialists of the respective domain (e.g. from life sciences) and the Semantic Web.

3.5.3 Web Ontology Language

One popular language to express ontologies is the W3C's Web Ontology Language (Ontology Working Language; OWL) [76].[43] OWL disposes of more vocabulary to describe properties and classes than RDF or RDF Schema: It can describe relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, more types of properties and characteristics of properties (e.g. symmetry) and others. OWL is written in XML, allowing OWL information to be easily exchanged between different types of computers using different types of operating system and application languages.

Listing 9 describes how OWL can be used to describe a stadium, for instance. In the example this is done by defining a stadium with a couple of domain-specific identifiers. The example from Listing 9 works with the fictitious OWL `class:Stadium` for the description of stadiums. The properties used for describing a stadium are OWL `property:name` (specifying the stadium name, e.g. Villa Park), `property:home-team` (specifying the stadium home team, e.g. Aston Villa FC), `property:location` (specifying the stadium location, e.g. Birmingham, UK), `property:latitude` and `property:longitude` (specifying the geographic coordinates of the stadium).

---

[41] Cp. Gruber, 1993, describing an ontology as explicit formal specifications of the terms in a domain and relations among them.
[42] Cp. http://www.schemaweb.info, 05.09.2006.
[43] Cp. http://en.wikipedia.org/wiki/Web_Ontology_Language, http://www.w3schools.com/rdf/rdf_owl.asp, and http://www.w3.org/TR/owl-guide/, 05.09.2006.

Listing 9: OWL example describing a stadium

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xml:base="http://www.daml.org/2001/10/html/stadium-ont">

<owl:Ontology rdf:about="">
  <owl:versionInfo>$Id: stadium-ont.daml,v 1.1 2006/01/01 18:00:00 mdean Exp
$</owl:versionInfo>
  <rdfs:comment>Stadium</rdfs:comment>
</owl:Ontology>

<rdfs:Class rdf:ID="Stadium">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#name"/>
      <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#location"/>
      <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#home-team"/>
      <owl:allValuesFrom rdf: resource="http://www.w3.org/2001/XMLSchema#string"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</rdfs:Class>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#latitude"/>
      <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#longitude"/>
      <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
    </owl:Restriction>
  </rdfs:subClassOf>

<owl:DatatypeProperty rdf:ID="home-team"/>
<owl:DatatypeProperty rdf:ID="latitude"/>
<owl:DatatypeProperty rdf:ID="location"/>
<owl:DatatypeProperty rdf:ID="longitude"/>
<owl:DatatypeProperty rdf:ID="name"/>

</rdf:RDF>
```

3.5.4 OWL Sublanguages

OWL provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users. The W3C describes the three OWL flavours in an overview document [75] as follows:

- OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies. Owl Lite also has a lower formal complexity than OWL DL.

- OWL DL supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL (see section 3.6.1).

- OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

Each of these three sublanguages is an extension of its simpler predecessor, both in what can be legally expressed and in what can be validly concluded. The following Figure 22 shows the set of relations holding, while their inverses do not.

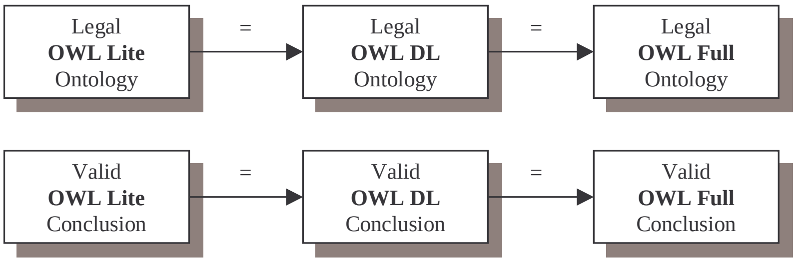

Figure 22: Relations between OWL Lite, OWL DL, and OWL Full

Figure 22 explains that every legal OWL Lite ontology is a legal OWL DL ontology, and every legal OWL DL ontology is a legal OWL Full ontology. The same holds for conclusions, with every valid OWL Lite conclusion being a valid OWL DL conclusion, and every valid OWL DL conclusion being a valid OWL Full conclusion.

3.6 Layer 5: Logic

The next layer is the logic layer, which introduces ways of writing logic into documents. Logic[44], coming from Classical Greek λόγος (logos), originally means the word, or what is spoken. Nowadays it is rather used for the meaning of thought or reason. Logic is about finding criteria for the evaluation of arguments, finally enabling one to distinguish logical

[44] Cp. http://en.wikipedia.org/wiki/Description_logic, 07.09.2006.

arguments from flawed ones. As a formal science, logic investigates and classifies the structure of statements and arguments, both through the study of formal systems of inference and through the study of arguments.

The Semantic Web's logic layer allows introducing logic into documents. Although the introduction of schemas and ontologies (Layer 4; see Section 3.5) already means an important step towards a machine-processable web, it is of further significance that logical principles can be stated. Such logical statements allow computers (or agents) to make inferences[45] and deductions[46], e.g. checking of a document against a set of rules regarding self-consistency of data, or the resolution of a query by converting unknown terms into known ones.

Logic, particularly predicate logic (a.k.a. first-order logic) to a large extent is the foundation of knowledge representation (KR). KR has been studied long before the emergence of the Semantic Web, particularly in artificial intelligence and philosophy. The importance of logic can be traced back to a couple of facts according to [2]:

- Providing a high-level language for knowledge to be expressed in a transparent way and with a high expressive power.
- Having well-understood formal semantics, which assign an unambiguous meaning to logical statements.
- Disposing of a precise notion of logical consequence, which determines whether a statement follows semantically from a set of other statements (premises).
- There exist proof systems that can automatically derive statements syntactically from a set of premises.
- The existance of proof systems for which semantic logical consequence coincides with syntactic derivation within the proof system. Proof systems should be sound (all derived statements follow semantically from the premises) and complete (all logical consequences of the premises can be derived in the proof system).
- Predicate logic is unique in the sense that sound and complete proof systems do exist. More expressive logics (higher-order logics) do not have such proof systems.
- Because of the existence of proof systems, it is possible to trace the proof that leads to a logical consequence. In this sense, the logic can provide explanations for answers.

Languages as RDF and OWL are specialisations of predicate logic. These specialized languages have the advantage of providing a syntax prepared for the intended usage, i.e. for Web languages strongly working with tags. Furthermore, they define reasonable subsets of logic also taking the use cases into account. This is particularly advantageous as there is a trade-off between the expressive power and the computational complexity of certain logics, with more expressivity in the language meaning less efficiency in the proof systems. As an example, OWL Lite and OWL DL correspond to a description logic, i.e. a subset of predicate logic for which efficient proof systems exists.
Another subset of predicate logic disposing of efficient proof systems comprises the rule systems (a.k.a. Horn logic). Horn logic is orthogonal to description logics, i.e. neither of them being a subset of the other. Therefore it is impossible to state that persons being born and

[45] Inference is the act or process of deriving a conclusion based solely on what one already knows. Cp. http://en.wikipedia.org/wiki/Inference, 20.08.2006.
[46] Deduction can refer to deductive reasoning (i.e. inference in which the conclusion is of no greater generality than the premises) or natural deduction (an approach to proof theory that attempts to provide a formal model of logical reasoning as it "naturally" occurs). Cp. http://en.wikipedia.org/wiki/Deduction, 20.08.2006.

living in the same town are “home-grown persons” in OWL, whereas this is no problem using rules:

```
born(X,Y),lives(X,Z),loc(Z,U) → homegrownPerson(X)
```

Rules, on the other hand, are not able to state that a person is either a man or a woman. In OWL this is easily done using disjoint union.

A simple application example utilising the Logic layer’s capabilities is linking two databases on the Web. Assuming two databases, originally designed individually and independently, the Logic layer can help to link both by introducing semantic links (by means of RDF). As a consequence queries on one database can be converted into queries on the other database. Figure 23 gives an example of two databases, one describing friends and the other describing places, that can be linked as shown below.

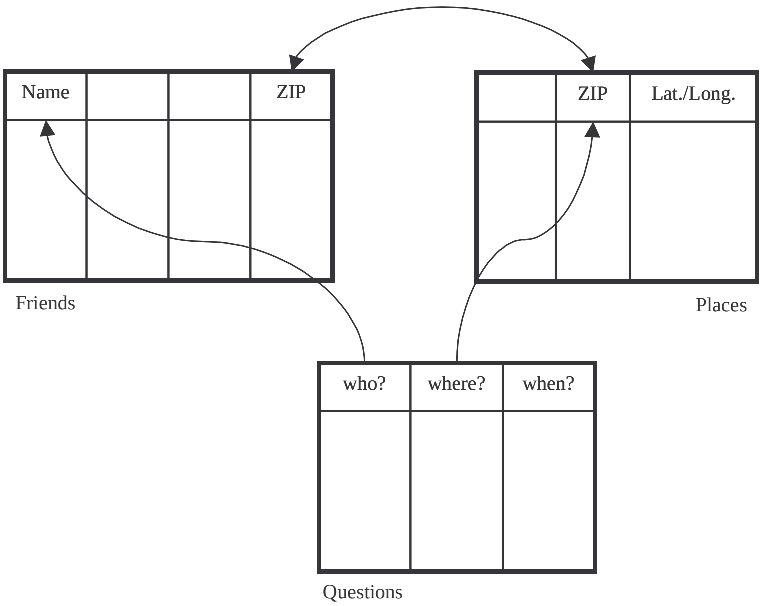

Figure 23: Converting queries on the Logic layer[47]

Computer programs trying to derive answers from a knowledge base are called inference engines or reasoners. Such programs form the “brain” used by expert systems to reason about the information in the knowledge base with the ultimate aim to formulate new conclusions. In general, three types of inference are differentiated:

- Deduction, finding the effect with the cause and the rule.
- Abduction, finding the cause with the rule and the effect.
- Induction, finding the rule with the cause and the effect.

A reasoner can derive new formally annotated facts from a set of predefined formally annotated facts. Starting with information and rules from an ontology a reasoner is able to draw further conclusions.

[47] Cp. http://www.w3.org/DesignIssues/Semantic.html, 07.09.2006.

3.6.1 Description Logic

Description logics (DL) are a family of knowledge representation languages. They are used to represent the terminological knowledge of an application-specific domain in a structured and formally well-understood way. These languages represent subsets of first-order logic, which are expressive enough and also decidable regarding inference mechanisms. A specific feature of DL is that classes (concepts) can be described by properties that must be fulfilled for an object to belong to this class [70].

The name description logic refers, on the one hand, to concept descriptions used to describe a domain and, on the other hand, to the logic-based semantics which can be given by a translation into first-order predicate logic (see above). DL was designed as an extension to frames and semantic networks, which were not equipped with a formal logic-based semantics. It has become an important cornerstone of the Semantic Web for its use in the design of ontologies with OWL being based on description logic.[48]
With DL it becomes possible to deduce new knowledge from implicit knowledge, i.e. it is no longer necessary to store all the knowledge available in databases. An example is to store in a database the information that a person A is parent of a person B. With that sort of knowledge available it is possible to deduce that A is an ancestor of B with out explicitly stating it within the database, for instance.

3.7 Layer 6: Proof

The Semantic Web's proof layer is used to proof some statement or some conclusion drawn previously. This is done by applying known rules or concluding from known rules. However, automatic proofing is nothing trivial and some not necessarily knows whether a proof is possible in a specific case. Heuristic engines are computer programs searching the Semantic Web for rules and ontologies until a statement is found to be true, or false.

Nevertheless, assuming a system of logic implemented, it will become increasingly possibly to prove things. So different people can write logic statements and a machine can be used to follow these "semantic links" and try beginning to prove facts.

This will lead to a Semantic Web with different functionalities: some sites will merely provide data, others will search for and compare data to build rules, others will follow this rules and statements to draw conclusions and place them on the Web again.

One example could be a client submitting a proof to a server in order to verify its access rights to some information requested. The server then will have to validate the client's proof. After that, the server will answer with another proof returning the information requested initially. Figure 24 shows such an example where an intelligent agent could proof whether some person has access rights to the University's online library. Such an agent could validate that Kurt Reichinger is Ph.D. student at the DBAI Group; and that DBAI Group is an institute of the Vienna University of Technology. It should be noted that both information needed (i.e. the names of active students as well as the University structure with related institutes) has to be available to the agent. This could be hard-coded within the agent's software, stored on some corporate/private server(s) or being freely available on the Web. As the agent further knows about the rule of University institutes' Ph.D. students having access to the online library, Kurt Reichinger can be granted access accordingly.

[48] Cp. http://en.wikipedia.org/wiki/Description_logic, and http://www.inf.unibz.it/~franconi/dl/course/, 15.10.2006.

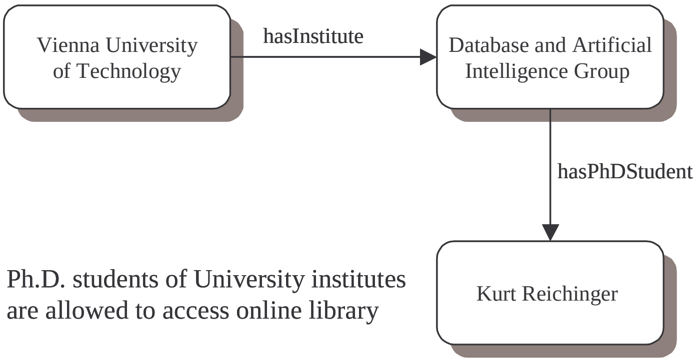

Figure 24: Example for constructing proofs

Unfortunately, the example from Figure 24 has some back draws, as long as the information provided by the client (i.e. a person pretending to be Kurt Reichinger being a Ph.D. student at DBAI Group), the information from other sources (i.e. DBAI Group being an institute at Vienna University of Technology), and the rule itself (i.e. Ph.D. students of University institutes having access to online library) are not trustworthy. Section 3.8 deals exactly with that issue.

3.7.1 Rule Markup Language

The Rule Markup Language (RuleML) is providing an XML namespace and allows for the exchange of rules. RuleML permits both forward (bottom-up) and backward (top-down) rules in XML for deduction, rewriting, and further inferential-transformational tasks as stated on the RuleML website.[49] RuleML is a family of sublanguages whose root allows access to the language as a whole and whose members allow to identify customized subsets of the language. Examples of such RuleML sublanguages are hornlog or datalog.[50]

3.7.2 Datalog

Datalog is a declarative (programming) language. This means that the programmer does not write a program that solves some problem but instead specifies what the solution should look like, and a Datalog inference engine (or deductive database system) tries to find the way to solve the problem and the solution itself. This is done with rules and facts. Facts are the input data, and rules can be used to derive more facts, and in the best case, the solution of the given problem [15].[51]

A successful example implementation of Datalog is the Lixto visual wrapper [6], unfolding the structure of some desired pieces of information. The wrapper extracts information from a (usually not well-structured) format and maps it to a structured format. In the case of Web extraction as performed by the mentioned visual wrapper, the mapping is usually carried out from HTML to XML or a relational database. [52]

Another example is the DLV system [38] for disjunctive datalog with constraints, true negation and queries. Disjunctive datalog is an extension of datalog in which the logical OR expression (the disjunction) is allowed to appear in the rules which is not the case in basic

[49] Cp. http://www.ruleml.org/, 16.10.2006.
[50] Cp. http://www.ruleml.org/modularization/#Model, 16.10.2006.
[51] Cp. http://www.dbai.tuwien.ac.at/proj/dlv/tutorial/, 15.10.2006.
[52] Cp. http://www.lixto.com/show.php?page=vw_overview&lg=en, 15.10.2006.

datalog. DLV is a deductive database system well suited for all kinds of nonmonotonic reasoning, including diagnosis and planning.

3.7.3 F-Logic

F-logic (frame logic) [37] is another formal language for data and knowledge representation. It accounts in a declarative fashion for structural aspects of object-oriented and frame-based languages. First, F-logic was used for deductive and object-oriented databases; later it was adapted for implementing ontologies. Features include, among others, object identity, complex objects, inheritance, polymorphism, query methods and encapsulation.[53]

3.7.4 Semantic Web Rules Language

The Semantic Web Rules Language (SWRL) [82] is combining sublanguages of the OWL Web Ontology Language (OWL DL and Lite; see Section 3.5.4) with those of the Rule Markup Language (Unary/Binary Datalog; see 3.7.1 and 3.7.2).[54]

The SWRL foundations are described by the W3C as follows: A rule axiom consists of an antecedent (body) and a consequent (head), each consisting of a (possibly empty) set of atoms. A rule axiom can also be assigned a URI reference, which could serve to identify the rule. Informally, a rule may be read as meaning that if the antecedent holds (is "true"), then the consequent must also hold. An empty antecedent is treated as trivially holding (true), and an empty consequent is treated as trivially not holding (false). Rules with an empty antecedent can thus be used to provide unconditional facts; however such unconditional facts are better stated in OWL itself, i.e., without the use of the rule construct. Non-empty antecedents and consequents hold if all of their constituent atoms hold, i.e., they are treated as conjunctions of their atoms.

SWRL significantly extends the expressivity of OWL, allowing expressing relations like an uncle is the brother of a father (see Listing 10), that an adult is a person with an age above 18 years, or that an international flight needs airports in different countries.

Listing 10: SWRL code fragment for uncle – brother – father relation[55]

```
<ruleml:imp>
  <ruleml:_rlab ruleml:href="#example1"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasParent">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x2</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom swrlx:property="hasBrother">
      <ruleml:var>x2</ruleml:var>
      <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="hasUncle">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
```

[53] Cp. http://de.wikipedia.org/wiki/F-Logic, and http://en.wikipedia.org/wiki/F-Logic, 15.10.2006.
[54] Cp. http://www.w3.org/Submission/SWRL/, 23.09.2006.
[55] Cp. http://www.inf.fu-berlin.de/inst/ag-nbi/lehre/05/S_MOD/Regelsprachen_180505.pdf#search=%22swrl%22, 24.09.2006.

```
  </ruleml:_head>
</ruleml:imp>
```

The code fragment from Listing 10 is divided into a RuleML body and a RuleML head with the head of a rule fulfilled (i.e. “hasUncle”) when all atoms of the body are fulfilled (i.e. “hasParent” and “hasBrother”). The SWRL ontology is a collection of facts and axioms, using e.g. subClass, and equivalentClass. It further uses a number of built-in’s for various purposes as listed in [7]:

- Comparisons Built-Ins (e.g. equal, notEqual, lessThan, or greaterThan)
- Math Built-Ins (e.g. add, subtract, multiply, divide, mod, round, sin, or cos)
- Boolean Values Built-Ins (e.g. boolean, or booleanNot)
- Strings Built-Ins (e.g. stringLength, upperCase, or lowerCase)
- Date, Time Built-Ins (e.g. date, time, subtractDate, or subtractTime)
- URIs Built-Ins (e.g. resolveURI, or anyURI)
- Lists Built-Ins (e.g. member, length, first, or list)

3.8 Layer 7: Trust

The Proof layer (layer 6) described in section 3.7 is strongly inter-related with layer 7, the Semantic Web’s Trust layer.

Trust is a complex subject relating to belief in the honesty, truthfulness, competence, reliability, etc. of a person or service [30]. Trust typically is specified in terms of a relationship between a *trustor*, the subject that trusts a target entity, and a *trustee*, i.e. the entity that is trusted. In general, a trust relationship is not absolute: a trustor trusts a trustee with respect to its ability to perform a specific action or provide a specific service within a *context*. A trust relationship can be one-to-one between two entities; however it has not to be symmetric. It can also be one-to-many or many-to-many.

As “anybody can say anything about anything” on the Web, it becomes a crucial question which information to trust, and which not. It is evident that not everything found on the World Wide Web is true; and the Semantic Web is not able to change that in any way. Truth of information as a general rule has to be evaluated on a case-by-case approach by each application that processes information on the Web. According to Deutsch [20], trust is the readiness of a trustor to rely on the actions or information of a trustee. Trust in this context is related to the following aspects:

- Trust emerges from a social process
- Trust is based on a subjective decision of the trustor
- Trust can change over time
- Trust depends on various factors and situations

Applications working with trust measures have to decide what they trust by using the context of the statements provided, e.g. who said what and when and what related credentials are made available. Two questions arise in this context:

1. Is the information from the person that claims to be the author?
2. Is the author trustworthy?

The first question regarding authenticity of information can be addressed with one of the following approaches:

- Appending of digital signatures to the underlying RDF models, i.e. the according RDF model or ontology is signed by the author using public key infrastructure (PKI) schemes that is addressed in section 3.8.1.
- Alternatively someone could simply consider the source of an RDF model on the Web trustworthy, because the data is hosted at some renowned University's web server. Of course, this second approach provides a lower security level with the advantage of less effort necessary for the information provider.

The second question regarding the trustworthiness of an author can be addressed using different approaches as outlined by Bizer [12]:

- Web of Trust with explicit trust statements: is an approach being based on a strong network with the trustor defining his/her trustees, e.g. a person has to explicitly name the trusted persons. Only statements from the trustor's trustees are used for trust calculations. The Web of Trust relationships can be visualised with a directed, weighted graph. The success of this approach depends on actuality and quality of the statements, leading to a high degree of maintenance and administration. See section 3.8.3 for further details regarding the Web of Trust.
- Network analysis with explicit trust statements: is an approach where people not being part of a network use an existing Web of Trust. Examples for this approach are the various reputation services on electronic market platforms, e.g. Ebay[56] or Amazon[57], where users can post statements on products or on each other. Another example is specific platforms for opinion exchange, e.g. Tripadvisor[58], Dojoo[59] or Ciao[60]. Figure 25 shows a screenshot snippet from the Tripadvisor Web site offering a mixture of hard facts, e.g. address or official hotel class, and soft facts, e.g. traveller reviews and overall Tripadvisor ratings.

Figure 25: Screenshot snippet from Tripadvisor Web site, 27.09.2006

- Network analysis with implicit trust statements: is an approach using existing RDF statements. An example of such a statement could be "Trust all statements regarding subject X of University professors heading a department dealing with subject X".

[56] Cp. http://www.ebay.com, 26.09.2006.
[57] Cp. http://www.amazon.com, 26.09.2006.
[58] Cp. http://www.tripadvisor.com, 26.09.2006.
[59] Cp. http://www.dooyoo.com, 26.09.2006.
[60] Cp. http://www.ciao.com, 26.09.2006.

- Pragmatic approach: is checking available information against known ontologies or knowledge bases, i.e. verify whether some information is in line with some other known facts. An example of this is checking an ontology for e.g. cardinalities, value ranges or other axioms, e.g. “a girl of 3 years of age cannot be a mother”.

Assuming the intelligent agent from the example presented in section 3.7 concludes that all rules are holding (i.e. all rules are fulfilled) it is about how much the agent is willing to trust the information gathered. Depending on some trust information or trust level provided by the information sources, it is the agent’s task to decide whether some information is trustworthy, or not.

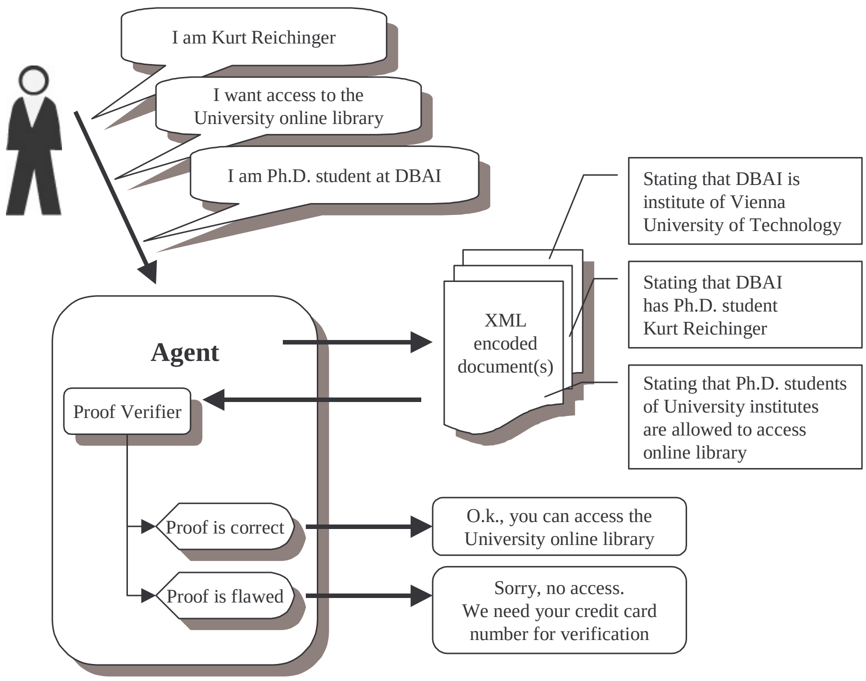

Figure 26: Example agent working with proof and trust layer

Figure 26 extends the example from section 3.7 by introducing an agent, deciding whether some person gets access to the University online library, or not. The agent in the example works with three input parameters, i.e. that a person named Kurt Reichinger is Ph.D. student at DBAI and wants to access the University online library. These facts are proofed by the agent’s proof verifier against XML documents provided by authorised entities, e.g. the University library providing the rule for access or the University administration providing the list of active students and the university institute structure. If the agent’s proof verifier comes to the conclusion, that all data provided is correct and all rules are holding, the person is granted access to the online library. If the proof verifier cannot be convinced that all data provided is correct, access is denied and further actions may be requested from the student. In Figure 26 the application asks for identification using credit card verification, in the case of a proof being deemed flawed.

Two measures are used for providing trust information on the Semantic Web’s layer 7. The one is digital signatures, and the other is encryption. The following sections highlight on this two important building blocks.

3.8.1 Digital Signatures

The first building block is digital signatures, which generally provide cryptographically based proof that a certain person wrote (or agrees with) a document or statement. This makes it possible to explicitly tell an agent which sources are classified trustworthy, and which are not.

Often, digital signatures are used with public key infrastructure (PKI) schemes, where a digital identity certificate (issued by a certificate authority) is tied to a user. Such PKI systems typically use asymmetric key cryptography to bind user information like name and address to a public key, a process comparable with that of a notary endorsement. Most digital signature schemes work with two complementary algorithms, one for signing and one for checking this signature at a later time. The output of the signature process is called a digital signature.

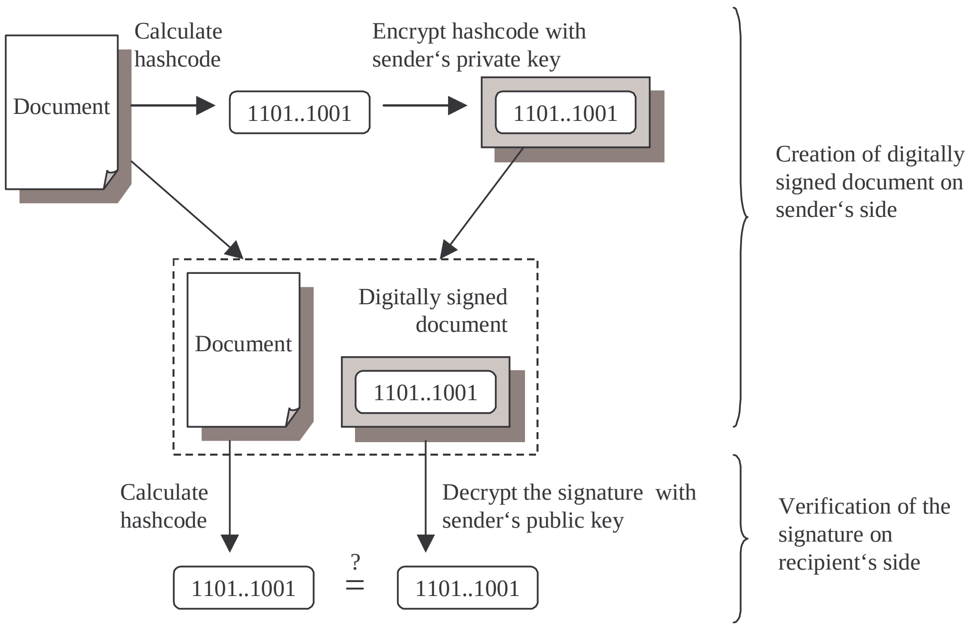

Figure 27: General principle of using digital signature

The reasons for applying a digital signature to communications are authentication, integrity and non-repudiation.

- Authentication: PKI systems allow the encryption of a message (a document) with a user’s private key, where it is not necessary to send the message itself in cipher text. Instead, a hash of the message is generated and then protected with encryption. Because it is not possible to alter the message (the document) without changing the hash, this method helps to introduce authentication. After receiving an encrypted message, the hash is decrypted using the sender’s (known) public key. Then the result is checked against a newly generated hash of the alleged plaintext. If the hashes match, the message recipient can feel quite confident that the original message has been encrypted with the sender’s private key and therefore comes from the sender, who should be the only one being able to sign messages with that specific private key.
- Integrity: Sender and recipient usually have an interest that a message is not altered during transmission. Encryption may make it harder for a third party to read the message, but it may still be possible to alter the message, without actually reading it. This may have malicious effects to sending and/or receiving party.

- Non-repudiation: In a PKI context, repudiation is about disclaiming responsibility for a message, i.e. claiming that a message was not sent by the person meant to be the sender. A recipient may call on a sender to attach a signature to a message, in order to have a claim to its origin later on.

3.8.2 Encryption

The second building block of the Semantic Web's trust layer is encryption. In cryptography, encryption is about obscuring information in order to make it unreadable for unauthorised readers without special knowledge. While encryption in the past was used mainly for communication with a significant need for secrecy (e.g. governments, embassies, military or secret services), it has emerged into the public domain in the last decades, being used e.g. in Internet applications, bank transfers or telephone networks, nowadays.
While encryption can be used to ensure secrecy, other techniques are still needed to make communications secure. See section 3.8.1 for issues regarding authenticity and integrity of messages for example.

With digital signatures and encryption introduced, a computer – in theory – can evaluate all factors found and make a decision how trustworthy a piece of information is. This could be a simple thumb up / thumb down information or a more complex statement on the various trust factors involved. However, with millions of authors and billions of documents on the Web, this will not be a very successful (and scalable) thing to do. That is where the "Web of Trust" steps in.

3.8.3 Web of Trust

As briefly discussed above, the Web of Trust is a network of trust measures taking into account to what extent members of the network trust each other. Each member individually has to set trust parameters for other members of the network, therefore explicitly defining which information sources are deemed trustworthy. Only information from these sources will be trusted in future transactions or communications. Such a network can be visualised using a directed, weighted graph, with an arrow indicating a trust relationship between two nodes (persons). The direction of the arrow indicates that node A is trusting node B, but not necessarily vice versa. In addition, a weight can be attached to an arrow indicating the extent some node is trusted.

Using a trust ontology as suggested and introduced by Golbeck [27] an according RDF statement reads as shown below (see Listing 11). Working with RDF trust statements ranging from "trustsAbsolutely" to "distrustsAbsolutely", the code fragment simply says that a person named John highly trusts ("trustsHighly") a person named Sally.

Listing 11: RDF Code fragment setting (weighted) trust level

```
<Person rdf:ID="John">
  <mbox rdf:resource="mailto:john@example.com"/>
  <trustsHighly rdf:resource="#Sally"/>
</Person>
```

Figure 28 shows the statement from Listing 11 as a directed, weighted graph, with the arrow's direction denoting that John trusts Sally, and the trust weight being "trustsHighly".

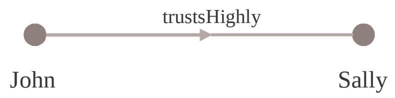

Figure 28: Weighted graph visualizing Listing 11

Such a network of weighted trust can be used to calculate trust levels for members of the network not known to the querying party. Imagine John having set a trust relation to Sally only (“trustsHighly”), but Sally having a trust relationship (e.g. “trustsAbsolutely”) with a third person named Carol. Now, John could say, if he highly trusts Sally and Sally absolutely trusts Carol, it should be possible for him to trust Carol as well (e.g. “trustsModerately”).

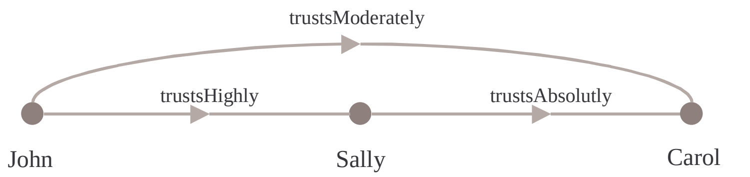

Figure 29: Weighted graph visualising the relationships between John, Sally and Carol

In this way, a third party trust relationship is created between two members of the network not personally known to each other and not having set mutual trust relationships for each other. Based on these considerations, Golbeck further introduced a reputation network analysis for email filtering [28].

This works fine for rather small, closed communities, but becomes increasingly difficult to manage when the number of members increases or new (unknown) users want to join. Furthermore, trust differs depending on the specific situation or topic, making it increasingly hard to derive conclusions from a Web of Trust.

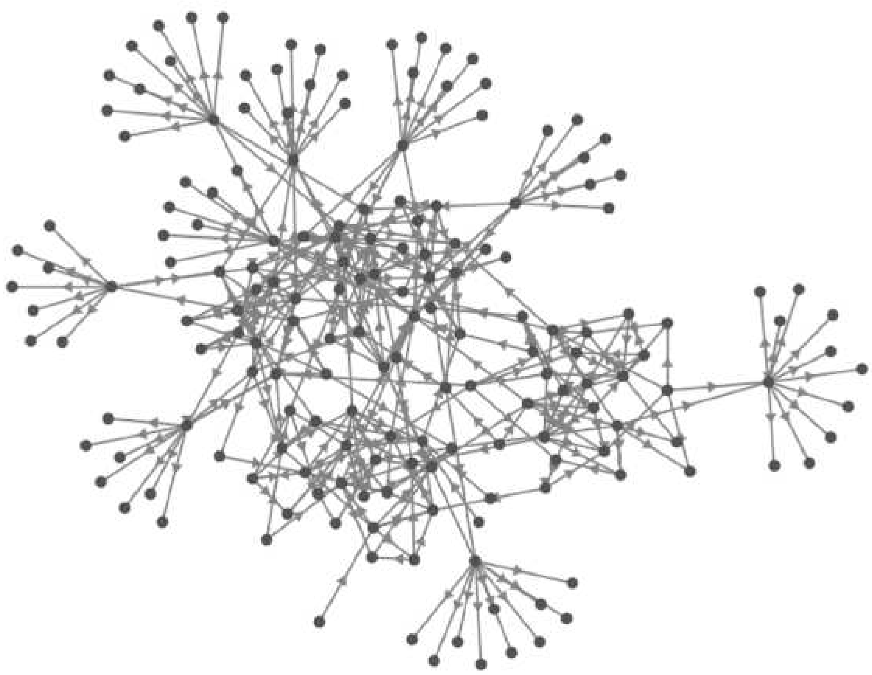

Figure 30: Directed graph visualizing a Web of Trust example (taken from [27])

With the community getting larger and more and more unknown people joining the network, new approaches have to be found. One possibility is to extend trust to unknown nodes, i.e. using the trust level set by others not necessarily known to each other.[61] This network analysis approach with explicit trust statements implies a principal trust in statements given by other unknown people, but works quite well in platforms like Ebay (where buyers are rating sellers, and vice versa) or Amazon (where members rate products). It is left to the individual user whether some rating (for a seller, a product, a hotel or anything else) is deemed trustworthy, and to what extent.

Talking RDF, the Web of Trust (WOT) RDF ontology[62] supports the cause of digital signatures and PKI. The WOT schema is designed to facilitate the use of PKI tools (e.g. PGP[63] or GPG[64]) to sign documents and to document these signatures.

[61] This is trust based on e.g. previous statements of some person or affiliation to some group or community.
[62] Cp. http://xmlns.com/wot/0.1/, 10.10.2006.
[63] Pretty Good Privacy is a cryptography system developed by Phil Zimmermann in 1991.
[64] Gnu Privacy Guard is a free cryptography system implementing the OpenPGP standard as defined in RFC 2440 [14].

4 Web 2.0

In recent months, social network applications, also known as Web 2.0, have become an ever growing phenomenon both in terms of available applications and usage figures. Web 2.0 applications are perfectly suitable to provide background information on people, as examples like del.icio.us, flickr, 43things, writely, plazes or CalendarHub show. Although the technologies and services that comprise Web 2.0 (generally) do not understand and extract meaning, as is the Semantic Web proponent’s vision, it is an important step in that direction. Related research mainly deals with social network issues, e.g. relations between users or between tagged items. Work in this area can be categorised in metadata acquisition, storing and enhancing metadata through reasoning, and browsing and visualising results [49].

4.1 Introduction

Web 2.0 is a term originating from a conference brainstorming session between O'Reilly and MediaLive International that led to the first Web 2.0 conference [55]. The term never was exactly defined since but is used today representing an open concept for new applications on the Web.

From a strategic point of view, Web 2.0 is positioning the Web as a platform. Often, Web 2.0 applications use public Web Service application programming interfaces (API), Asynchronous Javascript and XML (AJAX) and are easily accessible with standard Web browsers. In many cases, Web 2.0 applications use lightweight, easy-to-use mechanisms for ontology and metadata creation allowing for mass publishing and social networking between users, so-called folksonomies (from folk and taxonomy, i.e. people’s classification management). In contrast to controlled vocabularies (also known as taxonomies), folksonomies are less systematic and sophisticated. That lower complexity however, makes it easy for users to create their own tags and link them with each other [50]. From a user’s point of view, Web 2.0 gives control of data to the user. It is left to the user what data to publish and to whom data is made available. Therefore access to Web 2.0 data is user-configurable from strictly private to fully public available. Further corner stones of Web 2.0 are the concept of offering services instead of software packages, the philosophy of building architectures of participation involving the customers and their knowledge harnessing collective intelligence.

In terms of applications, Web 2.0 meanwhile comprises a multitude of applications available on the Web with the following being popular examples: flickr for photo sharing and tagging[65], del.icio.us for link sharing and tagging[66], 43things for creating to-do lists[67], writely for

[65] Cp. http://www.flickr.com, 20.08.2006.
[66] Cp. http://del.icio.us, 20.08.2006.
[67] Cp. http://www.43things.com, 20.08.2006.

desktop publishing[68], plazes for geographic tagging[69] or CalendarHub for sharing and tagging of calendar entries[70].

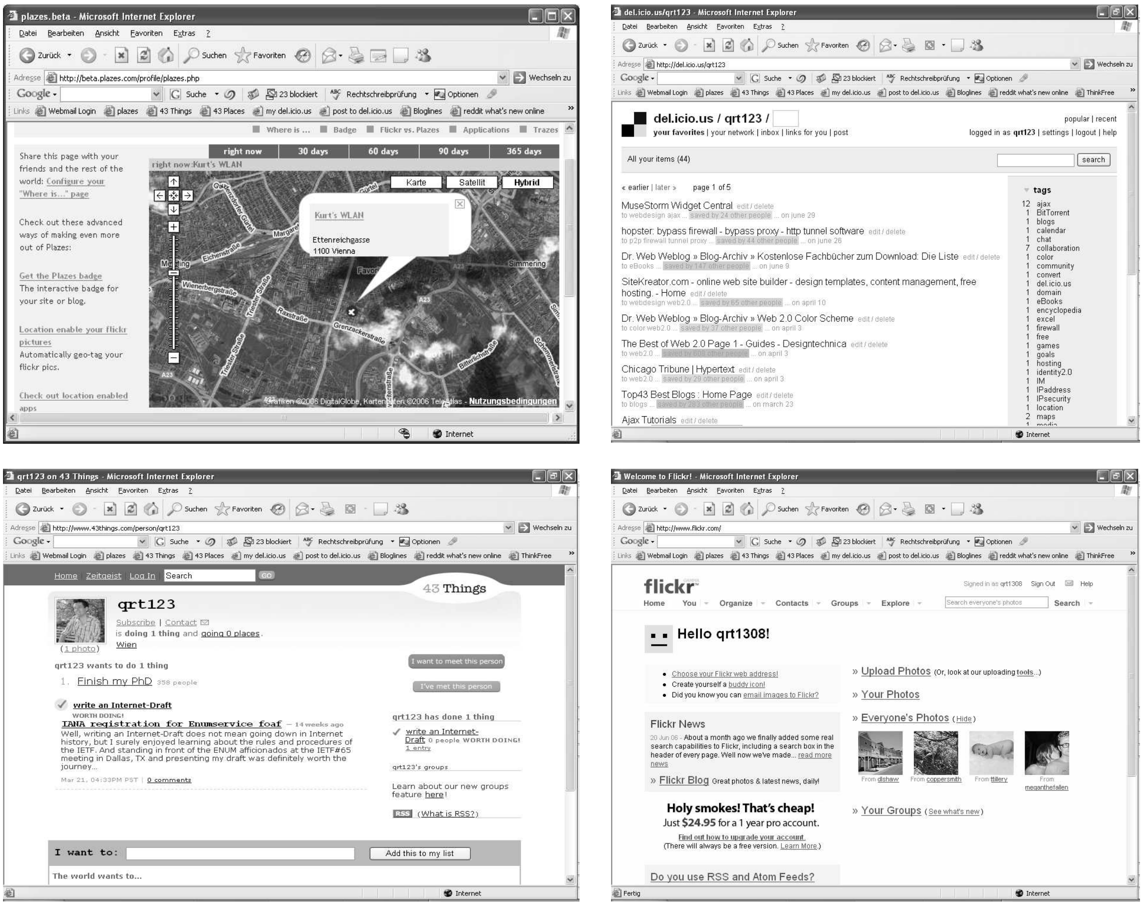

Figure 31: Screenshots from Web 2.0 websites plazes, del.icio.us, flickr and 43things (clockwise from top left) all showing the author's accounts on these sites

The above mentioned Web 2.0 examples have in common that they tend to build Web-based communities, which significantly change the face of the Web. While the former Web 1.0 could have been characterised as primarily being a source of information, Web 2.0 calls for the active participation of human users. This is fostered by the services' syndication (see below) and messaging capabilities.

Another group of Web 2.0 applications can be described as Web-based applications. Those include word processing, spreadsheet and slide-show presentation offering simple online clones to the significantly more sophisticated Microsoft™ offline products Word, Excel and Powerpoint. Typically, these Web 2.0 applications work with WYSIWYG concepts, often being supported by Ajax and Java technology.

Furthermore, so-called Rich Internet applications have entered the stage improving the users' experience with techniques like Ajax, Adobe Flash or Flex, that allow Web pages to partly update content without refreshing the whole page.

68 Cp. http://www.writely.com, 20.08.2006.
69 Cp. http://www.plazes.com, 20.08.2006.
70 Cp. http://www.calendarhub.com, 20.08.2006.

One of the most important contributions to Web 2.0 is the syndication of Web content already mentioned above. Syndication means, that end users now are enabled to dynamically use content from other Web sites in a different context. Protocols involved include Really Simple Syndication (RSS)[71], Resource Description Framework (RDF) and Atom[72]. Those are supported by Web communication protocols as REST[73] and SOAP[74]. In many cases standard Web service APIs are used often involving XML.

Tim O'Reilly suggested on O'Reilly radar Web site[75] differentiating related applications according to the following categories:

- Level-3 applications: the most "Web 2.0" applications belong to this group. They can only exist on the Internet, deriving their power from the human connections and network effects, and growing in effectiveness the more people use them. Examples are eBay, Wikipedia, del.icio.us, or Skype.
- Level-2 applications: these can operate offline but gain advantages from going online. An example is flickr, benefiting from its shared photo-database and from its community-generated tag database.
- Level-1 applications: they are also available offline but which gain features online. Examples include writely (gaining group-editing capability online) and iTunes (because of its music-store portion).
- Level-0 applications: these work offline as well. Examples are MapQuest, or Google Maps.
- Non-Web applications: this group comprises applications like email, instant-messaging clients or the telephone.

4.2 Web 2.0 vs. Semantic Web

It should be noted that Web 2.0 must not be equated with the Semantic Web, as the latter is defined as a concept in which machine-processable content builds the base for a whole set of new applications on the Internet. For Semantic Web applications to work and communicate with each other it is necessary to create a system of structured semantic knowledge and languages for ontologies and rule-systems. Moreover, shared ontology vocabularies have to be agreed on. The evolvement of metadata languages such as XML, RDF and others allow a web-wide realisation of ontologies, rules, proofs and logic. In fact, some sources use the term "Web 3.0" for the semantic Web.

The Semantic Web is about machine-machine interaction while Web 2.0 is still more about human-machine and human-human interaction. Web 2.0 applications in most cases do not explicitly deal with the meaning of statements, but they strongly work with "tagging". A tag is a keyword or term used to classify content usually individually chosen on an informal and

---

[71] RSS is formerly known as Rich Site Summary.
[72] Atom applies to the Atom Syndication Format which is an XML language used for web feeds; and to the Atom Publishing Protocol (APP) which is a simple HTTP-based protocol for creating and updating Web resources. Generally, Web feeds allow software programs to check for updates published on a web site, e.g. headlines, full-text articles, excerpts, summaries, or links to content on a web site.
[73] REST (Representational State Transfer) allows accessing and manipulating data on a server using the HTTP verbs GET, POST, PUT, and DELETE. The popular weblogs universe is mostly REST-based involving downloading XML files (in RSS, or Atom format) that contain lists of links to other resources.
[74] SOAP allows to POST XML messages and requests to a server that may contain complex, pre-defined, instructions for the server.
[75] Cp. http://radar.oreilly.com/archives/2006/07/levels_of_the_game.html, 17.10.2006.

personal basis by the author or creator of the item. Contrary to the Semantic Web, such a Web 2.0 tag is not part of a formally defined classification scheme. This can lead to various tags describing the same item or various items described by the same tag, making it particularly difficult for machines to read.

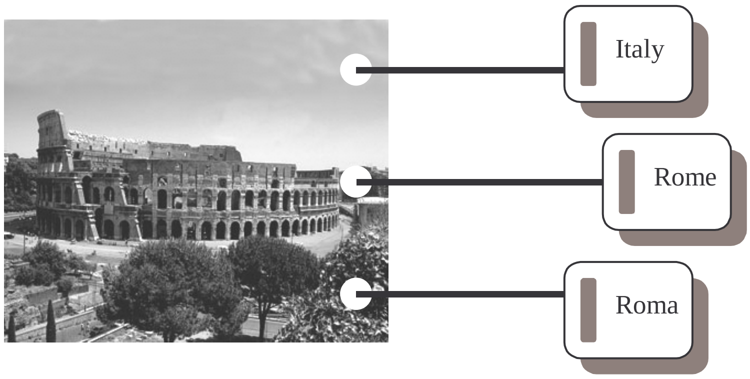

Figure 32: Various tags describing the same item

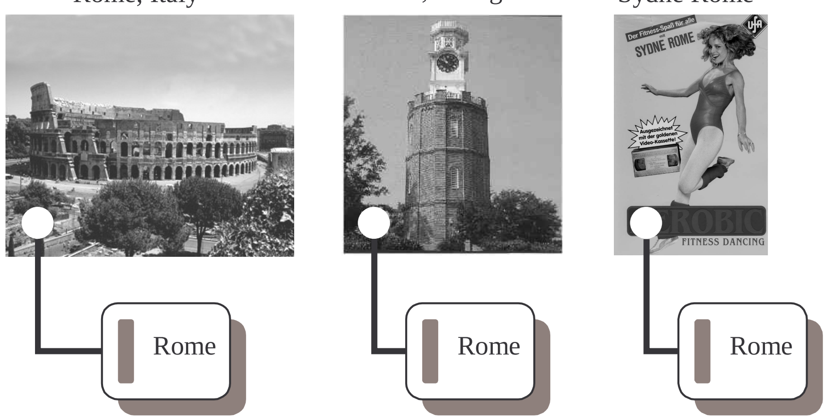

Figure 33: Various items described by the same tag

Figure 32 illustrates how a single item, i.e. a depiction of the ancient coliseum in Rome, may be described by different tags, i.e. “Italy”, “Rome”, “Roma” and many more one can imagine. Figure 33 illustrates how a single tag, i.e. “Rome” in the example above, may be used for describing different items, i.e. the coliseum in Rome, a town in the United States of America, or an actress.

Typically, tags are used in automatically generated taxonomies for online resources, e.g. files, Web pages, images and bookmarks. Web 2.0 applications typically are strongly related with tagging; providing links to items sharing the same tag, or to collections of tags. This allows browsing of content having been tagged by the users themselves.

Although this “Web of Tagged Items” is far from the functionality envisaged by Semantic Web proponents, it can be seen as an important step towards a fully-fledged Semantic Web. In fact, Web 2.0 can make a valuable contribution to the Semantic Web as shown in section 8.6.

5 Friend-of-A-Friend (FOAF)

The FOAF ("Friend of a Friend") [13] project is a frequently mentioned example of applied Semantic Web technology. This section gives a brief description of FOAF. FOAF plays an integral part regarding the PHOAF prototype, according applications and use cases presented in later sections of this thesis.

The FOAF project defines a vocabulary based on the W3C's Resource Description Framework (RDF) for expressing metadata about people and their interests, activities and relationships. FOAF is an open community-lead initiative with a clear Semantic Web vision of creating a machine-processable web of data. FOAF facilitates the creation of the Semantic Web version of a typical personal homepage: A FOAF file can contain e.g. a person's name, phone number and e-mail address, a personal website URL, home and work place address, a link to a depiction and many more. And just like the HTML counterpart, FOAF documents can be linked together to form a web of data, using well-defined semantics. That interconnected-ness gives the project its name, friend-of-a-friend.

5.1 FOAF Namespace

The FOAF vocabulary is published with its schema and specification at http://xmlns.com/foaf/0.1, called its namespace URI. The documentation includes definitions of classes and properties defined in the associated RDF schema. As FOAF is an RDF application it can easily be harvested and aggregated on the Web. FOAF is not restricted to its very own vocabulary classes and properties but can be combined with other RDF vocabularies leading to a rich set of metadata potentially available. In order to ensure compatibility amongst ontologies, FOAF classes have been related to their equivalents in other ontologies where appropriate. This allows applications built to understand these ontologies to immediately process FOAF data. At the same time it allows FOAF to integrate more complex concepts into the project.

5.2 FOAF Classes and Properties

The FOAF vocabulary can be grouped in five broad categories containing the following terms. Terms starting with upper case letters denote classes; terms with lower case letters denote properties:

- FOAF Basics: Agent, Person, name, nick, title, homepage, mbox, mbox_sha1sum, img, depiction (depicts), surname, family_name, givenname, firstName
- Personal Info: weblog, knows, interest, currentProject, pastProject, plan, based_near, workplaceHomepage, workInfoHomepage, schoolHomepage, topic_interest, publications, geekcode, myersBriggs, dnaChecksum

- Online Accounts and Instant Messaging: OnlineAccount, OnlineChatAccount, OnlineEcommerceAccount, OnlineGamingAccount, holdsAccount, accountServiceHomepage, accountName, icqChatID, msnChatID, aimChatID, jabberID, yahooChatID
- Projects and Groups: Project, Organization, Group, member, membershipClass, fundedBy, theme.
- Documents and Images: Document, Image, PersonalProfileDocument,, topic (page), primaryTopic, tipjar, sha1, made (maker), thumbnail, logo

5.3 FOAF and RDF/XML

The core component of FOAF is the Person class, simply describing a person in RDF/XML terms. The following example in Listing 12 says that there is a male person named Kurt Reichinger who has an e-mail address, a personal homepage and a Web log (blog).

Listing 12: RDF/XML code fragment

```
<foaf:Person>
   <foaf:name> Kurt Reichinger</foaf:name>
   <foaf:gender>Male</foaf:gender>
   <foaf:title>Mr</foaf:title>
   <foaf:givenname>Kurt</foaf:givenname>
   <foaf:family_name>Reichinger</foaf:family_name>
   <foaf:mbox >reiching@dbai.tuwien.ac.at</foaf:mbox >
   <foaf:homepage rdf:resource="http://www.dbai.tuwien.ac.at/proj/semnum/"/>
   <foaf:weblog rdf:resource="http://semnum.blogspot.com/ "/>
</foaf:Person>
```

While capturing some basic metadata about persons may already be useful for some applications, FOAF is mainly about persons' relations. The foaf:knows property is used to indicate some sort of relationship between two people. The example RDF/XML code fragment in Listing 13 below says that a person Kurt Reichinger has an e-mail address and knows a person Robert Baumgartner.

Listing 13: RDF/XML code fragment using the foaf:knows property

```
<foaf:Person>
   <foaf:name>Kurt Reichinger</foaf:name>
   <foaf:mbox >reiching@foo.bar</foaf:mbox >
   <foaf:knows>
       <foaf:Person>
       <foaf:name>Robert Baumgartner</foaf:name>
       </foaf:Person>
   </foaf:knows>
</foaf:Person>
```

In addition to the foaf:knows relationship, it is possible to link to another FOAF document using the rdfs:seeAlso property. That property indicates a resource that may contain additional information about its associated resource. Extending the example from Listing 13 this can be used to point to Robert Baumgartner's own FOAF description, as shown in Listing 14.

Listing 14: RDF/XML code fragment using the rdfs:seeAlso property

```
<foaf:Person>
   <foaf:name>Kurt Reichinger</foaf:name>
```

```
   <foaf:knows>
       <foaf:Person>
       <foaf:name>Robert Baumgartner</foaf:name>
       <rdfs:seeAlso>http://www.dbai.tuwien.ac.at/staff/baumgart/foaf.rdf
</rdfs:seeAlso>
       </foaf:Person>
   </foaf:knows>
</foaf:Person>
```

Both foaf:knows and rdfs:seeAlso can be used to build a web of machine-processable metadata making it possible for applications to spider (or "scutter" in FOAF terminology) these RDF hyperlinks and to build a database of FOAF data.

5.4 Publication of FOAF File

In order to make an RDF FOAF document publicly available it has to be published on the Web and linked to existing FOAF data. There are several ways to do this: First, using the terms foaf:knows and rdfs:seeAlso as described above; Second, using the FOAF Bulletin Board, And third, using auto-discovery linking to a FOAF document from an HTML page using the link element.

- Using foaf:knows and rdfs:seeAlso within a FOAF file allows linking to other persons and FOAF files, respectively. However, the creator of some FOAF file has to rely on others to get that FOAF file linked (and therefore become visible to FOAF scutters), as rdfs:seeAlso is designed for one-way linking, only.

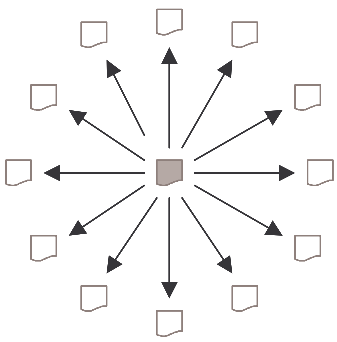

Figure 34: FOAF file one-way linking to other FOAF files using rdfs:seeAlso

- Using the FOAF Bulletin Board[76], i.e. a FOAF file directory set up by FOAF community members and linking to dozens of other FOAF files.

[76] Cp. http://rdfweb.org/topic/FOAFBulletinBoard, 20.08.2006.

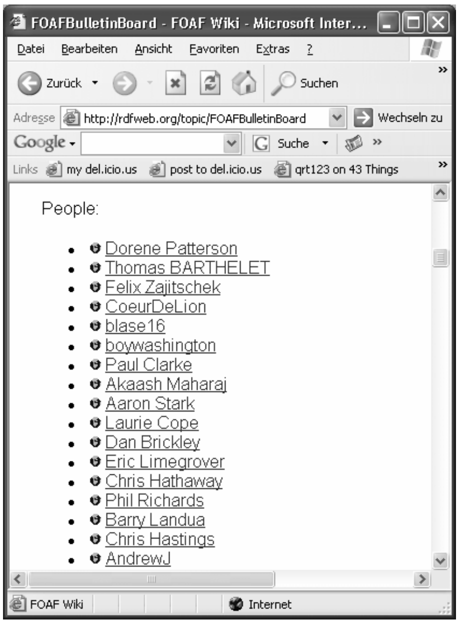

Figure 35: Screenshot from FOAF Bulletin Board, from 22.07.2006, 15:54

- Using auto-discovery linking to a FOAF document from an HTML page using the link element. An appropriate code fragment is shown in Listing 15, where "foaf.rdf" is a relative link to the actual FOAF file, with the name having to be changed appropriately.

Listing 15: HTML Code Fragment for FOAF auto-discovery

```
<link rel="meta" type="application/rdf+xml" title="FOAF" href="foaf.rdf" />
```

Once a FOAF file is published on the Web, it can be queried by appropriate tools to have the information contained presented to the user or another machine (agent). A typical result of such a query performed with the FOAF Explorer from the FOAF tools page[77] for presentation on a Web site is shown below (see Figure 36).

[77] Cp. http://xml.mfd-consult.dk/foaf/, 20.08.2006

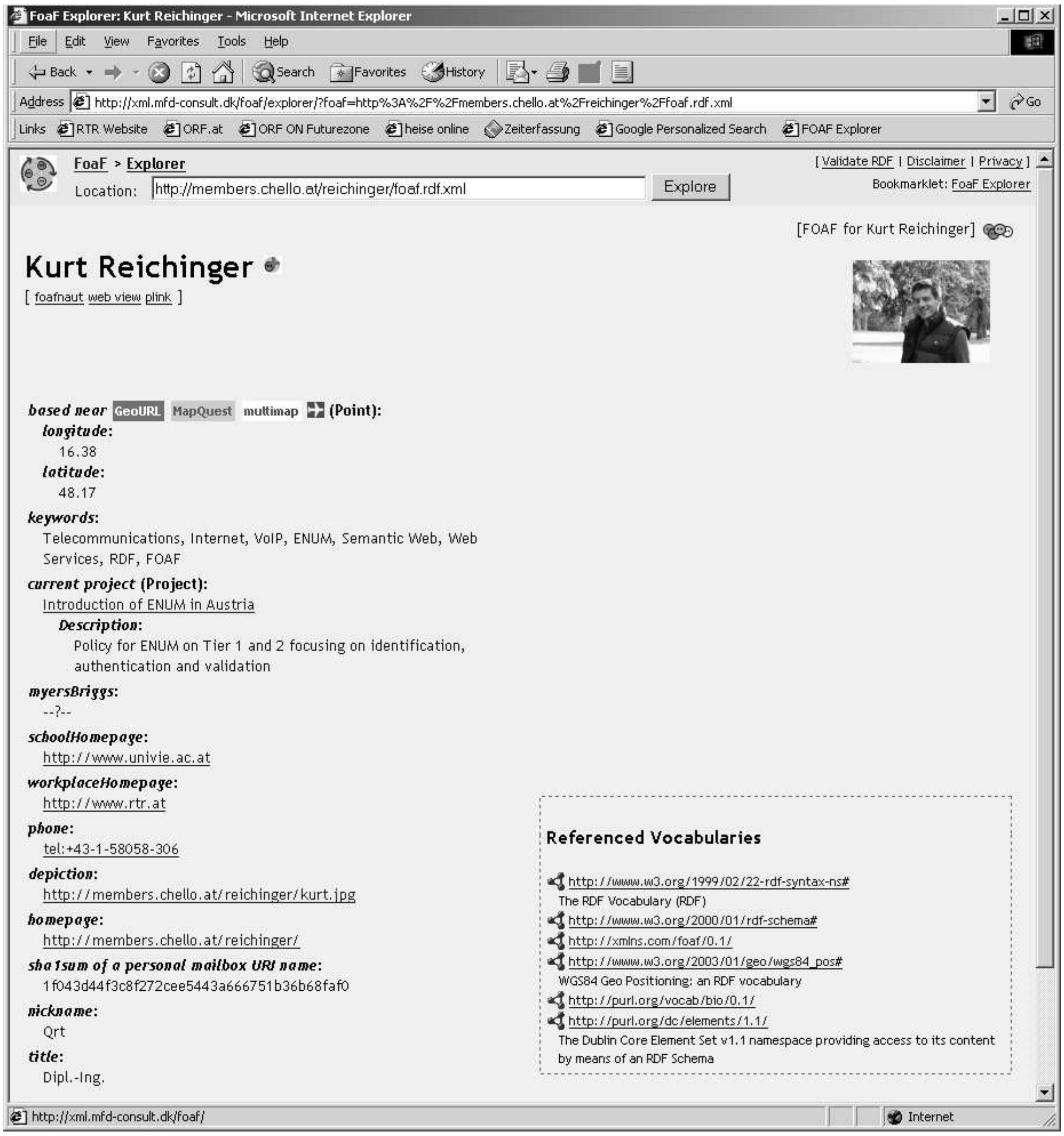

Figure 36: Result of FOAF/RDF query with FOAF Explorer

The FOAF Explorer Web site shows the content of some person's FOAF file. In the example from Figure 36 the author's FOAF data is presented starting with the full name, a depiction, location data based on longitude and latitude values, keywords like Semantic Web and ENUM, information on current projects, and so on. Furthermore the referenced vocabularies are listed in the bottom right hand corner of the screenshot, in this case RDF, RDFS, FOAF, WGS84, BIO [18] and DC.

A tool developed during the cause of this thesis will be presented in section 7 with the PHOAF prototype. PHOAF builds the technological basis for application examples presented later in this thesis.

6 Introducing ENUM to the Semantic Web

The Semantic Web as introduced in section 3 generally can be regarded as an evolutionary development of today's World Wide Web towards an increasingly machine-processable Web, with some of the recent Web 2.0 activities being an important intermediary step.

ENUM as introduced in section 2, on the other hand, is a comparably evolutionary development for the worlds of traditional telecommunications (telephony) and the Internet towards a significantly converged environment of the two. The distinction between User ENUM and Infrastructure ENUM as explained in section 2.10 means a further evolution making ENUM technology of increased interest to a larger community.

Both ENUM and the Semantic Web conceptually are focussing on a highly automated inter-machine communication rather than a human-machine communication. To identify relevant aspects of both ENUM and the Semantic Web, basic introductions have been given in sections 2 and 3. This section discusses issues connecting the Semantic Web and ENUM from a formal point of view. Based on this basic evaluation, possible advantages of a combination of the two areas are highlighted.

Taking the Semantic Web's layered architecture as guidance, it will be analysed from layer 1 up to layer 7 whether similarities or relations are to be found in ENUM. Furthermore, possible telecommunication applications of combined ENUM and Semantic Web technology in general will be pointed out.

6.1 Layer 1/2 Analysis: Identifiers and Documents

The similarities between ENUM and the Semantic Web start at the very basement layer with URIs building an integral part of both the Semantic Web and ENUM.

- On the Semantic Web, the URI is one of the major building blocks and being mainly used for naming all sorts of objects. As explained in section 3.2.1 objects in the context of the Semantic Web can range from people and their relations to things and their attributes or to ontologies and their classes and properties. All these objects (and many more) can be described using URIs.

- In ENUM, the URI is used in the form of URI RR (NAPTR RR) enabling the implementation of DNS redirect rules in ENUM with the final result being a URI, again.

As all these URIs are accessible and retrievable on the Web, they can be used in all sorts of applications, generating space for immediate interoperability between the Semantic Web and

ENUM. Furthermore, applications built for the Semantic Web or ENUM in general are prepared to work with URIs making integration of and interaction with the other area easier.

Regarding documents, as dealt with on the Semantic Web's layer 2, it is XML dominating the topic on the Semantic Web. XML provides a syntactical foundation with a high amount of interoperability, which allows concentrating on the larger issues of representing relationships and meaning that can be built on the basis of XML. In ENUM, information is stored in DNS NAPTR resource records (see section 2.7) providing simple rewrite rules. ENUM does not use a markup language as XML; the NAPTR records are plain text records, instead. However, these records come with a defined structure and syntax, allowing for a translation into other formats. Section 8.4 will introduce an application example dealing with that issue.

6.2 Layer 3/4 Analysis: Statements, Schemas and Ontologies

Looking at the occurrence of RDF vocabularies, ontologies and schemas in ENUM and the Semantic Web, this is an issue primarily related to the latter. While the Semantic Web is built around RDF and related standards, ENUM follows a considerably simpler approach as explained in previous sections.

At first glance, ENUM does not provide or use ontologies at all, as the information stored in NAPTR RRs only describes rather simple redirect rules for a single E.164 telephone number. However, as an ontology is defined as a way to describe the meaning and relationships of terms on the Semantic Web, the "ENUMservices" definition (see section 2.8) can be interpreted as describing the meaning of a specific URI. Figure 37 shows an example with an http: URI, where the ENUMservice finally denotes whether this URI is to be interpreted as identifier giving access to a website, to an announcement, to location information or to public key information. Additionally, ENUM holds information on the sequence URIs found should be processed in an ENUM resolving process, enabling the user to state preferences in ENUM how to be contacted.

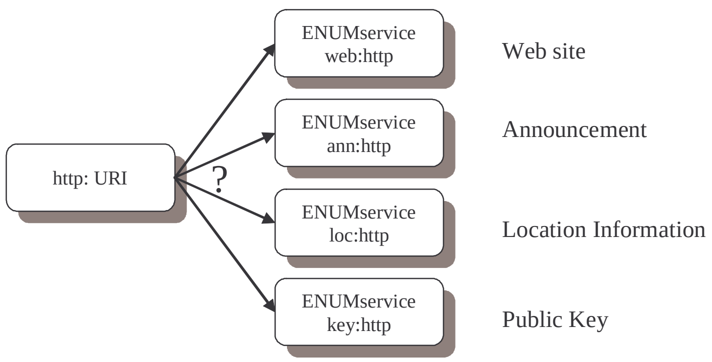

Figure 37: ENUMservice describing the meaning of an http: URI as example

Furthermore, ENUM information already publicly available in the DNS can be ontologically modelled to be used in combination with other ontologies or to be published again in a format better suitable for Semantic Web applications, i.e. in XML RDF. Section 9 introduces the SEMNUM RDF vocabulary created as a part of this thesis for ontological modelling of ENUM data. SEMNUM allows the serialisation of all data found in ENUM into an RDF XML document, making the ENUM data easily accessible for Semantic Web tools.

The combination of ENUM data with related ontologies is especially well suited to demonstrate the opportunities of integrating ENUM and Semantic Web concepts. An example for such an ontology is the FOAF vocabulary explained in section 5 and extensively used in applications presented in section 8; other vocabularies being well prepared for combination with ENUM include e.g. Dublin Core (DC) or BIO.
It is a further option to spice up existing taxonomies or ontologies with related information from ENUM. One example could be a taxonomy describing the structure of a company with departments, working groups, employees and their relations. In combination with ENUM and its communication-centric approach of providing related identifiers for e.g. e-mail, Web or telephony it becomes possible to implement an intelligent ("optimized") call forwarding mechanism taking into account facts and rules retrieved from the company taxonomy and related ENUM entries. Tasks like some calling party wanting to reach Mr. X from company A, alternatively another quality manager of that company, and as a third choice someone from the sales department, can be addressed using ENUM in combination with a taxonomy as described above, i.e. a system for naming and organising things into groups that share similar characteristics.

6.3 Layer 5/6 Analysis: Logic and Proof

Logic and Proof layer are meant to introduce logical principles to the Semantic Web allowing agents to make inferences and deductions. With such a system of logic implemented, it will become increasingly possibly to prove statements on the Web (see section 3.6 and 3.7). This leads to a situation where different people can write logic statements; machines can then be used to follow these "semantic links" trying to prove facts or derive new, formerly unknown facts.

Simplified, ENUM is a large database in the DNS providing redirect rules in response to qualified DNS queries. Logic and proof as used in Semantic Web terminology are conceptually not included in ENUM nor built into current ENUM implementations. However, ENUM implicitly carries information that can be used for reasoning purposes as performed on the Semantic Web. Considering ENUM data to be serialised into RDF XML (using the author's SEMNUM or other RDF vocabularies) documents, it becomes possible for Semantic Web reasoning engines to perform their operations on that transcoded data.
An example for such a reasoning using ENUM data is the task to "find all people on a given list that are affiliated with the Vienna University of Technology". The respective rule for this reasoning could be that people affiliated with Vienna University of Technology dispose of an e-mail address ending with the tuwien.ac.at domain. With the given persons' telephone numbers as a starting point, a reasoning engine can check the ENUM data (conveniently transcoded to RDF XML) for e-mail addresses containing the tuwien.ac.at domain. Those found to possess such an e-mail account are listed as University affiliates (as is the rule). This simple example[78] shows that although ENUM is not explicitly built to support the Semantic Web's logic or proof concept, there is enough information in ENUM capable of supporting the task, at least.

Therefore the synergies possible are twofold: On the one hand, Semantic Web technologies can help to improve search results on ENUM, and on the other hand existing Semantic Web standards can be spiced up with additional information gained from ENUM. Depending on

[78] See section 8.2 for further examples of ENUM being used in the context of logic and proof.

the respective logic and proof levels, the combination of ENUM and the Semantic Web definitely has the potential to add extra quality to both areas.

6.4 Layer 7 Analysis: Trust

Trust is a big issue in both the Semantic Web and ENUM. The Semantic Web's top layer is often referred as being a topic of ongoing discussion and research but minimal practical relevance, so far. In the past years, the other layers attracted more attention as the basement of the Semantic Web layer cake has to be built first. In ENUM, it seems to be just the other way round: trust was one of the early hot topics being primarily discussed in combination with the (legal) right of a subscriber to use a specific E.164 telephone number and how to validate this. In User ENUM, all rights regarding ENUM delegations are with the owner of the telephone number: Only that person is allowed to have his/her number delegated in ENUM, only that person is allowed to change the actual information contained in the ENUM NAPTR RRs. In fact, open validation issues nearly brought the ENUM development to a halt, one or two years ago. The involvement of national and international administrations (e.g. ministries, numbering authorities, regulators and domain registries) and standardisation organisations (e.g. ITU, ETSI and IETF) meant a massive administrative and bureaucratic hurdle to be taken. However, the early engagement in trust issues clearly brings advantages for ENUM nowadays.

6.4.1 Validity of Information on the Web

As discussed in sections 3.7 and 3.8, proof and trust are crucial issues for the Semantic Web as both are needed to realize the vision of a trustworthy Web, giving the user (at least) an indication of how proved or trusted a piece of information is.

In User ENUM, validity of information is primarily discussed in combination with the (legal) right of a subscriber to use a specific E.164 telephone number, as only the owner of a telephone number is allowed to have his/her number delegated in ENUM. In order to validate some person's right of use, various validation mechanisms have been developed and introduced by the ENUM community. Due to the implementation of those validation mechanisms a user now has a high degree of certainty that a number delegated in ENUM really is under the sole control of its owner.
Regarding the information stored within the NAPTR RRs (i.e. the redirect information), no proof or trust measures have been taken so far. As every ENUM domain is thought to be under the direct control of the ENUM subscriber freely editing personal data, the quality of data cannot be guaranteed. However, as invalid ENUM data would directly effect the owner's all-day electronic communication, it can be assumed that ENUM data normally is up-to-date and valid.

On the Semantic Web, validity of information is addressed using mechanisms belonging to the proof and trust layers. As discussed in section 3.8, trust is mainly secured by appending digital signatures, using encryption and introducing Web of Trust concepts. Intelligent proof verifiers and reasoning engines can be used to detect discrepancies in documents or statements.

While Semantic Web data can come with signature and encryption, ENUM delegations have the advantage of being validated by a certified authority. The combination of trust measures used on the Semantic Web and in ENUM, respectively, are well suited for supporting each other adding additional quality to the validity of data in both areas.

6.4.2 Circle of Trust

The circle of trust concept often is mentioned in the context of single-sign-on (SSO) policy, allowing a user to log-in with his/her credentials (e.g. user name and password) just once to get access to different applications or Web sites.[79] The concept is based on a mutual trust relationship between a number of applications or entities, taking advantage of a validation already done by someone else. With a circle of trust and SSO implemented, multiple user identities with multiple adjacent passwords are necessary no longer. One user identity with one password becomes sufficient to log on to multiple applications or Web sites. The downside is the vulnerability in the case of credentials being stolen or cracked, as multiple applications or Web sites then can be accessed by a trespasser.

The circle of trust concept could be used in combined Semantic Web/ENUM scenarios taking either advantage of the strict validation schemes implemented in ENUM or of the signed and encrypted information on the Semantic Web.

6.4.3 Web Services / Trust Services

6.4.3.1 Web Services

In the context of the Semantic Web, a Web Service is described as a resource on the Web (e.g. a Web site) that does not only provide rather static information but allows one to interact, to trigger some action or change of state. Web Services can be described, published, located and invoked over a network, generally the World Wide Web. These self-contained, modular applications can be anything from simple requests to complicated business processes. An example Web Service application is presented in section 8.1.2. On a commercial basis, different visions of Web Service platforms have been introduced, e.g. Microsoft .Net Services, Sun ONE, HP eSpeak and IBM's Web Services.

The Web Ontology Language for Web Services (OWL-S) as defined by the W3C [77], formerly DARPA Agent Markup Language for Web Services (DAML-S)[80], is providing an ontology of services, that will enable human users and software agents to discover, invoke, compose and monitor resources on the Web. The main parts of this ontology are

- The service profile for advertising and discovering services
- The process model which describes a service's operation in detail
- The grounding which describes how to interoperate with a service via messages

ENUM in this context can be described as a simple Web Service itself, as it gives one or more output parameters (i.e. URIs stored in NAPTR RRs) as an answer to one input parameter (i.e. the E.164 telephone number or the associated ENUM domain). ENUM therefore could be introduced as an additional Web Service on the Semantic Web; and the ENUM Web Service(s) could be described using OWL-S.

6.4.3.2 Trust Services

Web Services as described above use the inherently insecure Web for potentially mission critical business transactions. It is inevitably necessary to introduce trust into Web Services if

[79] Cp. http://www.projectliberty.org/, 28.09.2006.
[80] Cp. http://www.daml.org/services/, 28.09.2006.

they are to be used on a broad scale in business environments. Web Service trust architectures should address a variety of issues as outlined in [21] and listed as follows:

- Authentication: helps the service provider to be confident that the requestor is really the one he claims to be
- Authorization: helps the provider to determine whether the requestor is authorized to use the service
- Confidentiality: helps to protect transactions from unauthorized access
- Integrity: helps the service requestor as well as the service provider to be confident that the transaction data was not tampered with in transit
- Non-repudiation: helps the service requestor as well as the service provider to be confident that the other entity cannot deny the participation in the service

As mentioned in section 6.4.1 the trust discussion in ENUM so far was mainly focused on validation issues related to the subscriber’s right to use a number or its associated ENUM domain, respectively. However, the community’s focal point could easily move to the actual data stored in the NAPTR RRs itself, as soon as the first issue is solved satisfactorily.

The combination of ENUM and Trust Services could have potential for making the relevant ENUM data (i.e. the redirect rules or the URIs within the NAPTR RR) more trustworthy.
Another potential application area is in ENUM validation, where the introduction and use of Trust Services could help solving outstanding questions. Integrating the entities involved in today’s ENUM model in a Web Services Trust Architecture could bring possible benefits to this area.

6.5 Way forward

Sections 6.1 to 6.4 analysed ENUM and the Semantic Web in order to detect similarities and relations, and where convergent developments seem to be most promising.

The remainder of this thesis follows a merely application-oriented approach, though practical results are cross-checked with theoretic considerations and inputs from conferences, and vice versa. In the following, application examples are used to illustrate effective and potential benefits of introducing ENUM to the Semantic Web. At the same time the applicability of the proposed approach for real-world applications is evaluated; directions and tasks of future work in the area are pointed out.

7 PHOAF – A Prototype for ENUM and FOAF Queries

The theoretic concept of integrating ENUM technology and the Semantic Web's FOAF approach as outlined in previous sections needs a practical implementation in order to be effectively evaluated regarding the basic concept and to build a base for the implementation of application examples. Therefore a prototype developed during the course of this thesis is introduced in the following section. The prototype is named PHOAF, an acronym made up of the terms PHONE and FOAF indicating the combination of telecommunication and Semantic Web services, respectively.

7.1 PHOAF Functionalities

The main functionalities of the basic PHOAF prototype implementation are the following:

- Looking up the ENUM DNS database
- Retrieving the ENUM NAPTR data
- Detecting the location of FOAF RDF data (i.e. a foaf.rdf file) using different methods
- Parsing a FOAF file for data requested by the respective application.

It should be noted that an extended version of the basic PHOAF prototype is used to implement an example application introducing a unique key to access Web 2.0 data. The increased functionalities are described in detail in section 8.6.

7.2 PHOAF Architecture

PHOAF is a prototype basically coded in Java programming language[81] with the full code being made available on the DBAI Web site.[82] The PHOAF architecture is designed with a clear focus on simplicity, modularity and re-usability. It was the aim to demonstrate that the combination of ENUM and the Semantic Web's FOAF is possible using standard tools and libraries widely available.

7.2.1 Java Programming Language and Platform

As described in detail on Sun's Java Web site, Java technology is both a programming language and a platform.[83] The Java programming language is a high-level language that can be characterised as being simple, object-oriented, distributed, multi-threaded, dynamic, architecture-neutral, portable, robust and secure. In Java, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the Java compiler (javac). A .class file does not contain code that is native to a

[81] Cp. http://java.sun.com/, 20.08.2006.
[82] Cp. http://www.dbai.tuwien.ac.at/proj/semnum/, 26.11.2006.
[83] Cp. http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html, 20.08.2006.

specific processor; it instead contains bytecodes – the machine language of the Java Virtual Machine. The Java launcher tool (java) then runs the application with an instance of the Java Virtual Machine.

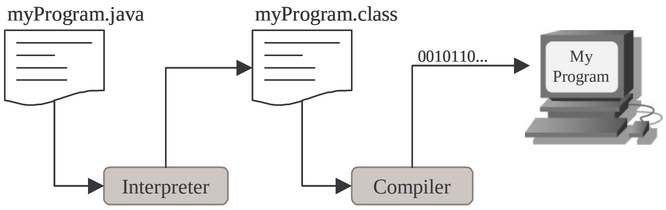

Figure 38: Running Java code on a computer

Because the Java Virtual Machine is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris TM Operating System (Solaris OS), Linux, or MacOS.

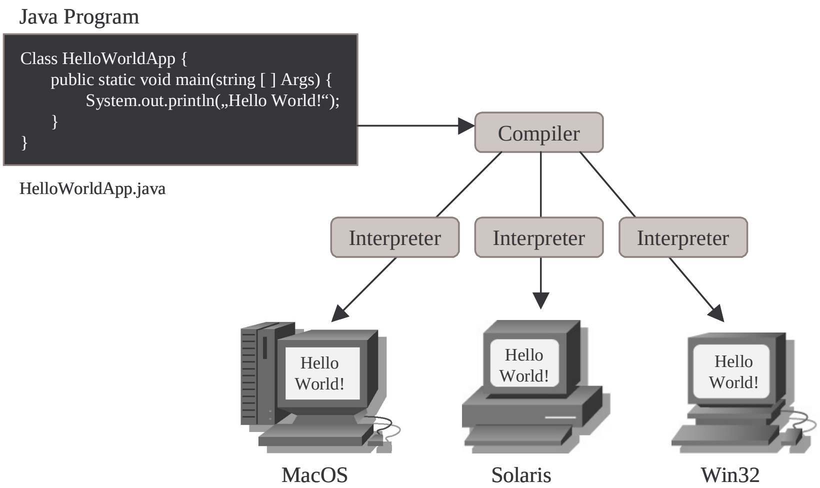

Figure 39: Running Java code on different operating systems

A platform is the hardware or software environment in which a program runs, e.g. Microsoft Windows, Linux, Solaris OS and MacOS that are some of the most popular platforms. Most platforms can be described as a combination of the operating system and underlying hardware. The Java platform differs from most other platforms in that it is a software-only platform that runs on top of other hardware-based platforms. The Java platform has two components:

- The Java Virtual Machine
- The Java Application Programming Interface (API)

The Java Virtual Machine is the base for the Java platform and is ported onto various hardware-based platforms. The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. It is grouped into libraries of related classes and interfaces; these libraries are known as packages.

The following Figure 40 depicts how the Java API and the Java Virtual Machine insulate the program from the hardware.

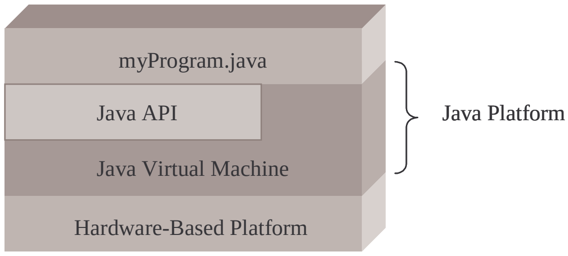

Figure 40: Java platform insulating program from hardware

As a platform-independent environment, the Java platform can be a bit slower than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability.

7.2.2 Using Java with PHOAF

PHOAF utilises the Java 2 Standard Edition 1.4 (JDK)[84] namely using the package javax.naming.directory (for DNS query) and the Java Swing[85] library's HTML parser (for FOAF link tag detection). Additionally, the SAX parser[86] is used for XML parsing of FOAF files.

The JDK javax.naming.directory extends the javax.naming package to provide functionality for accessing directory services. This package defines the directory operations of the Java Naming and Directory Interface (JNDI). It is designed to be independent of any specific naming or directory service implementation allowing a variety of services to be accessed in a common way. Summing it up, the package allows applications to retrieve and update attributes associated with objects stored in a directory, and to search for objects using specified attributes.

- Directory Context: The DirContext interface represents a directory context and defines methods for examining and updating attributes associated with a directory object. getAttributes() are used to retrieve the attributes associated with a directory object (for which the name has to be supplied). Attributes can also be modified using modifyAttributes(). It is possible to add, replace, or remove attributes and/or attribute values using this operation.
- Searches: DirContext contains methods for performing content-based searching of the directory. In the simplest and most common form of usage, the application specifies a set of attributes to match, and submits this attribute set to the search() method.

PHOAF utilises the javax.naming.directory's capabilities to query the DNS and to retrieve the ENUM NAPTR RR data.

Another package used with PHOAF is javax.swing.text.html.parser. The HTML parser contained in that package allows retrieving information from Web pages coded in HTML. The Java Swing HTML parser looks for defined information units in the HTML code that is

[84] Cp. http://java.sun.com/j2se/1.4.2/, 26.11.2006.
[85] Cp. http://java.sun.com/docs/books/tutorial/uiswing/index.html, 26.11.2006.
[86] Cp. http://www.saxproject.org/, 26.11.2006.

the HTML tags. Such events, i.e. the occurrence of specific HTML tags, are triggering call backs (see Figure 41).

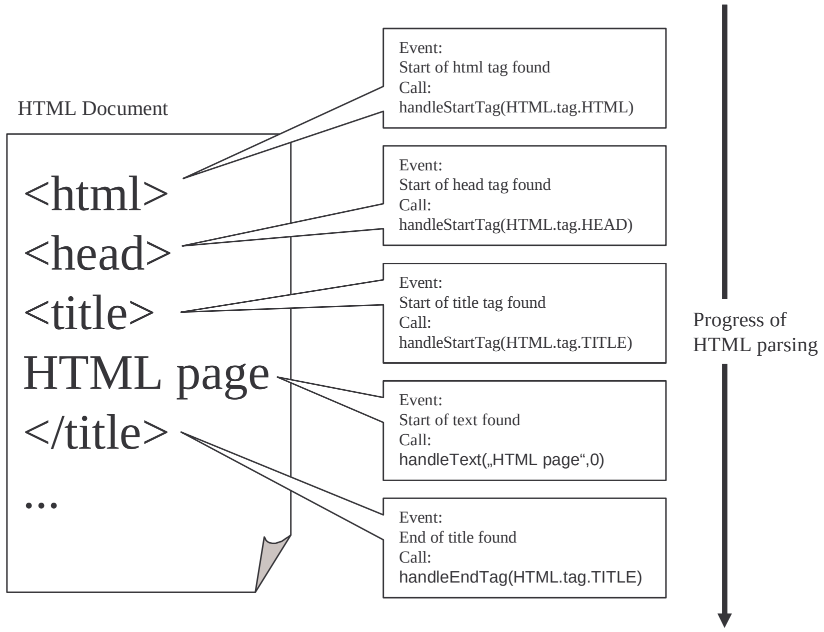

Figure 41: Operation principle of Java Swing HTML parsing[87]

PHOAF utilises the Java Swing HTML parser when looking for information in the HTML code where a foaf.rdf could be located, i.e. looking for a `<link>` tag in the `<head>` section of the HTML code (see section 5.4).

SAX stands for Simple API for XML, originally a Java-only API. SAX was the first widely adopted API for XML in Java, and has become a “de facto” standard. The SAX parser implementation follows an event-driven approach where the programmer provides callback methods. These methods are invoked by the parser as it traverses an XML document. The current version (at time of writing) is SAX 2.0.1, and there are versions for several programming language environments other than Java, such as Python, Perl, Pascal, C/C++, and COM. For Java, SAX is another package named org.xml.sax with extensions available in package org.xml.sax.ext.[88]

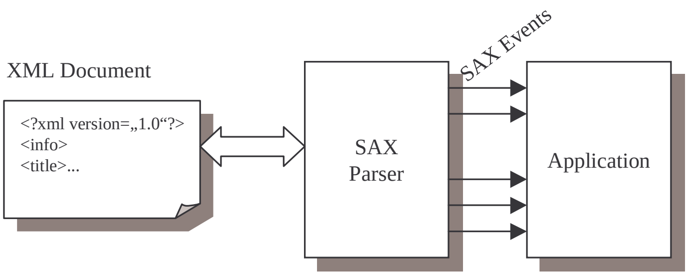

Figure 42: SAX parser reporting a document to an application as a series of events

PHOAF utilises SAX for parsing the FOAF/RDF files triggering on certain FOAF properties and retrieving the associated content.

---

[87] Cp. http://www.inf.fu-berlin.de/inst/ag-nbi/lehre/0304/V_NP/folien/10-HTML.pdf, 17.08.2006.
[88] Cp. http://www-128.ibm.com/developerworks/xml/library/x-saxapi/, 17.08.2006.

PHOAF is implemented as Web front-end in Java Server Pages (JSP)[89] directly calling static methods of the utility classes. JSP technology provides a simplified, fast way to create web pages that display dynamically-generated content. The JSP specification defines the interaction between the server and the JSP page, and describes the format and syntax of the page. JSP pages use XML tags and scriptlets written in the Java programming language to encapsulate the logic that generates the content for the page. It passes any formatting (HTML or XML) tags directly back to the response page. In this way, JSP pages separate the page logic from its design and display.[90]

7.2.3 PHOAF Implementation

Figure 43 provides an overview regarding the practical implementation of the PHOAF prototype. Four distinct building blocks all connected to the Public Internet[91] can be identified.

- Web and Application Server (hosting the PHOAF prototype)
- ENUM Name Server (hosting ENUM data)
- Web Server (hosting FOAF data)
- User Terminal

The core of the PHOAF deployment is two servers located on DBAI premises; a web server and an application server (that both can run on one physical machine). The web server is responsible for hosting the web page to be presented to the user. The web page is coded in HTML with code snippets embedded. The snippets contain Java code and are separated from the rest of the code with "<%……%>". The application server is executing the Java Runtime Environment (JRE). The server-sided PHOAF JSP Java Code can get accessed from a client outside (i.e. anywhere on the public Internet) allowing communication between client and server.

The server used in the practical PHOAF implementation deployed at DBAI during the course of the work on this thesis is an Apache Tomcat application server.[92]

[89] Cp. http://java.sun.com/products/jsp/, 17.08.2006.
[90] Cp. http://www.torsten-horn.de/techdocs/jsp-einfuehrung.htm, 17.08.2006.
[91] It will be shown in section 8 that it can be advantagous for specific use cases to operate PHOAF in closed environments, e.g. in private or corporate social networking applications. In those cases PHOAF may not be connected to the Public Internet, but work in private or corporate networks instead.
[92] Cp. http://tomcat.apache.org/, 26.11.2006.

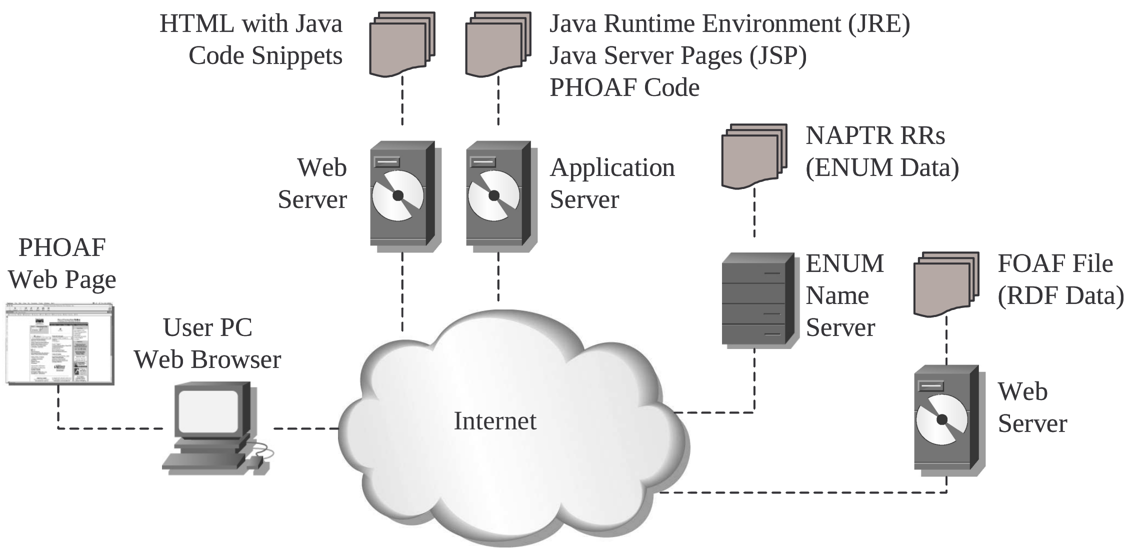

Figure 43: PHOAF practical implementation

Another building block is the server for ENUM and FOAF data, respectively. The ENUM name servers are part of the DNS infrastructure (see section 2.5); in the ENUM use case these servers are mostly run by ENUM registrars (ENUM tier 2) or private entities (hosting their own name server; ENUM tier 3), and act as a large database for hosting individual NAPTR RRs.
The FOAF files are stored on a Web Server chosen by the individual user; typically this is on some private or corporate Web space used for e.g. homepage content.

Finally, a client PC is needed to access the PHOAF prototype from anywhere on the Internet. As PHOAF is coded using JSP, there is no JRE needed on client-side. A standard Web browser (or any other client capable of accessing the Internet) is all needed to communicate with the PHOAF prototype.

7.3 PHOAF Workflow

PHOAF allows three different input parameters to start a query (Figure 44):

- Start ENUM DNS NAPTR query with given phone number – the result is a list of URIs and services of the queried phone number's owner
- Start FOAF RDF file lookup with given Web URL – the result is the exact location of a foaf.rdf file
- Start FOAF RDF file parsing with given location of foaf.rdf file – the result is data available from the foaf.rdf file

Appropriate results of any of these three operations can be used to start any further query. Should an ENUM query result contain another phone number, that phone number again can be used as input for another ENUM DNS NAPTR query. The same applies to FOAF data results, with e.g. a phone number used for an ENUM DNS NAPTR query or a website URL for a FOAF RDF file lookup.

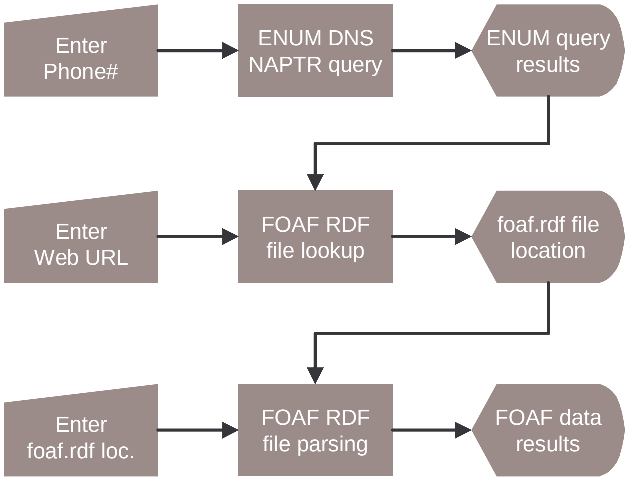

Figure 44: Flowchart of basic PHOAF prototype operations

The start screen of the PHOAF prototype reflects that three possible starting options with an input field each for entering a telephone number, an unspecified Web URL and an exact Web URL of a FOAF RDF file, respectively (see Figure 45).

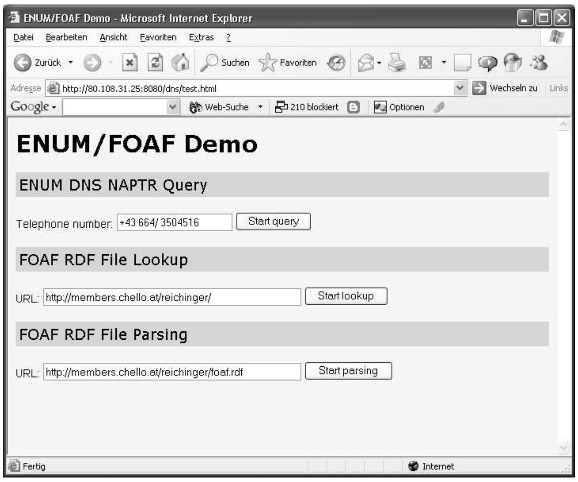

Figure 45: Screenshot of basic PHOAF start screen

The results of the queries can be manifold. In the case of an ENUM DNS NAPTR query using a given phone number a typical result is shown in the screenshot from Figure 46. The query result page indicates a successful query with two e-mail addresses and one web URL returned. As it is not clear in the first instance whether the web:http URI is just a simple website or the location of a foaf.rdf file, both “Visit website” and “Lookup FOAF RDF file” are listed as options for further action.

Choosing the “Visit website” option opens the website described by the URL retrieved in a browser window, while choosing “Lookup FOAF RDF file” starts another PHOAF operation looking to find a valid FOAF RDF file (named foaf.rdf) at the retrieved location.

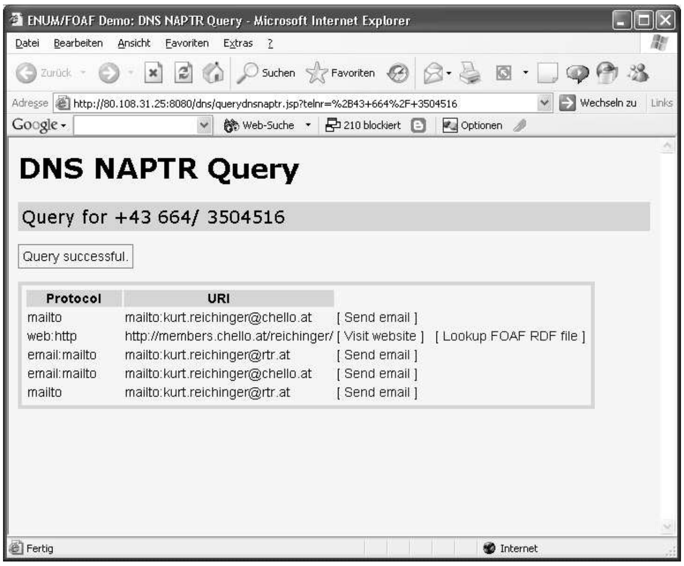

Figure 46: Screenshot of basic PHOAF DNS NAPTR query result

Looking for a FOAF RDF file can be done in different ways, as there are different ways the existence of such a file can be annotated (see section 5.4). PHOAF is designed to work with two options:

- Explicit annotation of FOAF RDF file using the HTML link tag: This method works in a similar way bloggers are pointing to their RSS feeds. A link is placed in the <head> section of an HTML file and looks like the following:
```
<link rel="meta" type="application/rdf+xml" title="FOAF" href="foaf.rdf" />
```
- No explicit annotation of FOAF RDF file: In this case, a given URL (e.g. found by doing an ENUM DNS NAPTR query) can be used for an "educated guess" regarding the location of a FOAF RDF file.

There are several ways of specifying a URL making the automatic detection of some file's correct location a non-trivial task. The following list shows a few examples of annotating a Web URL.

- Without the HTTP referrer, i.e. www.example.com
- With HTTP referrer, i.e. http://www.example.com
- With HTTP referrer and a slash at the end, i.e. http://www.example.com/
- With HTTP referrer, slash and the name of some file (most often the file named is the index.htm or index.html file), i.e. http://www.example.com/index.html

The FOAF RDF file detecting algorithm makes the following assumptions for detecting the location of a FOAF RDF file based on a given Web URL.

- First, PHOAF assumes the FOAF RDF file to be named foaf.rdf as suggested in the FOAF specification. In principle, the creator of a FOAF RDF file is free how to name that file, however, it has become a common understanding among FOAF users to stick to the specification proposal.

- Second, PHOAF assumes the FOAF RDF file to be located in the root path of a Web site, i.e. http://www.example.com/foaf.rdf. Should the FOAF RDF file be located in some other place on a Website, e.g. http://www.example.com/userprofile/foaf.rdf, this has to be indicated using the link tag in the header of the index.html file in the root path.

The PHOAF FOAF RDF file detection algorithm takes all that distinctions into account when searching for the foaf.rdf file based on some unspecificly annotated Web site URL.

Assuming a FOAF RDF file lookup having been successful, PHOAF offers to "Parse FOAF file", i.e. to subsequently retrieve information available from the foaf.rdf file. The PHOAF parsing algorithm is based on a simple XML parser, screening an XML/RDF file for the FOAF RDF properties needed by the applications proposed in this paper. However, it is an option to utilise RDF Query Languages [4] in future PHOAF versions. The screenshot from Figure 47 shows a typical result of such a FOAF RDF file parsing, first indicating the successful parsing and then listing the specific results, namely the person's full name, a phone number, a homepage and a link to a depiction. In addition, the parsing retrieves the names of two other known persons (Robert Baumgartner and Gerd Reichinger, both annotated using foaf:knows). One of these two persons (Gerd Reichinger) is found to be described having its own FOAF RDF file (annotated using rdfs:seeAlso) which can be parsed further, again using PHOAF.

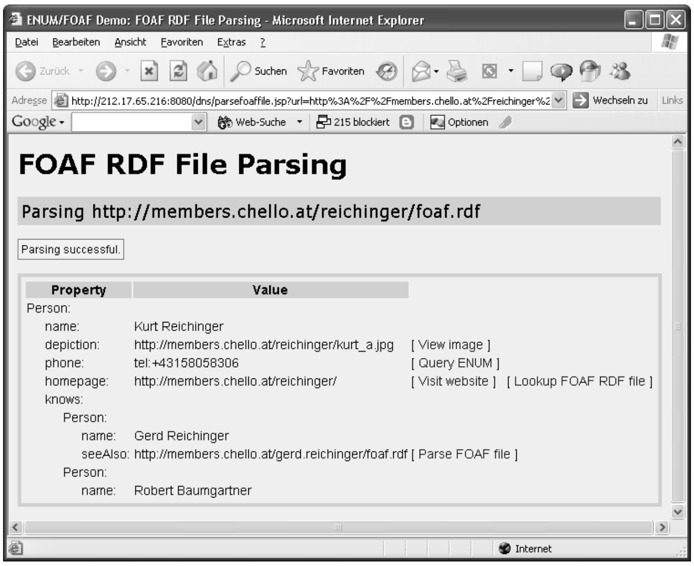

Figure 47: Screenshot of basic PHOAF FOAF RDF file parsing result

PHOAF offers the user to further access the results of the FOAF RDF file parsing by simple clicking on a link next to the result.

- When a depiction link is found, the image can be viewed.
- When a phone number is found, ENUM can be queried (using the PHOAF capability).
- When a Web site URL is found, that Web site can be accessed using the default Web browser or a FOAF RDF file lookup can be performed (using the PHOAF capability).
- When a rdfs:seeAlso tag, i.e. a FOAF file URL, is found, that FOAF file can be parsed (using the PHOAF capability).

The PHOAF building blocks and capabilities are designed in a modular way, allowing applications to use those blocks that are needed for fulfilling specific application tasks. As code and interfaces are open source it is easy to integrate PHOAF in existing applications or to build new ones incorporating PHOAF functionalities. In section 8 and 8.6, PHOAF-based application examples realised during the course of this thesis as well as suggestions for possible applications utilising PHOAF will be introduced and assessed with regard to usability and potential impact.

8 Applications utilising PHOAF

In this section application examples combining ENUM and the Semantic Web are presented and introduced in detail. All but one example take advantage of the PHOAF prototype's basic capabilities described in section 7; for one example application a new PHOAF version with extended functionalities was coded.

- VoIP Called Party Information Presentation: is an example application enriching the result of a standard ENUM query with information from FOAF. This is especially useful in VoIP where a full set of information on calling or called party can be presented to the other party on the screen of the VoIP device.

- VoIP Call Forwarding on Called Party Affiliation: is an example application using the taxonomy of company structures and employee working relationships with ENUM for automatic redirection of incoming calls.

- Phonebook Contact Network: is an example application spicing up an ordinary personal phonebook as used in mobile phones or personal digital assistants with information retrieved from ENUM and FOAF.

- ENUM/FOAF to RDF Transcoding: is an example application simply transcoding information available in ENUM into information accessible by Semantic Web tools.

- Trust on Corresponding Data in ENUM and FOAF: is an example application introducing a trust indicator on corresponding data in ENUM and FOAF.

- A Unique Key to Web 2.0 Application Content: is an example application introducing how to access a single person's Web 2.0 application content distributed on the WWW by means of a combination of ENUM and FOAF. For this application example a new ENUMservice is introduced, the FOAF vocabulary is extended and the PHOAF prototype is upgraded in order to support the application specific requirements.

8.1 VoIP Called Party Information Presentation

8.1.1 Application Features and Functional Range

The first application example combining ENUM and the Semantic Web by utilising PHOAF is enriching the result of a standard ENUM query. It uses a telephone number entered by the user (or an agent) to search for relevant data in ENUM and FOAF, and to present that information in aggregated form to the user.

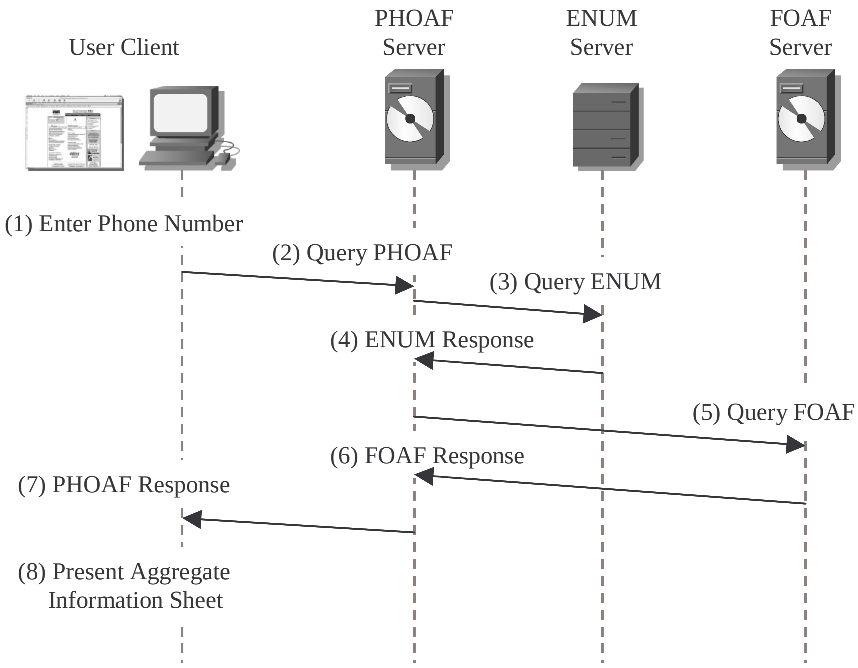

Figure 48: Workflow (simplified) of application utilising PHOAF for presentation of information on a contacted person or company

Figure 48 shows the workflow of the application. On entering a telephone number in the international format (1) the user client sends a query to the PHOAF Web server (2). The PHOAF prototype reacts with a query towards the DNS (3) looking for ENUM DNS NAPTR RR associated with the telephone number provided by the user client. The ENUM name server answers either by sending a set of NAPTR RRs, or a message declaring that no NAPTR RRs have been found (4). In the latter case, a respective message is sent back to the user client for immediate presentation and the PHOAF process is terminated. If one or more NAPTR RRs are received, PHOAF looks for a NAPTR RR with ENUMservice "web" entry. If successful, PHOAF tries to detect the correct Web URL of the FOAF RDF file (5) and to retrieve the data, subsequently (6). Finally, PHOAF prepares the retrieved information in aggregated form and submits it back to the user client (7) for immediate presentation (8).

In the following, application screenshots are presented, showing the start screen (Figure 49) and a typical output screen of a PHOAF query (Figure 50).

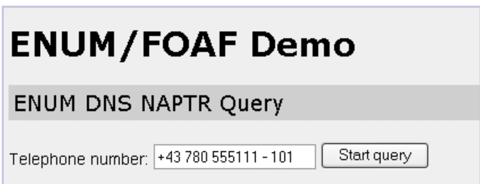

Figure 49: Screenshot of start screen for application utilising PHOAF for presentation of information on a contacted person or company

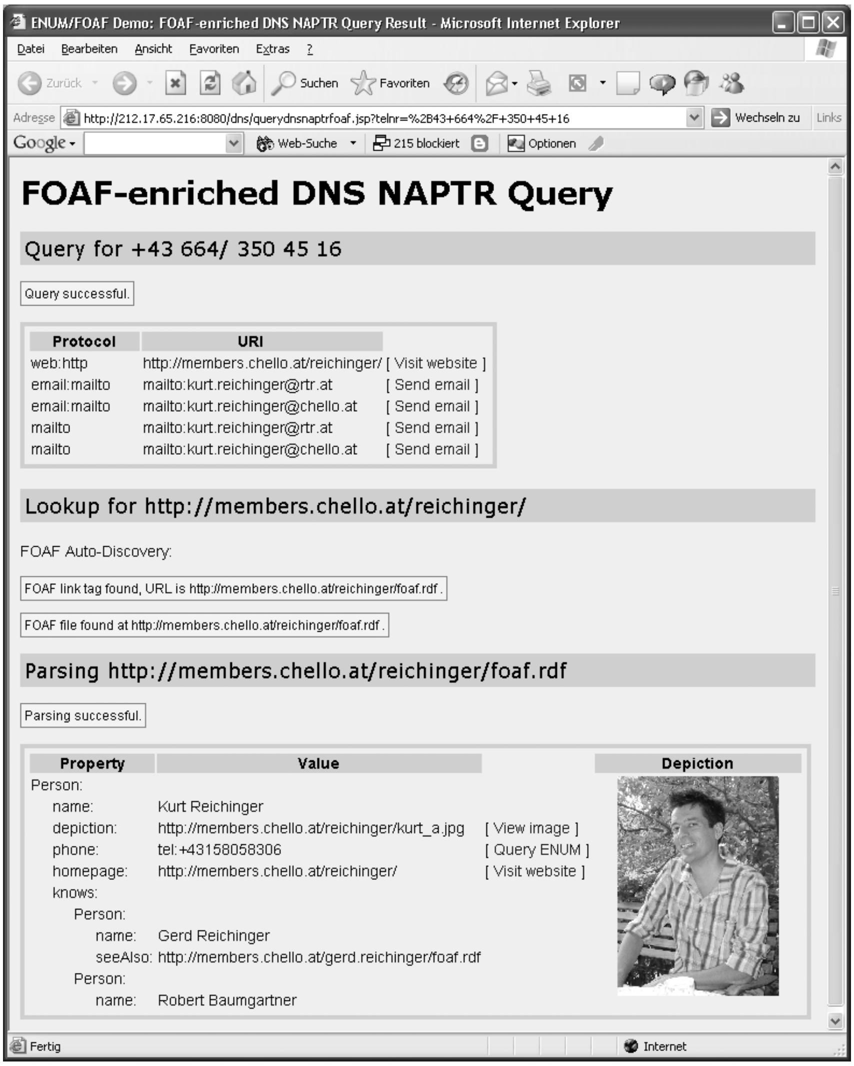

Figure 50: Screenshot of results for application utilising PHOAF for presentation of information on a contacted person or company

The application example shows (see screenshot from Figure 50) the called party's full name (i.e. Kurt Reichinger), a private homepage URL (http://members.chello.at/reichinger/) and two e-mail addresses (kurt.reichinger@chello.at and kurt.reichinger@rtr.at). All these information has been retrieved by means of an ENUM query, i.e. this data was stored in DNS NAPTR RRs.

Furthermore, the application returns another phone number (+43-1-58058-306) and a link to a depiction (http://members.chello.at/reichinger/kurt_a.jpg). In the case of a depiction link being detected, the application provides immediate presentation of the associated picture (see that guy with the smile on his face in Figure 50).

In addition, the application indicates two other persons to be known by the called party (by means of the foaf:knows property) with one of them (Gerd Reichinger) being described by an own FOAF RDF file.

8.1.2 Fields of Application

The application works with a telephone number in the international format as input. This input can be provided either manually by a human user (as in the example implementation from section 8.1.1) or by an agent importing the telephone number from a VoIP client, for instance.

In this type of application, PHOAF can be used in two principle implementation scenarios:

- Figure 51 shows a PHOAF implementation as a stand-alone client on the user's device. This can be e.g. a Java-enabled mobile phone using GPRS or UMTS to connect to the Internet and to directly access the servers holding ENUM and FOAF data, respectively. Therefore, both ENUM and FOAF queries are initiated and executed by the PHOAF client running on the end user's device.

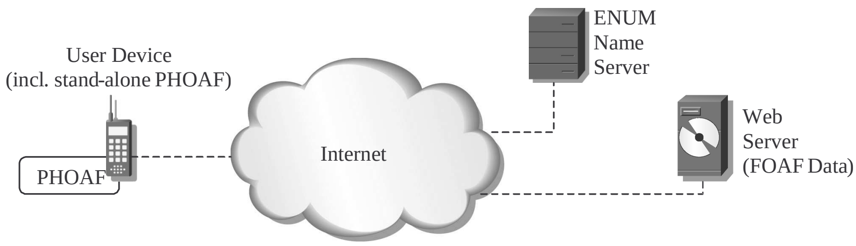

Figure 51: PHOAF implementation as stand-alone client on user device

- Figure 52 shows a server-sided PHOAF implementation with a VoIP provider being responsible for the ENUM/FOAF queries and transmitting data found (or explicitly requested by the calling party) to the customer by means of a push service. Assuming that the user client contacts the VoIP provider during the course of every VoIP call setup (which is the cause in a standard SIP setup procedure), the VoIP provider uses PHOAF to prepare an information fact sheet on the called party using the phone number submitted by the calling party that is pushed back to the customer during call setup.

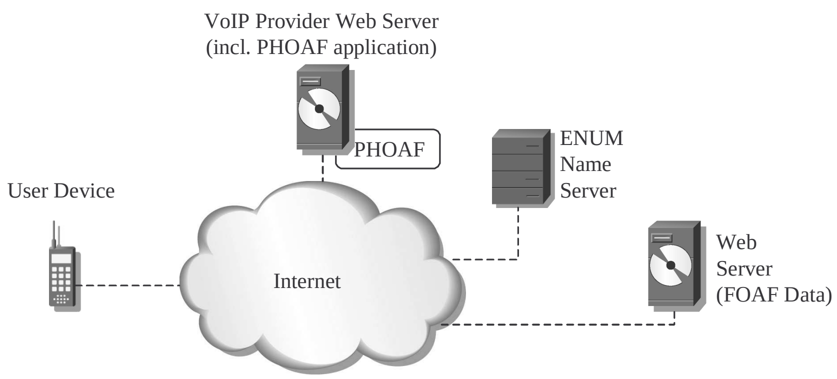

Figure 52: Server-sided PHOAF implementation in VoIP scenario

While the example applications from Figure 51 and Figure 52 provide one (calling) party with additional information on another (called) party, it is also perfectly possible to do the direct opposite. When a call is set up using a telephone number, the called party is provided with that telephone number (a.k.a. calling line identity or CLI). The CLI can be presented to the called party on the display of the end user's device. By knowing the calling party's phone number, it becomes possible for the called party to utilise PHOAF in order to get information on the calling party by means of an information fact sheet similar to the one presented to the calling party in the examples above.

The application examples presented above show two options how to implement PHOAF – on client-side and on server-side. A third option is to implement PHOAF as a Web Service, and offer its functionalities to all parties (end customers and providers) being able to access that service.

The W3C defines a Web Service as a software system designed to support interoperable machine-to-machine interaction over a network.[93] A Web Service has a public interface described in a machine-processable format, e.g. Web Services Description Language (WSDL). Essentially, this is an XML-based service description on how to communicate using the Web Service. Systems wanting access interact with the Web Service as prescribed by that public interface using messages that may be enclosed in SOAP (formerly for Simple Object Access Protocol) envelopes. Typically, these messages are conveyed using HTTP and normally comprise XML and other Web-related standards. Another building block of a Web Service is the Universal Description Discovery and Integration (UDDI) specification that is used to publish the Web Service information. The purpose of that protocol is to enable applications to look up the Web Service information in order to determine whether the respective Web Service fulfils the specific application requirements, or not.

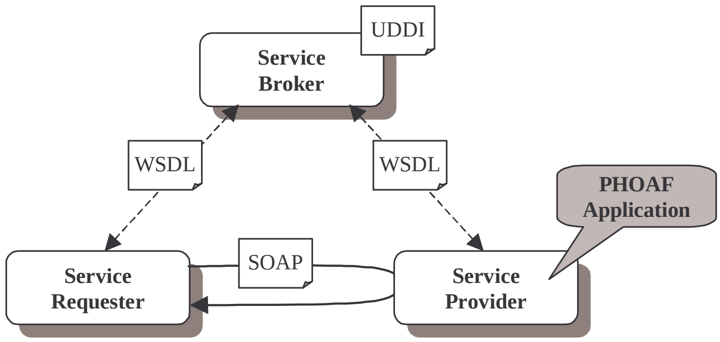

Figure 53: Implementing PHOAF application as Web Service[94]

Figure 53 explains a possible implementation of the PHOAF application as a Web Service. The PHOAF Web Service resides at the Service Provider edge getting announced using WSDL. A party wanting to access the PHOAF Web Service obtains information on the Web Service by contacting the Web Service Broker, also using WSDL. Assuming that the PHOAF Web Service fulfils the Service Requester's needs, the PHOAF Web Service is accessed using SOAP.

It is of further importance to note that these applications are perfectly suitable for all types of services using telephone numbers – be it fixed line telephony, mobile telephony or VoIP telephony.

Additional options are the integration of PHOAF in Semantic Web Portals [68] or to use a telephone number as unique key to semantic personalisation information in order to give a user optimal support in accessing, retrieving and storing information [5].

[93] Cp. http://en.wikipedia.org/wiki/Web_service, 17.08.2006.
[94] Cp. http://en.wikipedia.org/wiki/Image:Webservices.png, 17.08.2006.

8.2 VoIP Call Forwarding on Called Party Affiliation

The second example application deals with automatic call forwarding in VoIP environments depending on department (or working group) affiliation and other parameters of called and/or calling party.

Generally, a Public Branch Exchange (PBX) comprises an internal routing scheme allowing forwarding of a call to another extension when the one originally dialled is not available. This can be hard-coded in the PBX (e.g. fall-back to front office on extension busy) or individually programmed on the device (e.g. forward all calls to colleague's extension across the room). However, these forwarding rules are often far from optimal. The PHOAF application example presented in this section uses various sources (providing background information on called and calling party including their mutual relationship) in order to reach an "optimised" call forwarding decision.

For the example discussed, a company with an ENUM-enabled IP-PBX, i.e. a PBX based on VoIP technology and being capable of performing ENUM queries, should be assumed first. Furthermore, the extensions are assumed to be delegated in ENUM and all employees to have their own FOAF files populated with business relation-relevant data. Figure 54 shows a simple IP-PBX scenario with five extensions all being delegated in ENUM, and everyone having its own FOAF file published.[95]

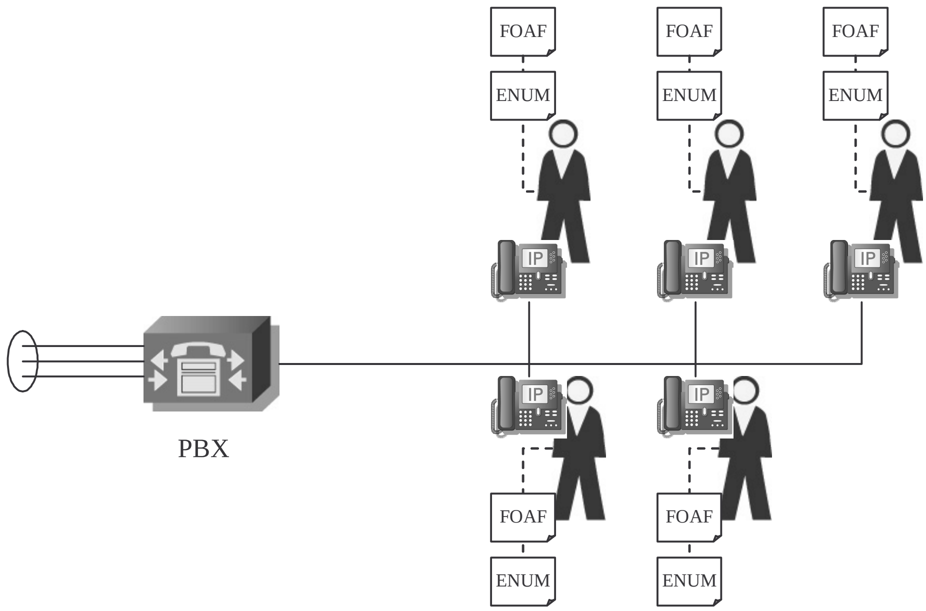

Figure 54: ENUM- and FOAF-enabled PBX

Figure 55 describes a typical call flow. An incoming call to one of the company's employees (B) will first enter the PBX coming from the traditional PSTN or from an IP-based network (1). As the PBX is assumed to be ENUM-enabled, it first performs an ENUM query (2) to an associated ENUM name server. This name server can be public or private, allowing for privacy of all data in the latter case. With the ENUM query performed, the PBX knows about the current preferences of the employee associated with the extension dialled (3) and can forward the call accordingly, e.g. directly routing it to the SIP address of the device at the

[95] As the VoIP Call Forwarding application example presented is to be used in corporate environments, the ENUM and FOAF entries typically will not be administered and maintained by the individual users but by some authorised entitiy (e.g. the IT department) according to corporate governing rules applying.

employee's desk, to a mobile phone or to a voice box (4). In the case of the extension dialled being busy or not available (5), the PBX consults the associated FOAF file (6). The exact URL of that FOAF file has already been determined with the initial ENUM query. In that application example's simplest form, the FOAF file associated with the extension dialled holds information on the employee/extension (C) the call should be forwarded to (7), e.g. by means of rdfs:seeAlso property. After a further ENUM query to retrieve the preferences of employee C (8, 9) the call can finally be routed to employee C (10).

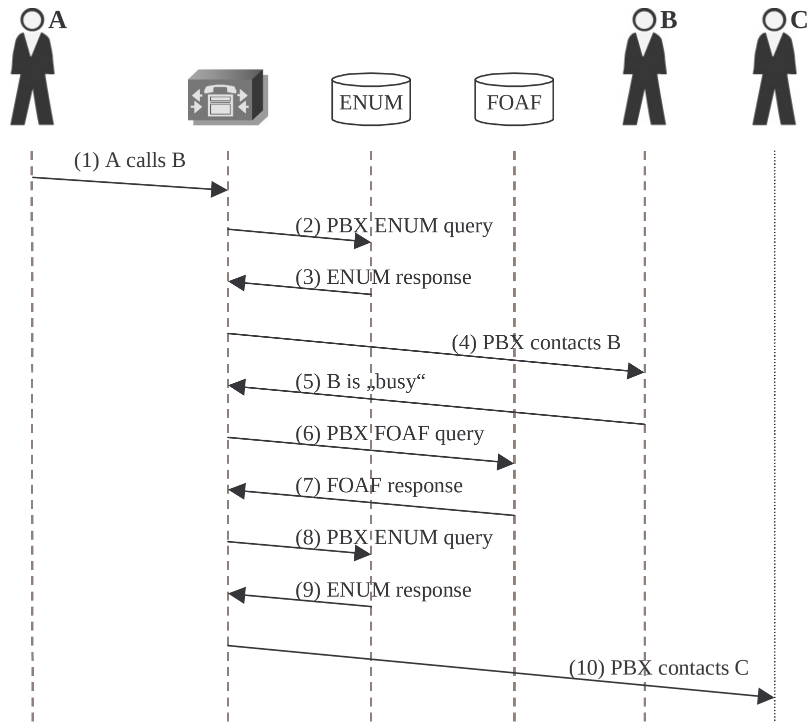

Figure 55: Call flow in ENUM-enabled IP-PBX

More sophisticated solutions can retrieve and utilise additional information about the called party, e.g. company and department structures, internal and external working groups or the employee's electronic diary, like Microsoft™ Outlook or IBM™ Lotus Notes. Together with information collected on the calling (!) party – using the incoming CLI for subsequent ENUM and FOAF queries – it becomes possible for intelligent PBX algorithms to calculate an optimised call forwarding target. In that way, the Semantic Web's reasoning capabilities add extra value by enabling the establishment of well-founded or semantically calculated redirect options or rules. This could be used in the form of a Computer Telephony Integration System typically installed in call centre environments for Customer Relationship Management.

8.3 Phonebook Contact Network

The Phonebook Contact Network example is spicing up an ordinary personal phonebook (directory) – as typically found in mobile phones or personal digital assistants – by means of additionally available ENUM and FOAF RDF information.

Modern end user communication devices, e.g. mobile phones, fixed line phones or personal digital assistants (PDA), typically dispose of a personal phonebook or directory. In most cases at minimum a contact's name and a phone number are stored on such devices for easy dialling and informing the owner on incoming calls (presenting the calling party's name on the display instead of a number).

In this environment PHOAF can be used to find additional information on the contacts already in the phonebook and even find new contacts. PHOAF first performs an ENUM DNS NAPTR query for each and every single phone number found in the personal phonebook. The data found in ENUM already means a valuable enhancement to a user's phonebook as URIs like e.g. e-mail address, SIP AoR, Web site URL or fax number could be added to the personal phonebook (see Figure 56).

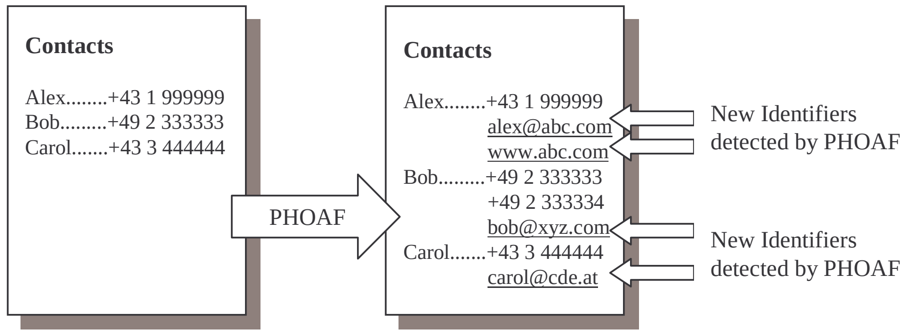

Figure 56: Personal phonebook on end user device (e.g. a mobile phone) before (left) and after (right) enhancing with ENUM data

However, it is the FOAF RDF file lookup and the FOAF RDF file parsing that adds even greater extra value. Using the information available from FOAF, it becomes possible to discover relations between contacts as well as previously unknown contacts. The data retrieved can be used for modelling a network of primary (already known) and secondary (previously unknown) contacts. Such a network can be built with contacts modelled as nodes and their corresponding relations modelled as directed arcs. A relation of two persons is annotated as a triple, with the arc pointing from the person that has stated to know another person (by means of a FOAF-knows relation) to that other person. As triples are one of the building blocks of RDF, the resulting phonebook contact relations can be easily annotated by e.g. a new FOAF RDF file forming a network of persons, their relations and individual descriptions.

Figure 57 shows the sequence of actions necessary. First, a simple network is created out of data directly available from the personal phonebook with the phonebook owner as central node and the contacts as adjacent nodes. In a second step, PHOAF performs ENUM and FOAF queries searching for further details of the contacts. With PHOAF following semantic links in FOAF RDF files (see the rdfs:seeAlso property in section 5.3) the original network begins growing. With more contacts (nodes) and relations (arcs) being detected and added to the initial network by PHOAF a web of contacts gets woven – the Contact Phonebook Network.

The advantage of this method of creating a social network is its immediate applicability. All that is needed is a personal phonebook with some contacts entered, a wide adoption of ENUM and FOAF, and the capabilities of the PHOAF prototype. In contrast to platforms like XING[96] or MySpace[97], it is not necessary to enter each and every contact on a Web site as the contacts

[96] Cp. http://www.xing.com, 02.12.2006.
[97] Cp. http://www.myspace.com, 02.12.2006.

Web is woven out of a couple of telephone numbers. The result however, could be entered into platforms as mentioned above.

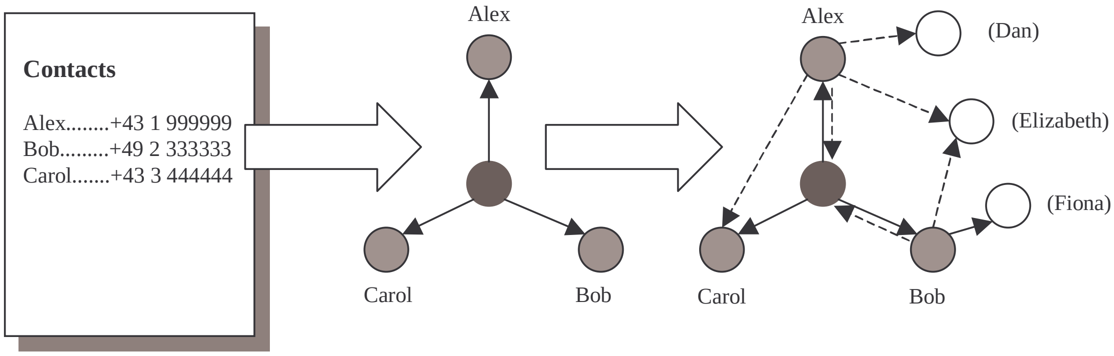

Figure 57: Weaving a contact's network out of a simple mobile phonebook utilising PHOAF

In the example from Figure 57 the phonebook owner has three (primary) contacts stored in the contacts list. Primary contacts are indicated by dark grey nodes and are connected to the phonebook owner with solid arcs. PHOAF is able to detect more information on these contacts:

- Alex has stated in his FOAF file to know the phone book owner[98], Carol and two other persons (i.e. Dan and Elizabeth) both annotated as white (secondary) nodes in Figure 57. The relations detected with PHOAF are labeled as secondary relations indicated by dashed arcs.
- Bob on the other hand has stated in his FOAF file to know the phonebook owner, Elizabeth (also known by Alex) and another person, Fiona.
- It has to be noted that Dan, Elizabeth and Fiona have not been known to the phonebook owner prior to the first run of PHOAF.

Assuming that a large amount of the contacts originally found in the personal phonebook have their own ENUM and FOAF entries, applying the PHOAF Phonebook Contact Network application may result in a huge network of (known and unknown) contacts tightly woven to each other. It is therefore left to the phonebook owner to decide whether those new contacts (as well as their relations) should get added to the personal phonebook, or not.
In this context the issue of privacy has to be raised. While PHOAF detects, retrieves and aggregates information freely available on the Web or the DNS, it can not guarantee the integrity or the legitimacy of the information being made public by third persons. It therefore is assured in no way, that Elizabeth wants or has agreed to have her contact details being published (by Alex and Bob, respectively) in the example from Figure 57 above. However, this is a matter of general FOAF policy and not within the scope of the PHOAF implementation.

8.4 ENUM/FOAF to RDF Transcoding

One of the main challenges on the road to the Semantic Web is how to make information already stored in databases worldwide available to Semantic Web applications. To achieve this it becomes necessary to annotate that data semantically. PHOAF can be used to do this

[98] In many cases the phonebook owner's contacts themselve will also know the phonebook owner, thus forming a mutual relationship in the resulting graph.

for information associated with a telephone number and being stored in ENUM and FOAF, respectively.

In a first step, PHOAF's ENUM data query and retrieval capabilities are utilised to check all phone numbers (for a specific phone directory, a whole number range or domain, or even all numbers worldwide) for associated ENUM domain delegations. In the case of an existing delegation with NAPTR RRs found, that data can be semantically annotated using RDF vocabularies available, including the SEMNUM RDF vocabulary developed in the course of the work for this thesis (see section 9 for a detailed introduction).

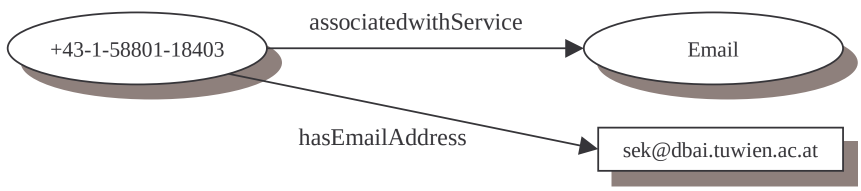

Figure 58: Graph showing association between telephone number and E-mail service

In a second step, PHOAF's FOAF data query and retrieval capabilities are utilised to further enhance the results later to be associated with a given telephone number. This time, the data found (i.e. FOAF data) already is coded in RDF, making no further transcoding necessary. The information found just has to be added to the semantically annotated content from step 1 (see above). The final result will be a telephone number associated with a variety of services and service identifiers, published on the Web, and annoted to be detected and accessed by Semantic Web tools and applications.

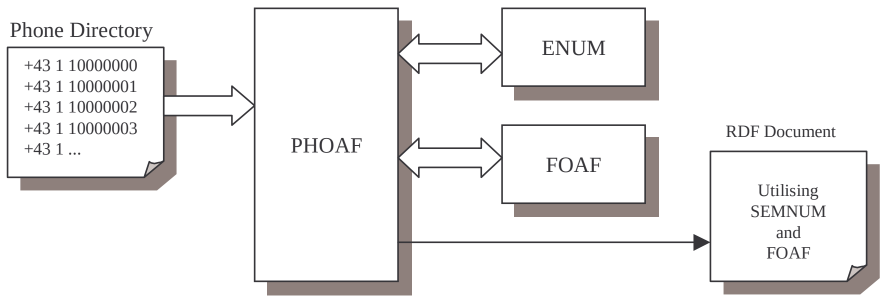

Figure 59: Basic concept of ENUM/FOAF to RDF Transcoding (creating an RDF phone directory)

As information in the original telephone number list[99], ENUM and FOAF can change over time the ENUM/FOAF to RDF Transcoding has to be repeated in an infinite loop (or in defined intervals, at least) in order to keep the RDF Phone Directory[100] up to date all time.

As outlined in section 8.1.2 the RDF phone directory could be implemented as a Web Service, responding to a qualified query with the full set of URIs associated with a phone number to be provided as input parameter. Contrary to the plain PHOAF Web Service introduced in section 8.1.2, the RDF Phonebook Web Service does not offer PHOAF functionality itself, but the

---

[99] It should be noted that this is not a usual phonebook associating the user's name with a telephone number, but just a list of telephone numbers. Although the numbers all are assigned to providers or individual users, the names of these persons are not necessarily retrievable from ENUM.

[100] Again, it should be noted that the output RDF Phone Directory is associating a telephone number with a set of URIs, but not necessarily with a user's name.

result of the ENUM/FOAF to RDF Transcoding. The PHOAF capabilities are only used in the background of the Web Service.

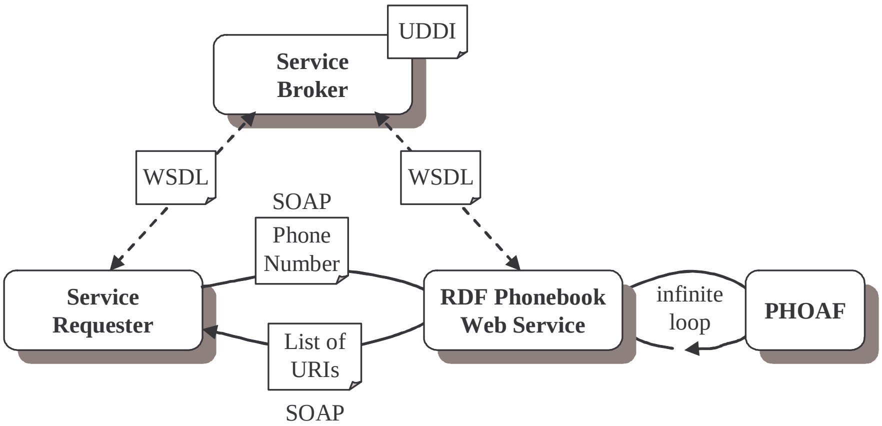

Figure 60: RDF Phone Directory Web Service

If the Service Requester just wants to know the corresponding URIs to a single telephone number it may be better to access the plain PHOAF (Web) Service as described in section 8.1.2. However, when a large amount of telephone numbers should be checked and possible semantic relationships are wanted to be detected, it is advantageous to have access to an RDF Phonebook already established as described in this section.

8.5 Trust on Corresponding Data in ENUM and FOAF

A further option presented in this thesis is to introduce a trust rating for corresponding data available both in ENUM and the Semantic Web's FOAF.

In the case of a comparison of URI's from ENUM and FOAF resulting in a match (e.g. when identical e-mail addresses are found in ENUM and FOAF), this gives an indication of a higher probability that the respective URI (e.g. the e-mail address) is correct. This is due to three facts:

- First, ENUM is used by service providers (cp. Infrastructure ENUM; see section 2.10) and end-users (cp. User ENUM; see section 2.10) for all-day communication purposes and therefore a rather high probability of the NAPTR RRs containing correct and up-to-date information can be expected. Otherwise ENUM-based services would not work properly, which is contradictory to the interest of the party maintaining the ENUM data (i.e. the called party or the provider) and paying for the respective ENUM service.
- Second, ENUM offers the opportunity to have the NAPTR RRs updated in rather short intervals depending on the respective application. So, the person (or an agent) responsible for the NAPTR RRs can alter the preferences within ENUM according to specific situations (e.g. no voice service URI at all from 9:00 to 10:30, voice URI from corporate extension from 10:30 to 12:30, voice URI from mobile phone number during lunch time, e-mail URI on weekends, and so on). Again, this means ENUM data must be correct and up-to-date for the associated applications to work.
- Third, ENUM and FOAF are both applications with a completely different background, focused on different usage scenarios and running on distinct platforms.

> Therefore it can be assumed in general that there is no inter-dependence between ENUM and FOAF data.

In conclusion, it becomes possible to introduce trust levels (or a trust indicator) for FOAF data, after a comparison with corresponding ENUM data. Should that comparison result in a match, a “thumbs up” indicator will be added to the FOAF data, otherwise a “neutral” or “thumbs down” indicator will be attached to the corresponding FOAF data.

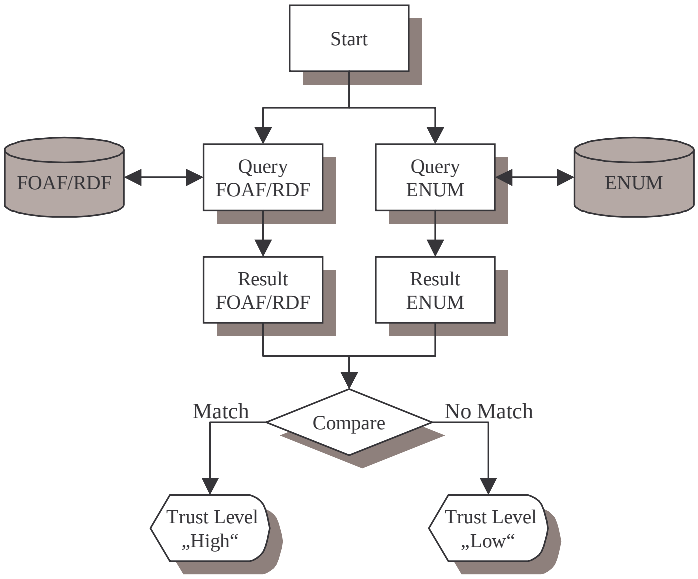

Figure 61: Flowchart for Trust Calculation utilising PHOAF

The PHOAF trust application needs a telephone number as input, performing subsequent ENUM and FOAF queries as explained in previous sections. As some data can be stored in both ENUM and FOAF, the results are checked for the occurrence of matching URIs. If e.g. the same E-Mail address or the same Web URL is to be found in both ENUM and FOAF, this will result in a higher trust level for the respective URI. In order to make the results found available for others (users, agents and applications), URI and trust indicator(s) can be added to the FOAF data by means of RDF/XML coding. Finally that data can get published on the Web again.

8.6 Unique Key to Web 2.0 Application Content

As explained in depth in section 4, Web 2.0 has become a growing phenomenon both in terms of available applications and usage figures. Web 2.0 applications are providing background information on people, as a growing number of examples show. Related research mainly deals with social network issues, e.g. relations between users or between tagged items. The application themed in this section deals with a different application aspect of social networks and folksonomies, centring the individual user and related content. When personal data is made publicly available on the Web by its owner, it can be accessed by others, assuming the location of that data is known or easy to detect. As Web 2.0 data is distributed widespread on the Web with no unique access point, it becomes nearly impossible to discover all of a single person’s Web 2.0 application data.

The introduction of ENUM to the Semantic Web allows the creation of a single key linking to semantically annotated content of a specific person as described previously (see section 6 and

[59]). In this section it is shown how that approach can be extended to detect a person's Web 2.0 content. By creating a new ENUMservice named “foaf” as well as extensions to the existing FOAF vocabulary, it becomes possible to discover all (or at least a large amount) of a person's Web 2.0 data by just knowing that person's telephone number. Furthermore, the data found can again be published on the Web using RDF/XML even making Web 2.0 data accessible for legacy Semantic Web tools. Building on top of the work of [13] the PHOAF prototype is extended in order to illustrate the approach.

8.6.1 ENUMservice “foaf”

At present time, the ENUM specifications from the relevant standardisation bodies do not include a specific ENUMservice enabling the immediate detection of a FOAF RDF file. Aiming to combine ENUM and FOAF, the basic PHOAF prototype design used the existing ENUMservice “web” combined with a subsequent educated guess method in order to search for a possible location of a FOAF RDF file (see section 7 and [60][62]). However, as a FOAF RDF file may be placed anywhere on a website and with auto-discovery links (see section 5.4) not always implemented, this did not provide a basis for further developments.

In order to overcome these restrictions the introduction of a new ENUMservice named “foaf” is proposed by the author. Using this new ENUMservice, querying a user's telephone number in ENUM will return a link to the exact URL of a corresponding FOAF RDF file. This gives users the opportunity to retrieve all sorts of identifiers giving access to a multitude of application data.

In short, the new ENUMservice “foaf” indicates that the resource identified by the associated URI is a source of FOAF data. The URI schemes to be used are http and https with the latter indicating that the resource can be fetched by using the Transport Layer Security (TLS) protocol or the Secure Socket Layer (SSL) protocol. The following main parameters are defined for the ENUMservice “foaf”.

- ENUMservice Name: "foaf"
- ENUMservice Type: "foaf"
- ENUMservice Subtype: N/A
- URI Schemes: "http", "https"

An ENUM entry referring to a FOAF RDF file could look like the following code fragment (see Listing 16).

Listing 16: Example ENUM entry utilising the 8.6.1 ENUMservice “foaf”

```
$ORIGIN 1.0.1.1.1.1.5.5.5.0.8.7.3.4.e164.arpa.
@ IN NAPTR 100 10 "u" "E2U+foaf"
!^.*$!http://www.dbai.tuwien.ac.at/proj/semnum/user1/foaf.rdf!" .
```

Performing an ENUM query for the Austrian E.164 telephone number +43780555111101 will result in a referral to the Web URL http://www.dbai.tuwien.ac.at/proj/semnum/user1/foaf.rdf indicating that a FOAF RDF file associated with that telephone number can be accessed there.

According to the relevant specifications in [23] a proposal for IANA registration of a new ENUMservice “foaf” has been submitted by the author to the IETF as an Internet-Draft [63] in February 2006 (see Annex 1). In addition, that draft was presented at the 65th IETF meeting in Dallas in March 2006. Currently the draft is (in its second version) at the IETF for discussion. Until acceptance from the IETF ENUM Working Group and final registration with IANA the new ENUMservice is used following the rules for experimental, non-registered ENUMservices being named “X-foaf” for this interim period.

8.6.2 FOAF Properties for Web 2.0

As Web 2.0 applications do not have a matching FOAF property, new properties have to be introduced. FOAF is an open-community project and RDF is designed to allow for easy integration of other ontologies. Therefore, the creation of Web 2.0 specific terms is possible, in general. The following new FOAF terms proposed by the author are examples showing the general approach of adding new properties describing URLs providing access to a person’s specific Web 2.0 data.

- Flickr Property foaf:flickr
- Del.icio.us Property foaf:delicious
- 43things Property foaf:fourtythreethings
- Plazes Property foaf:plazes

The example RDF/XML fragment from Listing 17 illustrates the usage of the new FOAF terms indicating that the person Kurt Reichinger has a flickr and a plazes account both characterised and accessible by means of a URL.

Listing 17: RDF/XML fragment with new FOAF properties

```
<foaf:Person>
   <foaf:name>Kurt Reichinger</foaf:name>
   <foaf:plazes rdf:resource="http://beta.plazes.com/user/kurt_r/"/>
   <foaf:flickr rdf:resource="http://www.flickr.com/photos/kurt_r/"/>
</foaf:Person>
```

It remains for further study (involving the FOAF community) which terms to include, how to name those terms and to which classes they should finally belong.

8.6.3 Extending PHOAF for Web 2.0 Data Detection and Fetching

To illustrate the advantages of integrating ENUM, FOAF and Web 2.0 the PHOAF prototype introduced in section 7 is extended. The main improvements concern detecting the location of FOAF RDF data using the new “foaf” ENUMservice and parsing the FOAF file found for Web 2.0 data available or being requested by an application.

The extended PHOAF prototype implementation needs a telephone number in the international format as input parameter. This input can be made manually (as in the demo setup) or by an agent importing the telephone number from a VoIP or E-mail client, for instance. Alternatively, PHOAF can be implemented as a Web Service (cp. Figure 53). Upon entering a phone number the following steps are performed by PHOAF:

- Looking up the ENUM DNS database
- Retrieving ENUM NAPTR data from the ENUM DNS database
- Detecting the location of FOAF RDF data using the new “foaf” ENUMservice
- Parsing the FOAF file for Web 2.0 specific FOAF RDF data
- Presenting the aggregated results from ENUM and FOAF

Depending on the application initiating the query, it is a further option to directly fetch data from distributed Web 2.0 application databases in an intermediary step. This could be e.g. a person's current location from plazes or favourite pictures from flickr, used for immediate presentation or further processing.

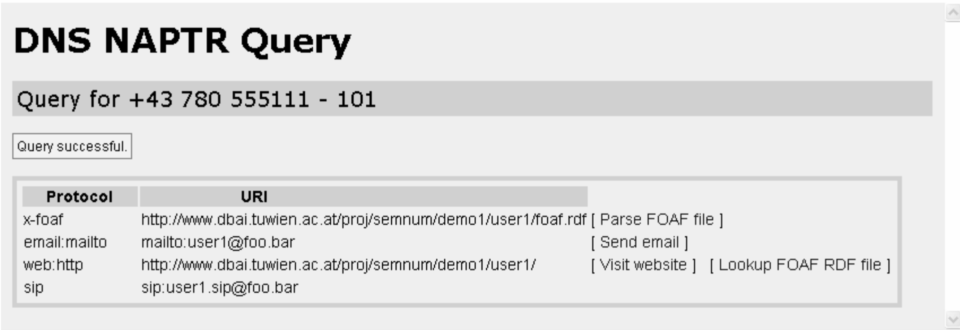

Figure 62: Screenshot of ENUM query results using the extended PHOAF prototype

Figure 62 shows the result of the ENUM query retrieving the associated DNS NAPTR resource records. In the example four values are returned: a FOAF file URI (using the new ENUMservice "foaf"[101] introduced in section 8.6.1), an e-mail address, a Web URL and a SIP AoR. Options for further action are listed next to the URIs, e.g. opening the e-mail client for sending e-mail or starting the Web browser for visiting a Web site.

Finally, Figure 63 shows the output screen presenting the FOAF RDF data fetched. That data is collected using the PHOAF parsing capabilities directly accessing the FOAF file URI retrieved from ENUM.

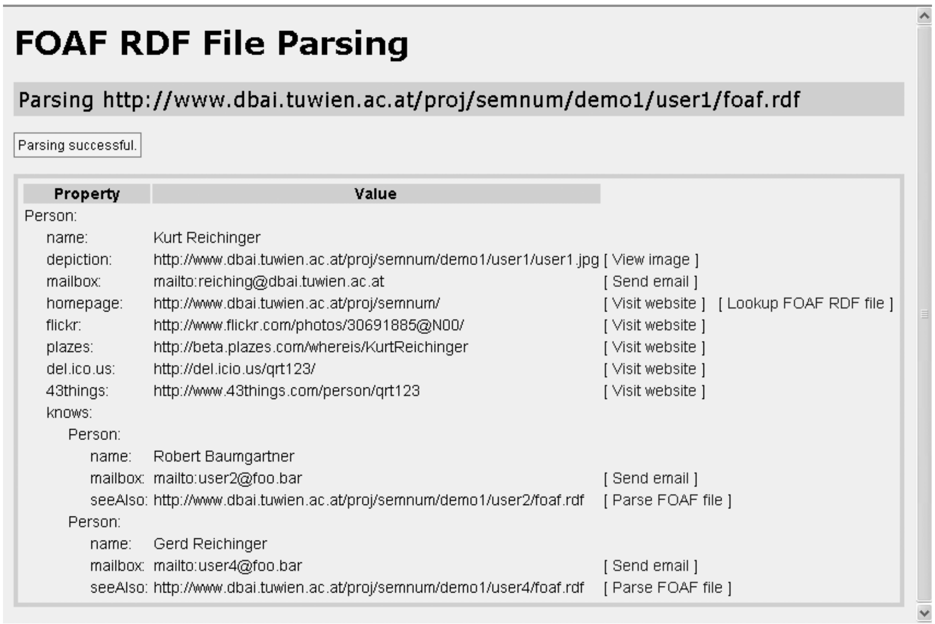

Figure 63: Screenshot of FOAF RDF parsing results using the extended PHOAF prototype

[101] As described in section 8.6.1 "x-foaf" is used until "foaf" is an officially registered by IANA.

The following data is presented: the queried person’s name, image URI, e-mail address, homepage URL as well as URLs for accessing that person’s Web 2.0 content at flickr, plazes, del.icio.us and 43things (using the new FOAF properties introduced in section 6). In addition, two other persons known by the queried person are listed, indicated to have their own FOAF files announced.

Data found in ENUM and FOAF can easily be aggregated, evaluated and combined with other data. Taking advantage of RDF and associated ontologies reasoning [2] becomes a possibility. This enables intelligent agents to prepare a complex enhanced personal information fact sheet on a queried person (or organisation) or to use it for other applications [60]. That aggregated data as well as data from other sources can be published again using RDF/XML, making the data accessible for legacy Semantic Web tools [19] and simply adding to the amount of semantically enhanced data available on the Web.

9 The SEMNUM RDF Vocabulary

SEMNUM is a vocabulary created by the author for describing information about (communication) identifiers stored in ENUM. All these identifiers are associated with a single E.164 telephone number (see section 2.8). As ENUM associates a telephone number with a person having the right to use this number, all the identifiers as a consequence are associated with that person as well. The vocabulary is designed to be compatible with RDF formats such as FOAF. It contains properties and classes structuring the information to be found through an ENUM query. The vocabulary is called SEMNUM as an abbreviation of "Semantic Numbers".

The purpose of this new RDF vocabulary is to have all terms contained in ENUM described in a single vocabulary. The translation of information found in ENUM into RDF terms, enables users or (more precisely) agents to further process the data found and to create new documents based on RDF. As data in ENUM originally is not stored in RDF format; this translation has to be performed in a second step after retrieving the information by means of an ENUM query.

SEMNUM is rather a simple vocabulary than an ontology. As described in [58] the difference between a vocabulary and ontology mainly lies in the fact, that an ontology provides additional constraints that increase the accuracy of implementations of a given vocabulary. For the course of the work on this thesis and the related examples, this was not deemed necessary for SEMNUM.

An example incorporating the SEMNUM feature is an agent crawling the Web for ENUM and RDF/FOAF data (see section 8.1 and 8.6) in order to prepare a new fact sheet on every person found. This person data could be stored in a new "verified fact sheet" using vocabularies like FOAF, BIO [18], VISIT [25], or SEMNUM. Such verified fact sheets could be stored on local machines building up an enhanced private directory. On a large scale, on the basis of a verified fact sheet database grown large enough, a (Web) service could be created, offering this additional information. Privacy is not deemed to be a critical issue for this application of ENUM and the SEMNUM vocabulary, as all the information collected is made freely available on the DNS or the Web by the respective owner.

9.1 Standards and RDF

The SEMNUM vocabulary does not define a new standard, but is meant to contribute another vocabulary to the Semantic Web. The specification of this vocabulary uses the conventions of the W3C's Resource Description Framework (RDF). As such SEMNUM adopts a syntax (using XML), a data model (RDF graphs) and a mathematically grounded definition for the rules that underpin the SEMNUM design.

SEMNUM is an application of the Resource Description Framework (RDF), because the complexity of the subject described – people and their various connections – is best addressed by a combination of different RDF vocabularies. SEMNUM by definition only deals with information defined for ENUM (see section 2.8), but with RDF being used SEMNUM is well prepared to be extended by several other RDF vocabularies dealing with the description of people, communication services and their relations. Due to the RDF nature of SEMNUM its classes and properties can also be incorporated in other RDF documents.

9.2 SEMNUM at a Glance

Consequently, the following classes and properties are defined in SEMNUM. They are explained in detail in sections 9.4 and 9.5.

Classes: Service | Email | Voice | Fax | Ifax | Video | Sms | Ems | Mms | Web | Announcement | Location | Key | Instantmessaging

Properties: mbox | homepage | securehomepage | file | database | phone | sipphone | securesipphone | h323phone | infotext

9.3 Namespace

The XML namespace for this vocabulary is http://www.dbai.tuwien.ac.at/proj/semnum/0.1/# and the recommended prefix is semnum. In RDF/XML documents the following attribute should be included in the rdf:RDF element.

```
xmlns:semnum="http://www.dbai.tuwien.ac.at/proj/semnum/0.1/"
```

This namespace definition is reflected in the description of SEMNUM properties and classes in the following sections.

Documents using the RDF SEMNUM vocabulary are also likely to use terms from other vocabularies, e.g. the Dublin Core (DC) and Friend-Of-A-Friend (FOAF) namespaces. Listing 18 shows a typical RDF/XML code snippet utilising several namespaces, as RDF, RDFS, DC, FOAF and SEMNUM.

Listing 18: RDF/XML code fragment showing use of XML namespaces

```
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:semnum="http://www.dbai.tuwien.ac.at/proj/semnum/0.1/">
<!—-
      more code here-->

</rdf:RDF>
```

Regarding vocabularies/ontologies and namespace considerations, it should be referred to Tim Berners-Lee, W3C director, who demanded to bring “the Web back into the Semantic Web” in a keynote speech at the ISWC’2005 conference in Ireland. The problem addressed is people creating vocabularies or ontologies (using URIs and namespaces, as explained above) without uploading it to Web servers. However, it is necessary to make vocabularies and ontologies publicly available in order to allow global inference. It is therefore essential that even namespaces used in research projects, as the SEMNUM vocabulary, are uploaded to the Web.

9.4 Vocabulary Classes

This section defines classes used within the SEMNUM vocabulary. There is one class defining “Service”; and several subclasses of “Service” defining more specific services all to be found in ENUM (see left-hand screenshot in Figure 64). Examples of these subclasses include “Announcement”, “Email”, “Sms”, “Fax”, “Voice”, “Video”, “Web” and many more.

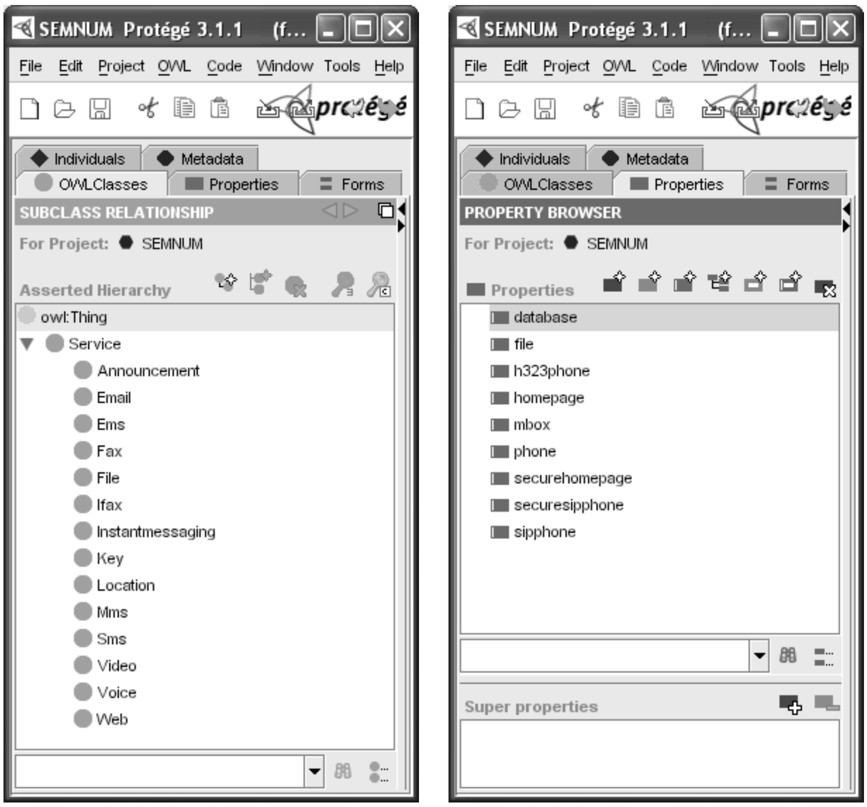

Figure 64: SEMNUM classes and properties (screenshots from Protégé editor[102])

9.4.1 Class: semnum:Service

URI `http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`
Label Service
Definition The class `semnum:Service` denotes a (communication) service of some kind. Subclasses of class `semnum:Service` include e.g. E-Mail (class `semnum:Email`), Web (class `semnum:Web`), Phone (class `semnum:Phone`), Facsimile (class `semnum:Fax`) and SMS (class `semnum:Fax`). The subclasses introduced reflect the respective ENUMservices agreed by IANA and used with the NAPTR Resource

[102] Cp. http://protege.stanford.edu, 17.08.2006.

Records. When IANA registers a new ENUMservice, the SEMNUM vocabulary will be updated accordingly.

9.4.2 Class: semnum:Email

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Email`
Label	Email
Definition	The class `semnum:Email` denotes a service providing E-Mail.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.3 Class: semnum:Voice

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Voice`
Label	Voice
Definition	The class `semnum:Voice` denotes a service providing voice communication, e.g. based on the Plain Old Telephone Service (POTS), the Integrated Services Digital Network (ISDN), the Global System for Mobile Communications (GSM) or Voice over IP (VoIP) Services.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.4 Class: semnum:Fax

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Fax`
Label	Fax
Definition	The class `semnum:Fax` denotes a service providing classical facsimile communications, e.g. based on the Plain Old Telephone Service (POTS), the Integrated Services Digital Network (ISDN), the Global System for Mobile Communications (GSM) or Internet Services.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.5 Class: semnum:Ifax

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Ifax`
Label	Ifax
Definition	The class `semnum:Ifax` denotes a service providing facsimile services according to IETF's RFC 2305 [69] or RFC 2532 [42]. It therefore describes a service which allows facsimile documents to be sent to e-mail addresses.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.6 Class: semnum:Video

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Video`
Label	Video
Definition	The class `semnum:Video` denotes a service providing video communications, e.g. the Integrated Services Digital Network (ISDN), the Universal Mobile Telecommunications Service (UMTS) or Voice over IP (VoIP) Services.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.7 Class: semnum:Sms

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Sms`
Label	Sms

Definition	The class `semnum:Sms` denotes a service providing the sending and receiving of Short Messages (Short Message Service; SMS), originally known from the mobile networks sector, but now also provided on fixed networks as well as the Internet.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.8 Class: semnum:Ems

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Ems`
Label	Ems
Definition	The class `semnum:Ems` denotes a service providing the sending and receiving of Enhanced Messages (Enhanced Message Service; EMS), originally known from the mobile networks sector, but now also provided on fixed networks as well as the Internet. EMS enabled devices can send and receive messages that have special text formatting (such as bold or italicised), animations, pictures, icons, sound effects and special ring tones.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.9 Class: semnum:Mms

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Mms`
Label	Mms
Definition	The class `semnum:Mms` denotes a service providing the sending and receiving of Multimedia Messages (Multimedia Message Service; MMS), supported by mobile networks and the Internet. MMS is a store-and-forward method of transmitting graphics, video clips, sound files and short text messages over wireless networks. MMS also supports e-mail addressing, meaning that the device can send e-mails directly to an e-mail address.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.10 Class: semnum:Web

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Web`
Label	Web
Definition	The class `semnum:Web` denotes a service providing access to a public document on the Web.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.11 Class: semnum:File

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/File`
Label	File
Definition	The class `semnum:File` denotes a service providing access to an addressed document (file) or file listing.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.12 Class: semnum:Announcement

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Announcement`
Label	Announcement
Definition	The class `semnum:Announcement` denotes a service providing access to an announcement, i.e. an instant information display. An announcement can be a text as well as a sound message, which can be specifically important

	for people with visual or hearing impairments. Typically announcements are intended to trigger automatic execution. As this involves significant risks due to unsolicited onward actions this service is for further study during the ENUM trials.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.13 Class: semnum:Location

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Location`
Label	Location
Definition	The class `semnum:Location` denotes a service providing access to some sort source for location information.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.14 Class: semnum:Key

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Key`
Label	Key
Definition	The class `semnum:Key` denotes a service providing access to some sort of source for public key information.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.4.15 Class: semnum:Instantmessaging

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Instantmessaging`
Label	Instantmessaging
Definition	The class `semnum:Instantmessaging` denotes a service providing access to some sort of Instant Messaging Service.
Sub Class Of	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/Service`

9.5 Vocabulary Properties

This section defines properties used within the SEMNUM vocabulary (see right-hand screenshot in Figure 64). It has to be noted that a SEMNUM property can belong to more than one domain, e.g. the property `semnum:mbox` is in domain of `semnum:Email`, `semnum:Ifax`, `semnum:Sms`, `semnum:Ems` and `semnum:Mms` (see Figure 65).

9.5.1 Property: semnum:mbox

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/mbox`
Label	mbox
Definition	A personal mailbox on the Internet associated with exactly one owner. Typically such a mailbox is identified by using the mailto: URI scheme defined in IETF's RFC 2368 [31]. It has to be noted that there are mailboxes which are not the `semnum:mbox` of anyone and that one (a thing in the OWL definition; inluding person) can have multiple `semnum:mbox`. Furthermore, the property `semnum:mbox` is identical to the FOAF property `foaf:mbox`.
Domain	`http://www.w3.org/2002/07/owl#Thing`
Range	`foaf:Agent`

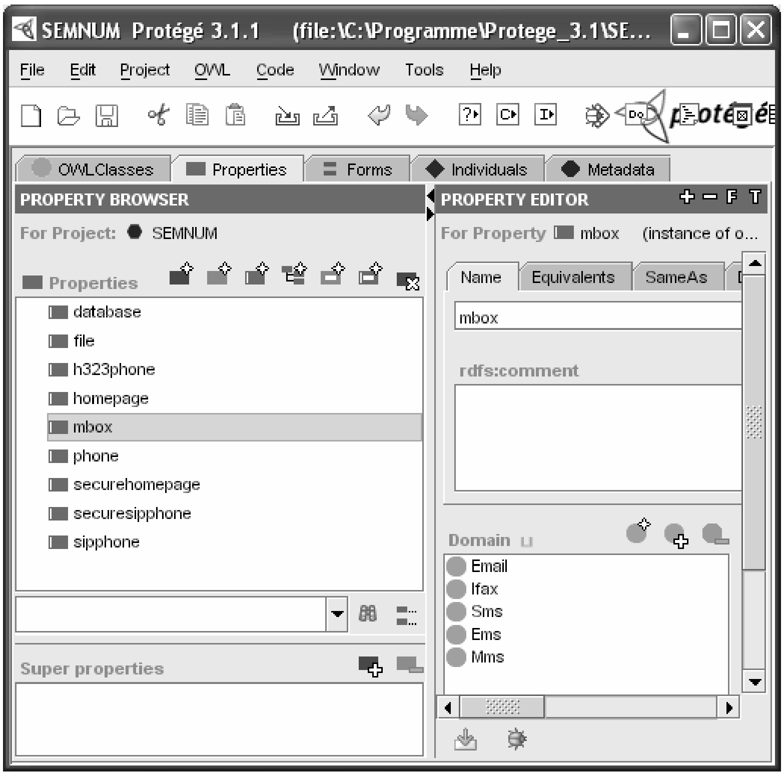

Figure 65: SEMNUM property `semnum:mbox` belonging to several domains (screenshot from Protégé editor)

9.5.2 Property: semnum:homepage

URI `http://www.dbai.tuwien.ac.at/proj/semnum/0.1/homepage`
Label homepage
Definition ENUM allows one (a thing in the OWL definition; including person) to have multiple homepages, but SEMNUM constrains `semnum:homepage` so that there can be only one person that has any particular homepage, i.e. not allowing two persons to have the same homepage. It remains for further study how to deal with e.g. family homepages. A homepage is usually controlled, edited or published by the thing whose homepage it is; as such one might look to a homepage for information on its owner from its owner. This works for people, companies, organisations etc.
A homepage in this sense is a public Web document, typically (but not necessarily) available in HTML format. In most cases such a document is retrievable by using the Hyper Text Transfer Protocol (http:// URI scheme) as defined in IETF's RFC 2616 [24]. Furthermore, the property `semnum:homepage` is identical to the FOAF property `foaf:homepage`.
Domain `http://www.w3.org/2002/07/owl#Thing`
Range `foaf:Document`

9.5.3 Property: semnum:securehomepage

URI `http://www.dbai.tuwien.ac.at/proj/semnum/0.1/securehomepage`
Label securehomepage
Definition A secure homepage is a public web document, typically (but not necessarily) available in HTML format. In most cases such a document is retrievable by

	using the Hyper Text Transfer Protocol Secure (https:// URI scheme), which enables encrypted transactions between browser and server.
Domain	`http://www.w3.org/2002/07/owl#Thing`
Range	`foaf:Document`

9.5.4 Property: semnum:file

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/file`
Label	file
Definition	A file in the context of SEMNUM is a public Web document, retrievable by using the File Transfer Protocol (ftp:// URI scheme), which enables encrypted transactions between browser and server.
Domain	`http://www.w3.org/2002/07/owl#Thing`
Range	http://www.w3.org/2000/01/rdf-schema#Literal

9.5.5 Property: semnum:database

URI	`http://www.dbai.tuwien.ac.at/proj/semnum/0.1/database`
Label	database
Definition	A database in the context of SEMNUM is a database, accessible by using the Lightweight Database Access Protocol (ldap) as specified in a series of IETF standard track RFCs as detailed in RFC 4510 [83]. LDAP enables transactions between client and database server for querying and modifying directory services running over TCP/IP.
Domain	`http://xmlns.com/foaf/0.1/Person`
Range	`http://www.w3.org/2000/01/rdf-schema#Literal`

9.5.6 Property: semnum:phone

URI	`http://dbai.tuwien.ac.at/proj/semnum/0.1/phone`
Label	phone
Definition	A `semnum:phone` describes a fully qualified telephone number, typically described by using the tel: URI scheme as defined in IETF's RFC 3966 [65].
Domain	`http://xmlns.com/foaf/0.1/Person`
Range	`http://www.w3.org/2000/01/rdf-schema#Literal`

9.5.7 Property: semnum:sipphone

URI	`http://dbai.tuwien.ac.at/proj/semnum/0.1/sipphone`
Label	sipphone
Definition	A `semnum:sipphone` describes a Voice over IP account using the Session Initiation Protocol (SIP) as defined in IETF's RFC 3261 [64], typically described by using the sip: URI scheme.
Domain	`http://xmlns.com/foaf/0.1/Person`
Range	`http://www.w3.org/2000/01/rdf-schema#Literal`

9.5.8 Property: semnum:securesipphone

URI	`http://dbai.tuwien.ac.at/proj/semnum/0.1/securesipphone`
Label	securesipphone
Definition	A `semnum:securesipphone` describes a Voice over IP account using the Session Initiation Protocol Secure as defined in IETF's RFC 3261 [64], typically described by using the sips: URI scheme.

Domain	`http://xmlns.com/foaf/0.1/Person`
Range	`http://www.w3.org/2000/01/rdf-schema#Literal`

9.5.9 Property: semnum:h323phone

URI	`http://dbai.tuwien.ac.at/proj/semnum/0.1/h323phone`
Label	h323phone
Definition	A `semnum:h323phone` describes a Voice over IP account using the Protocol defined in ITU-T Recommendation H.323, typically described by using the h323: URI scheme.
Domain	`http://xmlns.com/foaf/0.1/Person`
Range	`http://www.w3.org/2000/01/rdf-schema#Literal`

9.6 SEMNUM Example

9.6.1 Using SEMNUM with FOAF

The SEMNUM vocabulary simply slots into standard FOAF documents as shown in Listing 19. The code fragment starts with a declaration of being an RDF document with the according namespaces for RDF, RDFS, FOAF and SEMNUM all explicitly stated. Furthermore, it is declared that the document is to be interpreted as a FOAF RDF document describing a person. This is followed by a description of the person Kurt Reichinger utilising some FOAF properties (name, title, givenname, family_name, mbox_sha1sum, homepage, phone, workplaceHomepage, workInfoHomepage, schoolHomepage, knows) and some SEMNUM properties (mbox, homepage, phone, sms and infotext).

Listing 19: RDF code fragment utilising SEMNUM classes and properties

```
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:semnum="http://semnum.org/vocab/semnum/0.1/">
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:nodeID="me"/>
  <foaf:primaryTopic rdf:nodeID="me"/>
  </foaf:PersonalProfileDocument>
<foaf:Person rdf:nodeID="me">
<foaf:name>Kurt Reichinger</foaf:name>
<foaf:title>Dipl.Ing.</foaf:title>
<foaf:givenname>Kurt</foaf:givenname>
<foaf:family_name>Reichinger</foaf:family_name>
<foaf:mbox_sha1sum>1f043d44f3c8f272cee5443a666751b36b68faf0</foaf:mbox_sha1sum>
<foaf:homepage rdf:resource="members.chello.at/reichinger/"/>
<foaf:phone rdf:resource="tel:+43-1-58058-306"/>
<foaf:workplaceHomepage rdf:resource="www.rtr.at"/>
<foaf:workInfoHomepage rdf:resource="www.rtr.at/enum/"/>
<foaf:schoolHomepage rdf:resource="www.univie.ac.at"/>
<foaf:knows>
<foaf:Person>
<foaf:name>Gerd Reichinger</foaf:name>
<foaf:mbox_sha1sum>dcea15119cdefc1c93a0363ef76bbef0656525f3</foaf:mbox_sha1sum></fo
af:Person></foaf:knows>

<semnum:Service>
<semnum:Email>
<semnum:mbox rdf:resource="mailto:kurt.reichinger@chello.at"/>
</semnum:Email>
```

```
</semnum:Service>

<semnum:Service>
<semnum:Email>
<semnum:mbox rdf:resource="mailto:kurt.reichinger@rtr.at"/>
</semnum:Email>
</semnum:Service>

<semnum:Service>
<semnum:Email>
<semnum:mbox rdf:resource="mailto:kurt.reichinger@rtr.at"/>
</semnum:Email>
</semnum:Service>

<semnum:Service>
<semnum:Web>
<semnum:homepage rdf:resource="http://semnum.org/kurt/"/>
</semnum:Web>
</semnum:Service>

<semnum:Service>
<semnum:Phone>
<semnum:phone rdf:resource="tel:+43-1-58058-306"/>
</semnum:Phone>
</semnum:Service>

<semnum:Service>
<semnum:Phone>
<semnum:phone rdf:resource="tel:+43-664-3504516"/>
</semnum:Phone>
</semnum:Service>

<semnum:Service>
<semnum:Sms>
<semnum:phone rdf:resource="tel:+43-664-3504516"/>
</semnum:Sms>
</semnum:Service>

</foaf:Person>
</rdf:RDF>
```

9.6.2 SEMNUM Graph

FOAF documents generally can be described by means of a graph. Therefore the FOAF file listed in section 9.6.1 is presented in graphical form using IsaViz.[103] IsaViz is a visual environment provided by the W3C RDF developers group for browsing and authoring RDF models represented as graphs. According to the W3C IsaViz description the tool basically features the following:

- 2.5D user interface allowing smooth zooming and navigation in the graph
- Creation and editing of graphs by drawing ellipses, boxes and arcs
- RDF/XML, Notation 3 and N-Triple import
- RDF/XML, Notation 3 and N-Triple export, but also SVG and PNG export

[103] Cp. http://www.w3.org/2001/11/IsaViz/, 18.08.2006.

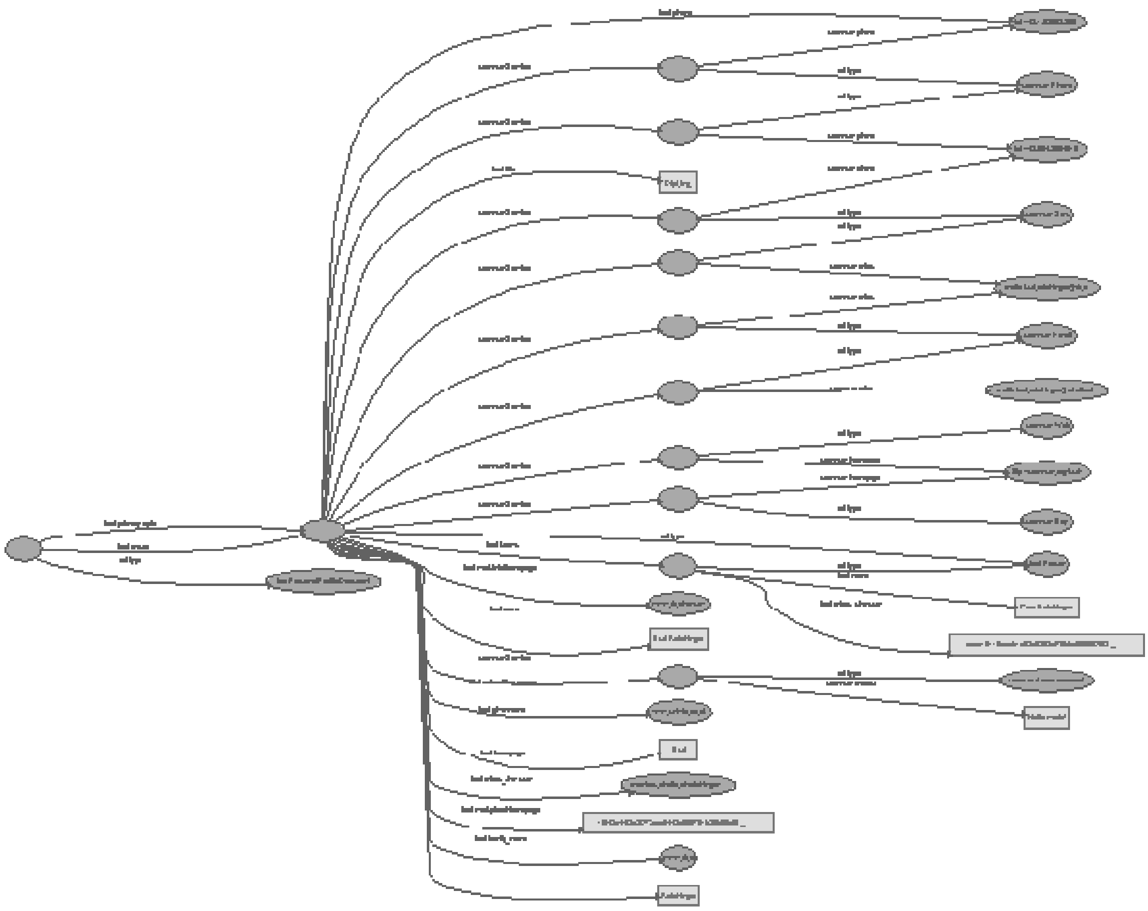

Figure 66: FOAF file as a graph

As the graph shown in Figure 66 is impossible to print on A4 sheets in readable size, it is simplified to some major relations (depicted in Figure 67) making it easier to describe.

On the left-hand side the graph in Figure 67 explains that the document described is a Personal Profile Document and that it has a Maker, i.e. a person that created the document. That person is further described by its name (as Literal, indicated graphically by a square shape), and its workplace homepage (as Resource, indicated graphically by an oval shape). Furthermore the document maker is described by some services and associated identifiers that could be used for contacting that person. First, there is a mailto: URI to be used for E-Mail communication; and second, there is a telephone number to be used for SMS communication.

It should be pointed out that the document maker in the example graph from Figure 67 is described using both FOAF properties (i.e. foaf:name and foaf:workplacehomepage) and SEMNUM properties (i.e. semnum:mbox and semnum:phone) in the same RDF document. As mentioned introductorily, this becomes possible because of both vocabularies being based on RDF XML standards, and therefore being perfectly suited for integration and cooperation with each other.

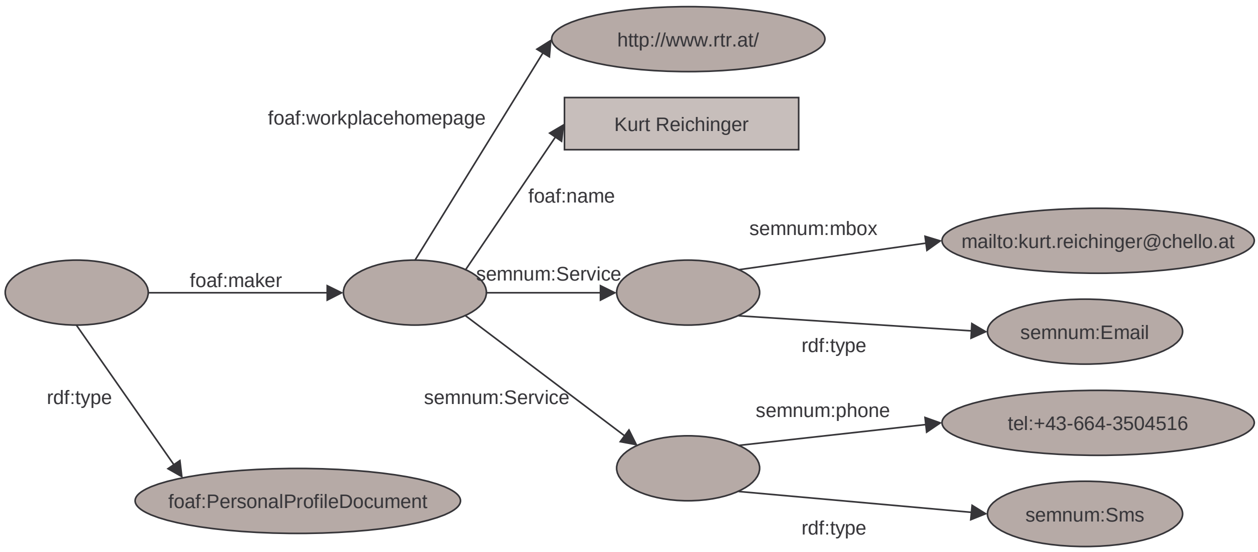

Figure 67: Simplified version of FOAF file graph from Figure 66

10 Conclusions

Telephone Number Mapping (ENUM) and the Semantic Web both represent highly innovative and seminal developments in their respective areas; that is telecommunication on the one hand, and the World Wide Web on the other. While both ENUM and the Semantic Web are expected to massively impact their respective areas in the near- to medium-term future, the combination of the two neither has been looked at nor been analysed so far.

The first major part of this thesis introduced both areas, ENUM and the Semantic Web, from a formal and theoretic point of view figuring out possible synergies to be expected when combining the two respective concepts.
First, the thesis revealed that ENUM means a big step forward towards increased convergence between the world of traditional telecommunication and the Internet. ENUM connects a telephone number (and its owner) to a multitude of communication (and other) identifiers belonging to that person (number) by simply mapping the telephone number to an Internet domain name, i.e. pointing from a telephone number to a Uniform Resource Identifier. ENUM itself may be nothing more than a simple mapping algorithm, but the range of opportunities for the creation of new services and applications is vast.
Second, the considerations showed that the Semantic Web can be regarded as a substantial, far-reaching and visionary development of today's World Wide Web towards an increasingly machine-processable Web, with some of the recent Web 2.0 activities deemed to be an important intermediary step in that direction. The Semantic Web is built around a whole new concept of annotating and sharing documents on the Web reflected in the Semantic Web layer cake. Following that layered approach, identifiers, documents and their structure, statements, schemas and ontologies, query languages, logic, proof and trust are all described, aiming to add explicit meaning to the content of documents, thus enabling machines (agents) to better understand the information available and to enable them to draw conclusions.
Based on these basic introductory sections, the similarities between the two concepts were looked at and found to be manifold: Both ENUM and the Semantic Web conceptually are focussing on a highly automated inter-machine communication rather than a human-machine communication; both rely heavily on URIs accessible and retrievable on the Web; both introduce meaning to content, however with the Semantic Web having a much stronger focus on that issue than ENUM; while the Semantic Web works with taxonomies, ontologies and schemas, this is not used in ENUM today, however it is shown in this thesis that the creation of ENUM RDF vocabularies can bring significant advantages; logic and proof are domains of the Semantic Web, with ENUM being able to provide valuable input for proof verifying and reasoning; and finally trust, that again is an issue in both areas.

The second major part circled around the PHOAF prototype developed during the course of the thesis. Following an engineer's hand-on approach, the PHOAF prototype was used to implement first application examples all bridging the gap between ENUM and the Semantic Web. As PHOAF is entirely based on standard Java classes it is prepared for both easy

implementation and integration in other applications. The focus of this application-oriented part of the thesis was on identifying possible synergies, advantages and remaining problems.
Until now, voice over IP (VoIP) is the main use case for ENUM implementations and the first two examples presented in this thesis followed that path. In the first application, the combination of ENUM and the Semantic Web's Friend-of-A-Friend (FOAF) project enabled the creation of a called (or calling) party information to be presented on the calling (or called) party's terminal; the second example used the ENUM/FOAF combination in the context of company structures and employee working relationships in order to calculate an optimised redirection of incoming calls when the called party is not available. Both examples showed that the introduction of Semantic Web technology has the potential for immediate impact on every-day communication services.
Another group of applications presented in this thesis is connected to the popular issue of social networking, in one or the other way. First, the combination of ENUM and FOAF was used for spicing up an ordinary personal phonebook as used in mobile phones or PDAs with additional information. This led to the creation of a phonebook contact network consisting of known and formerly unknown contacts and their relations. A second example utilised an upgraded version of the PHOAF prototype to access a single person's Web 2.0 application content distributed on the WWW. In the context of this application a new ENUMservice was introduced, and the FOAF vocabulary was extended in order to support the application specific requirements. A third example regarded a simple ENUM/FOAF to RDF transcoding tool for translating information available in ENUM into information accessible by Semantic Web tools. To support that purpose, a new RDF ENUM vocabulary (SEMNUM) created as a part of this thesis was introduced.
Finally, an application example residing on the trust layer was presented, introducing a trust indicator on corresponding data in ENUM and FOAF. This enabled one to compare data from two distinct databases and to draw a conclusion regarding trustworthiness.

Based on the results of both the theoretic and the application-oriented part of this thesis it can be concluded that the proposed convergence of ENUM and the Semantic Web definitely has the potential for significant impact in both areas. With the concepts being mature and the technology available, further engagement in that research topic is valued as both reasonable and promising. The examples presented illustrate the range of possible applications, which are easy to implement and well suited for use in every-day communication scenarios.

References

[1] Albitz, P. and Liu, C.: DNS and BIND, Fourth Edition, O'Reilly, April 2001.

[2] Antoniou, G. and Van Harmelen, F.: A Semantic Web Primer, The MIT Press, 2004.

[3] Antoniou, G., Baldoni, M., Baroglio, C., Baumgartner, R., Bry, F., Eiter, T., Henze, N., Herzog, M., May, W., Patti, V., Schindlauer, R., Tompits, H., and Schaffert, S.: Reasoning Methods for Personalization on the Semantic Web, Annals of Mathematics, Computing & Teleinformatics 2 (1), 2004.

[4] Bailey, J., Bry, F., Furche, T., and Schaffert, S.: Web and Semantic Web Query Languages: A Survey. In: Reasoning Web, First International Summer School 2005, Malta, July 2005.

[5] Baldoni, M., Baroglio, C., Henze, H.: Personalization for the Semantic Web. Reasoning Web, First International Summer School, LNCS, 2005.

[6] Baumgartner, R., Flesca, S. and Gottlob, G.: Declarative Information Extraction, Web Crawling, and Recursive Wrapping with Lixto, Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning LPNMR'2001, September 2001.

[7] Baydoun, J.: Regelsprachen, Seminar Praktische Modellierung SS 2005, FU-Berlin, http://www.inf.fu-berlin.de/inst/ag-nbi/lehre/05/S_MOD/Regelsprachen_180505.pdf, Mai 2005.

[8] Berners-Lee T., Hendler, J., Lassila, O., The Semantic Web, Scientific American, May 2001.

[9] Berners-Lee, T., Fielding, R. and Frystyk, H.: Hypertext Transfer Protocol -- HTTP/1.0, RFC 1945, IETF, May 1996.

[10] Berners-Lee, T., Fielding, R., and Masinter, L.: "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, August 1998.

[11] Berners-Lee, T.: Notation 3 Specification, World Wide Web Consortium, W3C, http://www.w3.org/DesignIssues/Notation3.html, 1998.

[12] Bizer, C.: Bausteine einer Vertrauens- und Sicherheitsinfrastruktur für das Semantic Web, http://www.wiwiss.fu-berlin.de/suhl/bizer/SWTSGuide/semtrust.ppt, Workshop on XML-Technologies for Middleware, Middleware for XML Applications[104] XMIDX 2003, February 2003.

[13] Brickley D., Miller L.: FOAF Vocabulary Specification, http://xmlns.com/foaf/0.1/, July 2005.

[104] Original title in German: „Workshop XML-Technologien für Middleware, Middleware für XML-Anwendungen".

[14] Callas, J., Donnerhacke, L., Finney, H., and Thayer R.: OpenPGP Message Format, RFC 2440, IETF, November 1998.

[15] Ceri, S., Gottlob, G. and Tanca, L.: Logic Programming and Databases". Springer-Verlag, Berlin, 1990.

[16] Daconta, M., Obrst, L., Smith, K.: The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management, Wiley, ISBN: 0-471-43257-1, 2003.

[17] Davies, J., Fensel, D., van Harmelen, F.: Towards the Semantic Web, Ontology-driven Knowledge Management, Wiley, ISBN 0470 84867 7, 2003.

[18] Davis, I. And Galbraith, D.: BIO: A vocabulary for biographical information, http://purl.org/vocab/bio/0.1/bio-vocab-20040305, 2004.

[19] Decker, S., Van Harmelen, F., Broekstra, J., Erdmann, M., Fensel, D., Horrocks, I., Klein, M. and Melnik, S.: The Semantic Web: The Roles of XML and RDF, IEEE Internet Computing, 2000.

[20] Deutsch, M: The Resolution of Conflict, Yale University Press, 1973.

[21] Entrust: Web Services Trust and XML Security Standards, Version 1.0, April 2001.

[22] European Telecommunications Standardization Institute: Minimum Requirements for Interoperability of ENUM implementations, TS 102 172 V1.2.1, ETSI, May 2005.

[23] Faltstrom, P., Mealling, M.: The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM), IETF RFC 3761, April 2004.

[24] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol -- HTTP/1.1, RFC 2616, IETF, June 1999.

[25] Frederiksen, M.: VISIT, A set of terms for indicating visited countries and states, http://www.wasab.dk/morten/2004/07/visit, July 2004.

[26] Geroimenko, V.: Dictionary of XML Technologies and the Semantic Web, Springer, ISBN 1-85233-768-0, 2004.

[27] Golbeck J., Hendler J., Parsia B.: Trust Networks on the Semantic Web, World Wide Web Conf. WWW2003, May 2003.

[28] Golbeck, J. and Hendler, J.: Reputation Network Analysis for Email Filtering, Proceedings of the Conference on Email and Anti-Spam (CEAS), July 2004.

[29] Golbeck, J., Sharing and Using Links in Social Networks, IEEE Intelligent Systems 20 (1), 2005.

[30] Grandison, T. and Sloman, M.: A Survey of Trust in Internet Applications IEEE Communications Surveys and Tutorials, Fourth Quarter 2000, January 2001.

[31] Hoffman, P., Masinter, L. and Zawinski, J.: The mailto URL scheme, RFC 2368, IETF, July 1998.

[32] Huston G., Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa"), RFC 3172, September 2001.

[33] Huston, G.: ENUM – Mapping the E.164 Number Space into the DNS, The Internet Protocol Journal, Volume 5, Number 2, June 2002.

[34] International Telecommunications Union, ITU-T Recommendation E.164, The International Public Telecommunication Numbering Plan, incl. Supplements 1 – 4, approved May 1997.

[35] International Telecommunications Union, List of ITU-T Recommendation E.164 Assigned Country Codes, May 2005.

[36] Internet Architecture Board, Approval of Interim Procedures and Assurances of Stability Liaison Statement, IAB (ISOC), September 2002.

[37] Kifer, M., Lausen, G. and Wu, J.: Logical Foundations of Object Oriented and Frame Based Languages, Journal of the Association for Computing Machinery, May 1995.

[38] Leone, N., Pfeifer, G., Faber, W., Calimeri, F., Dell'Armi, T., Eiter, T., Gottlob, G., Ianni, G., Ielpa, G., Koch, C., Perri, S. and Polleres, A.: The DLV System. In Sergio Flesca, Sergio Greco, Giovambattista Ianni, and Nicola Leone, editors, Proceedings of the 8th European Conference on Logics in Artificial Intelligence JELIA, September 2002.

[39] Lind, S. and Pfautz, P: Infrastructure ENUM Requirements, draft-ietf-enum-infrastructure-enum-reqs-03, Internet Draft, IETF, August 2006.

[40] Livingood, J., Hoeneiesen, B, Mayrhofer, A.: Guide and Template for IANA Registrations of ENUMservices, draft-ietf-enum-enumservices-guide-01.txt, Internet Draft, IETF, June 2006.

[41] Livingood, J., Pfautz, P. and R. Stastny: The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application for Infrastructure ENUM, draft-ietf-enum-infrastructure-00, Internet Draft, IETF, April 2006.

[42] Masinter, L. and Wing, D.: Extended Facsimile Using Internet Mail, RFC 2532, IETF, March 1999.

[43] McLaughlin, B.: Java & XML, 2nd Edition, O'Reilly, Second Edition, ISBN: 0-596-00197-5, September 2001.

[44] Mealling, M.: Dynamic Delegation Discovery System (DDDS) Part 1: The Comprehensive DDDS, RFC 3401, IETF, October 2002.

[45] Mealling, M.: Dynamic Delegation Discovery System (DDDS) Part 2: The Algorithm, RFC 3402, IETF, October 2002.

[46] Mealling, M.: Dynamic Delegation Discovery System (DDDS) Part 3: The Domain Name System (DNS) Database, RFC 3403, IETF, October 2002.

[47] Mealling, M.: Dynamic Delegation Discovery System (DDDS) Part 4: The Uniform Resource Identifiers (URI) Resolution Application, RFC 3404, IETF, October 2002.

[48] Mealling, M.: Dynamic Delegation Discovery System (DDDS) Part 5: URI.ARPA Assignment Procedures, RFC 3405, IETF, October 2002.

[49] Mika P., Flink: Semantic Web Technology for the Extraction and Analysis of Social Networks, Journal of Web Semantics 3 (2), 2005.

[50] Mika P.: Ontologies are us: A unified model of social networks and semantics, Proceedings of the Fourth International Semantic Web Conference ISWC2005, November 2005.

[51] Mika, P.: Social Networks and the Semantic Web. Proceedings of the International Conference on Web Intelligence WI2004, IEEE Computer Society Press, 2004.

[52] Moats, R.: URN Syntax, RFC 2141, IETF, May 1997.

[53] Mockapetris, P.: Domain names – implementation and specification, STD 13, RFC 1035, November 1987.

[54] Noy, N. and D. McGuinness: Ontology Development 101: A Guide to Creating Your First Ontology, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[55] O'Reilly, T: What is Web 2.0, Design Patterns and Business Models for the Next Generation of Software, http://www.oreilly.com, 2005.

[56] Passin, T.: Explorer's Guide to the Semantic Web, Manning, ISBN 0470 84867 7, 2004.

[57] Postel, J. and Reynolds, J.: File Transfer Protocol (FTP), RFC 959, IETF, October 1985.

[58] Powers, S.: Practical RDF, O'Reilly, ISBN 0596002637, 2003.

[59] Reichinger, K. and Baumgartner R.: Introducing ENUM to the Semantic Web, Proceedings of the IASTED International Conference on Web Technologies, Applications and Services, WTAS2005, July 2005.

[60] Reichinger, K., Baumgartner, R., and Reichinger, G.: Applications Utilising the PHOAF Prototype for Integrated ENUM and FOAF Queries, Proceedings of Information Resources Management Association International Conference, IRMA2006, May 2006.

[61] Reichinger, K., Baumgartner, R.: SEMNUM RDF Vocabulary Specification v.01, DBAI Technical Report, http://www.dbai.tuwien.ac.at/local/reports/semnum.pdf, December 2004.

[62] Reichinger, K., Reichinger, G., Baumgartner, R.: A Prototype for Integrated ENUM and FOAF Queries, Demo Paper and Poster, International Semantic Web Conference, ISWC2005, November 2005.

[63] Reichinger, K.: IANA Registration for ENUMservice foaf, draft-reichinger-enum-foaf-01.txt, Internet Draft, IETF, August 2006.

[64] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E.: SIP: Session Initiation Protocol, RFC 3261, IETF, June 2002.

[65] Schulzrinne, H.: The tel URI for Telephone Numbers, RFC 3966, IETF, December 2004.

[66] Shockey, R.: IETF – ENUM, ITU-T Workshop for International Regulators, Presentation Slides, www.itu.int/osg/spu/enum/workshopjan01/annex4-shockey.ppt, January 2001.

[67] Staab, S. and Stuckenschmidt, H.: SemanticWeb and Peer-to-Peer, Decentralized Management and Exchange of Knowledge and Information, Springer, ISBN-10 3-540-28346-3, 2006.

[68] Stollberg, M., Lausen, H., Lara, R., Ding, Y., Han, S. and Fensel, D.: Towards Semantic Web Portals, Proceedings of the WWW2004, May 2004.

[69] Toyoda, K., Ohno, H., Murai, J. and Wing, D.: A Simple Mode of Facsimile Using Internet Mail, RFC 2305, IETF, March 1998.

[70] Visser, U.: Intelligent Information Integration for the Semantic Web, Springer, ISBN: 3-540-22993-0, August 2004.

[71] World Wide Web Consortium, "Resource Description Framework", W3C Recommendation, February 2004.

[72] World Wide Web Consortium, "Web Ontology Language", W3C Recommendation, February 2004.

[73] World Wide Web Consortium: Extensible Markup Language (XML) 1.0 (Fourth Edition), W3C Recommendation, http://www.w3.org/TR/REC-xml/, August 2006.

[74] World Wide Web Consortium: Extensible Markup Language (XML) 1.1 (Second Edition), W3C Recommendation, http://www.w3.org/TR/xml11/, August 2006.

[75] World Wide Web Consortium: OWL Web Ontology Language Overview, W3C Recommendation, http://www.w3.org/TR/owl-features/, February 2004.

[76] World Wide Web Consortium: OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation, http://www.w3.org/TR/owl-semantics/, February 2004.

[77] World Wide Web Consortium: OWL-S: Semantic Markup for Web Services, W3C Member Submission, http://www.w3.org/Submission/OWL-S/, November 2004.

[78] World Wide Web Consortium: RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, http://www.w3.org/TR/rdf-schema/, February 2004.

[79] World Wide Web Consortium: RDQL - A Query Language for RDF, W3C Member Submission, http://www.w3.org/Submission/RDQL/, January 2004.

[80] World Wide Web Consortium: SPARQL Protocol for RDF, W3C Candidate Recommendation, http://www.w3.org/TR/rdf-sparql-protocol/, April 2006.

[81] World Wide Web Consortium: SPARQL Query Language for RDF, W3C Working Draft, http://www.w3.org/TR/rdf-sparql-query/, October 2006.

[82] World Wide Web Consortium: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission, http://www.w3.org/Submission/SWRL/, May 2004.

[83] Zeilenga, K.: Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map, RFC 4510, IETF, June 2006.

Annex

Annex 1: IANA Registration for ENUMservice foaf

This section contains the original copy of the author's Internet draft "IANA Registration for ENUMservice foaf" [63] as submitted to the IETF in August 2006. The former (initial) version was submitted in February 2006 and presented at the 65th IETF meeting in Dallas, March 2006.

ENUM -- Telephone Number Mapping Working Group — K. Reichinger, TU Wien
Internet-Draft — August 17, 2006
Intended status: Informational
Expires: February 18, 2007

IANA Registration for Enumservice foaf
draft-reichinger-enum-foaf-01

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on February 18, 2007.

Copyright Notice

Abstract

This memo registers the Enumservice "foaf" using the URI schemes "http" and "https" according to the IANA Enumservice registration process defined in RFC3671. The Enumservice "foaf" is to be used to refer from an ENUM domain name to the location of a FOAF RDF file using the corresponding E.164 telephone number.

Clients may use data retrieved from a FOAF RDF file to provide caller

or callee with information available within the Friend-Of-A-Friend (FOAF) Semantic Web application. For example, the caller might be presented with personal information on the callee (e.g. name, gender and various online attributes) as well as information on the callee's social context (e.g. relations to friends or colleagues). Information collected from FOAF can be used before, during or after a communication is established.

Table of Contents

1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [5].

2. Introduction

ENUM [1] uses the Domain Name System (DNS) [3] for mapping E.164 telephone numbers [13] to Uniform Resource Identifiers (URIs) [4]. Therefore E.164 numbers are converted to ENUM domain names through means described further in RFC3761.

'Friend-Of-A-Friend' (FOAF) [2] describes a Semantic Web [16] project for machine-readable modelling of homepage-like content and social networks. The FOAF specification defines terms to be used in statements someone can make about someone else, such as name, gender and various online attributes, e.g. e-mail address, instant messaging identifier, VoIP address or web URL. FOAF is based on the Resource Description Framework (RDF) [11] defined using the Web Ontology Language (OWL) [12]. Typically, the FOAF RDF file is named foaf.rdf and made publicly available on the Web. The usage of FOAF to describe people and their relationships has become popular amongst bloggers and in emerging Web 2.0 applications.

Integrating FOAF and ENUM [14] potentially offers a variety of Semantic Web applications [15] to be accessed by means of a telephone number. The introduction of a specific Enumservice dedicated to FOAF significantly eases that integration.

This memo registers an Enumservice according to the guidelines given in RFC3761 to be used for provisioning in the services field of a NAPTR [13] resource record to indicate what class of functionality a given end point offers. The registration is defined within the Dynamic Delegation Discovery System (DDDS) [6][7][8][9][10] hierarchy, for use with the "E2U" DDDS application defined in RFC3761.

This memo registers the Enumservice "foaf" using the URI schemes "http" and "https" according to the IANA Enumservice registration process defined in RFC3671. The Enumservice "foaf" is to be used to refer from an ENUM domain name to the location of a FOAF RDF file using the corresponding E.164 telephone number.

3. Enumservice Registrations - foaf

The Enumservice registered in this section indicates that the resource identified by the associated URI is a source of FOAF data.

Enumservice Name: "foaf"

Enumservice Type: "foaf"

Enumservice Subtype: N/A

URI Schemes: "http", "https"

Functional Specification:
This Enumservice indicates that the resource identified by the associated URI is a source of FOAF data. If the URI scheme "https" is used, the resource can be fetched by using TLS or the Secure Socket Layer protocol.

Security Considerations: see Section 5

Intended Usage: COMMON

Authors: Kurt Reichinger (see 'Authors' section for contact details)

4. Example

An example ENUM entry referring to a FOAF RDF file could look like following:

```
$ORIGIN 1.0.1.1.1.1.5.5.5.0.8.7.3.4.e164.arpa.
@ IN NAPTR 100 10 "u" "E2U+foaf" !^.*$!http://foo.bar/foaf.rdf!" .
```

Performing an ENUM query for the Austrian E.164 telephone number +43780555111101 will result in a referral to the web URL http://foo.bar/foaf.rdf indicating that a FOAF RDF file associated with that telephone number can be accessed there.

5. Security & Privacy Considerations

5.1. ENUM Record

With ENUM utilising the DNS - a globally distributed and publicly accessible database - all information contained in DNS records must be considered publicly available. Thus, data can be harvested, stored and re-used by third parties, e.g. for generating lists of targets for sending of unrequested information. This could result in being targeted with SPAM (e-mail), SPIT (VoIP calls), junk fax, junk SMS or other unwanted information. Even after removing the DNS entry and the referred resource, copies of the information could still be available.

Information published in ENUM records could reveal associations between E.164 numbers and their owners - especially if DNS records contain personal identifiers or domain names for which ownership information can easily be obtained.

However, it is important to note that the ENUM record itself does not need to contain any personal information. It just points to a location where access to personal information could be granted.

ENUM records pointing to third party resources can easily be provisioned on purpose by the ENUM domain owner - so any assumption about the association between a number and an entity could therefore be completely bogus unless some kind of identity verification is in place. This verification is out of scope for this memo.

5.2. FOAF File

FOAF files describe persons and online communities explicitly focusing on making the content easily machine-readable, which makes FOAF potentially vulnerable to automated data collecting (by e.g. crawlers or scutters). Furthermore, in most application scenarios FOAF relies on information being publicly available on the Web, although use cases in closed environments are possible as well.

FOAF files potentially contain links to a rich variety of personal data making it of interest to data harvesters, e.g. for generating lists of targets for unrequested information. This could result in being targeted with SPAM (e-mail), SPIT (VoIP calls), "junk" fax, "junk" SMS or other unwanted information. Even after removing the FOAF RDF file and referred resources, copies of the information could still be available.

Content, administration and publication of FOAF RDF files is under the responsibility of the individual FOAF RDF file owner. FOAF files easily can be created and published on the Web by anyone - so any assumption about data from a FOAF RDF file and an entity could therefore be completely bogus unless some kind of identity verification is in place. This verification is out of scope for this memo.

6. IANA Considerations

This memo requests registration of the "foaf" Enumservice according to the definitions in this document and RFC3761.

7. References

7.1. Normative References

[1] Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 3761, April 2004.

[2] Brickley, D. and L. Miller, "FOAF Vocabulary Specification", http://xmlns.com/foaf/0.1 , July 2005.

7.2. Informative References

[3] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[4] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[6] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part One: The Comprehensive DDDS", RFC 3401, October 2002.

[7] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm", RFC 3402, October 2002.

[8] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

[9] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Four: The Uniform Resource Identifiers (URI)", RFC 3404, October 2002.

[10] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Five: URI.ARPA Assignment Procedures", BCP 65, RFC 3405, October 2002.

[11] World Wide Web Consortium, "Resource Description Framework",

W3C Recommendation , February 2004.

[12] World Wide Web Consortium, "Web Ontology Language", W3C Recommendation , February 2004.

[13] ITU Telecommunication Standardization Sector, "The International Public Telecommunication Numbering Plan", ITU-T Recommendation E.164, May 1997.

[14] Reichinger, K. and R. Baumgartner, "Introducing ENUM to the Semantic Web", Proceedings of the IASTED International Conference on Web Technologies, Applications and Services WTAS2005, July 2005.

[15] Reichinger, K., Baumgartner, R., and G. Reichinger, "Applications Utilising the PHOAF Prototype for Integrated ENUM and FOAF Queries", Proceedings of Information Resources Management Association International Conference IRMA2006, May 2006.

[16] Berners-Lee, T., Hendler, J., and O. Lassila, "The Semantic Web", Scientific American , May 2001.

Author's Address

Kurt Reichinger
Vienna University of Technology - DBAI Group
Favoritenstrasse 9
A-1040 Vienna
Austria

Phone: +43 1 58058 306
Email: reiching@dbai.tuwien.ac.at
URI: http://www.dbai.tuwien.ac.at/

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

Annex 2: PHOAF Code

Java Class Diagram

A class diagram is a type of static structure diagram describing the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It gives an overview with regard to the principal operations and functions of a software system. Typically, a class in such a software system is represented by a box with the name of the class written inside. Within the same box, the compartment below the class name shows the class's attributes, i.e. its properties. Each attribute is shown with its name, and optionally with its type, initial value, and other properties. The class's operations, i.e. its methods, can appear in another compartment. Each operation is shown with at least its name, and optionally also with its parameters and return type.

The Java class diagram for the PHOAF prototype (as depicted in Figure 68 and Figure 69) illustrates the relations between the following Java classes used in PHOAF, e.g. explicitly showing which class imports or instantiates data from another class.

- DC.java
- DnsNaptrRecord.java
- DnsNaptrResult.java
- EnumFoafException.java
- EnumFoafUtil.java
- FOAF.java
- FoafDocument.java
- FoafPerson.java
- HtmlRenderer.java
- NaptrRecordList.java
- RDF.java
- Result.java
- SimpleFoafHandler.java

In addition, the following HTML and CSS files are used for the PHOAF user interface. The file test.html is the Web user interface for standard PHOAF, the file enumplus.html provides the interface for enhanced PHOAF (allowing for Web 2.0 integration as introduced in detail in section 8.6). The file enumfoaf.css is a Cascading Style Sheets document defining the format used from both HTML files mentioned above.

- test.html
- enumplus.html
- enumfoaf.css

Furthermore, the following JSP files are used as explained in section 7.

- lookupfoaffile.jsp
- parsefoaffile.jsp
- querydnsnaptr.jsp
- querydnsnaptrfoaf.jsp

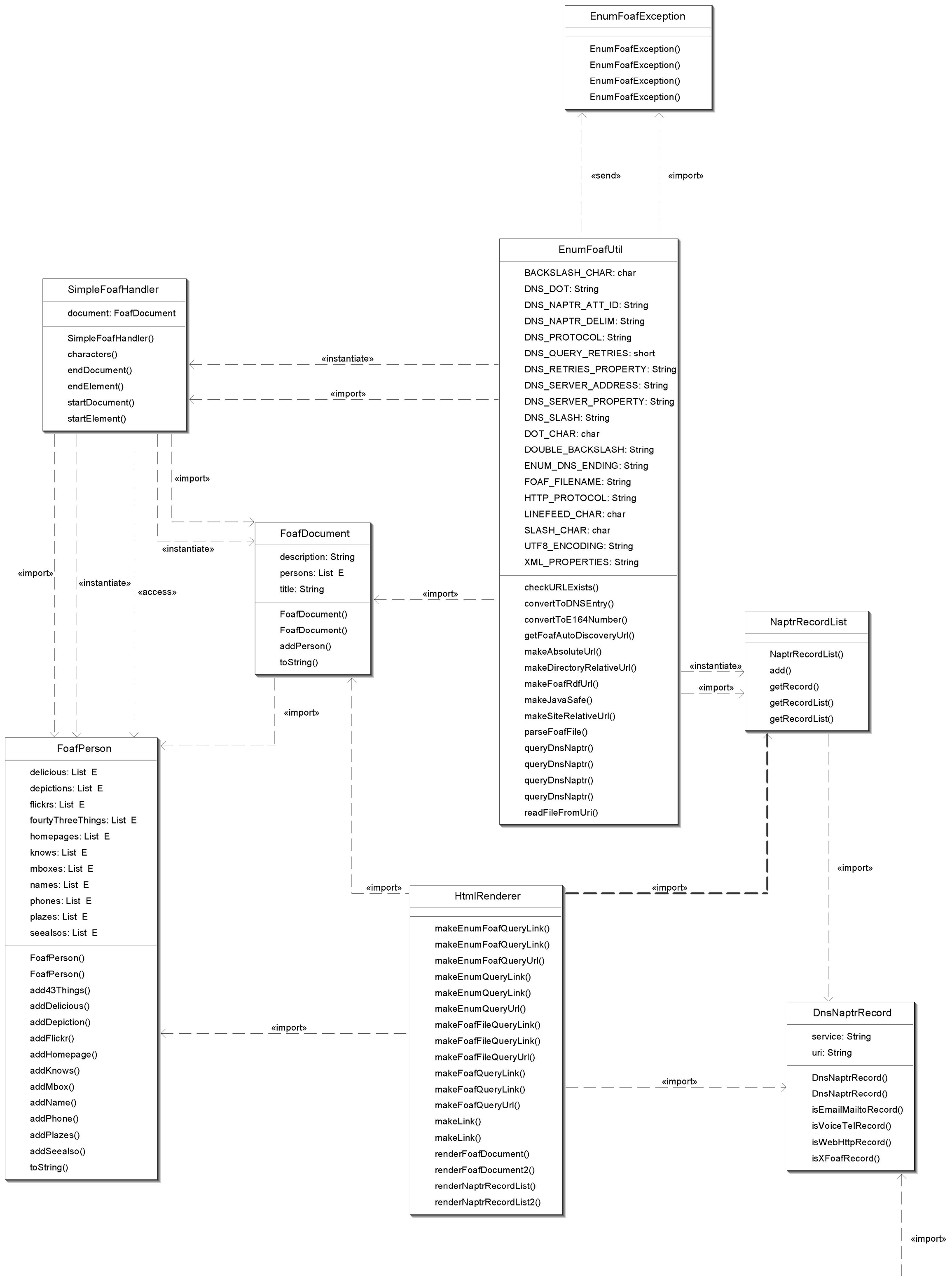

Figure 68: PHOAF Java class diagram (part 1) – continued on next page

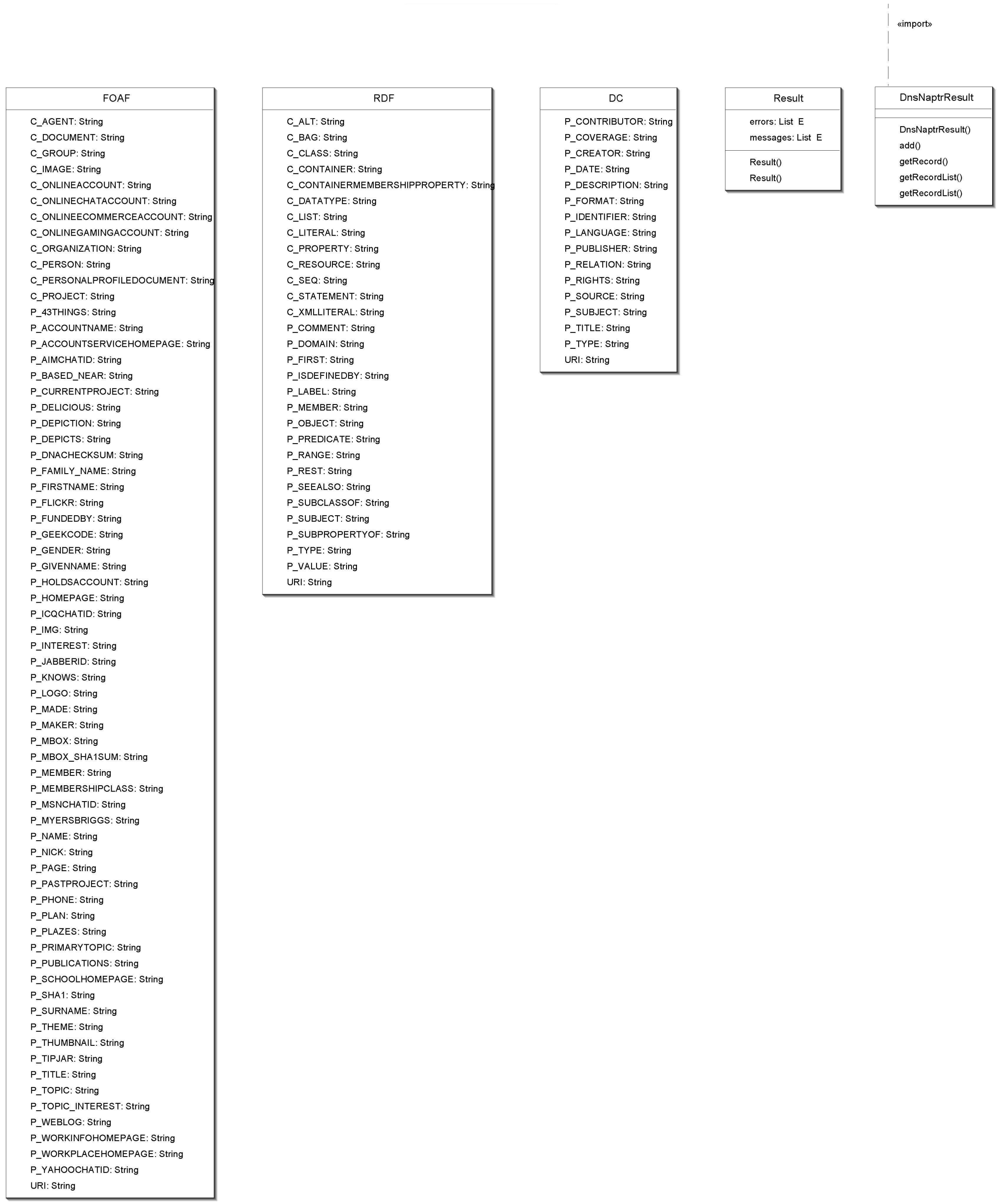

Figure 69: PHOAF Java class diagram (part 2) – continued from previous page

Curriculum Vitae

Name:	Dipl.-Ing. Kurt Reichinger
Address:	Ettenreichgasse 50 / 4 / 53 1100 Vienna, Austria
E-mail:	reiching@dbai.tuwien.ac.at kurt.reichinger@rtr.at
Date of Birth:	August 13, 1965
Place of Birth:	Vienna, Austria
Nationality:	Austrian
Family Status:	Married

Education

2003 –	TU Wien (Ph.D. in Computer Science) Target Degree: Dr. techn.
2000 – 2001	FernUni Hagen (Post Graduate Course Business Administration) Degree: Diploma Certificate
1984 – 1996	TU Wien (M.Sc. in Communications Engineering) Degree: Dipl.-Ing.
1979 – 1984	HTBLVA Wien 1 Schellinggasse (Communications Engineering) Degree: Ing.
1975 – 1979	High School – BRG 16 Schuhmeier-Platz, Wien
1971 – 1975	Primary School – Volksschule Koppstraße, Wien

Work Experience

2001 –	Austrian Regulatory Authority for Broadcasting and Telecoms (RTR) Technical Department – Senior Analyst
1999 – 2001	Telekom Austria AG International Networks Department – Director
1997 – 1999	Telekom Austria AG Project Management Services Department – Deputy Director
During studies	Hardware Prototyping at various Companies OGM – Österreichische Gesellschaft für Marketing Schrack AG Philips AG