Workshop held at the
Fifth International Semantic Web Conference
ISWC 2006
November 5 - 9, 2006

Proceedings of the First International Workshop on Applications and Business Aspects of the Semantic Web SEBIZ 2006

November 6, 2006
Athens, Georgia, USA

Edited by
Elena Paslaru Bontas Simperl,
Martin Hepp, and
Christoph Tempich

The workshop Website is available online at http://www.ag-nbi.de/conf/SEBIZ06/

Contents

Introduction

Motivation

Within the past five years, the Semantic Web research community has brought to maturity a comprehensive set of foundational technology components, and this both at the conceptual level and in the form of prototypes and software. This includes, among other assets, ontology engineering methodologies, standardized ontology languages, ontology engineering tools, and other infrastructure like APIs, repositories, and scalable reasoners, plus a plethora of work for making the Deep Web and computational functionality in the form of Web Services accessible at a semantic level. However, in order for these achievements to provide a feasible basis for ontologies to start-up at large scale corporate applications, they should be complemented by methods, validated by practical application, which allow enterprises to:

- Effectively adopt ontology based systems in the existing infrastructure. In particular this requires
  - best practices and convincing showcases
  - means to monitor the quality of the ontology development and deployment processes
  - estimate and control the costs involved in the development and usage of ontologies
  - investigate the costs and benefits of applying particular development or deployment strategies in specific application settings
- Evaluate the quality of existing ontologies and ontology engineering methodologies, methods and tools. In particular the dissemination of ontology-based technologies at corporate level requires methods to measure the usability of a particular ontology in a specific business scenario estimate the business value of ontologies, but also objective means to compare among methodologies, methods and tools dealing with them.

The availability of best practices, convincing showcases, metrics as well as quantitative and qualitative measurements assisting particular stages of ontology engineering processes are essential requirements for organizations to be able to optimize these processes.

The Workshop

The SEBIZ Workshop on Applications and Business Aspects of the Semantic Web brought together 30 professionals affiliated to both industry and academia. The workshop program included a short introduction talk held by the organizers, four technical presentations and extensive discussions.

The workshop organizers, Elena Simperl, Martin Hepp and Christoph Tempich, received 7 submissions of papers in response to the call for papers. As a result of the peer reviewing process, 5 of these were selected for publication in these proceedings. The program committee consisted of the following Semantic Web experts from industry or academia: Richard Benjamins (iSOCO), Chris Bizer (Free University of Berlin), Christoph Bussler (Cisco Systems), Jorge Cardoso (University of Madeira), Oscar Corcho (University of Manchester), Roberta Cuel (University of Trento), John Davies (BT), Jos de Bruijn (University of Innsbruck), Tommaso Di Noia (Politecnico di Bari), John Domingue (Open University), Dieter Fensel (University of Innsbruck), Doug Foxvog (National University of Ireland Galway), Fausto Giunchiglia (University of Trento), Michel Klein (Vrije Universiteit Amsterdam), Juhnyoung Lee (IBM Research), Alain Leger (France Telecom), Miltiadis Lytras (Athens University of Economics & Business), Dumitru Roman (University of Innsbruck) , York Sure (University of Karlsruhe), Robert Tolksdorf (Free University of Berlin), Ioan Toma (University of Innsbruck) and Yuxiao Zhao (Linköping University). The organization committee would like to thank all PC members for their thorough and substantial reviews, which were crucial for the success of the actual workshop.

In the first session of the workshop Elena Simperl held, on behalf of the organizers, a short talk, which outlined the motivation and objectives of the workshop, and introduced the technical program. This consisted of two sessions of paper presentations, followed by a closing session in which the attendees participated in a lively discussion on the main topics covered by the event, and pointed out open issues for making the Semantic Web a success at industry level. In the following, we summarize the main points.

Technical Presentations

The presentations held on this workshop covered the following topics: Business Intelligence (BI), Ontology-based content integration tasks in business and public sector applications, Human Resources (HR), Metrics to evaluate and compare existing ontologies, Metrics to determine the usability of a particular ontology in a specific business scenario, and Quality frameworks for ontologies.

The paper *"OntoCAT: An Ontology Consumer Analysis Tool and Its Use on Product Services Categorization Standards"* by Valerie Cross and Anindita Pal gives an in-depth overview of the field of ontology evaluation. It introduces OntoCAT, a tool which computes a comprehensive set of metrics for use by the ontology consumer or knowledge engineer to assist in ontology evaluation for re-use.

The paper *"Improving the recruitment process through ontology-based querying"* by Malgorzata Mochol, Lyndon Nixon and Holger Wache approaches the problem of approximate reasoning in the context of an eRecruitment scenario. It describes a query relaxation method which demonstrates the benefit of using formal ontologies for improving the retrieval performance and the user-friendliness of a semantic job portal.

Leticia Gutierrez Villarias presented the paper *"Enhancing Data and Processes Integration and Interoperability in Emergency Situations: a SWS based Emergency Management System"* by Alessio Gugliotta, Leticia Gutierrez Villarias, Vlad Tanasescu, John Domingue, Mary Rowlatt, Marc Richardson and Sandra Stincic. The talk describes how semantic technologies, and in particular Semantic Web Services, can be successfully deployed to integrate data and applications in the field of emergency management.

A second use case for the Semantic Web was discussed in the paper *"Personalized Question Answering: A Use Case for Business Analysis"* by VinhTuan Thai, Sean O'Riain, Brian Davis and David O'Sullivan. The approach provides evidence on the importance of using domain semantics in question answering tasks to resolve ambiguities and to improve the recall for retrieving relevant passages.

Conclusions and Outlook

The workshop gave clear evidence that semantic technologies are experiencing a shift from a pure research topic to real-world applications. Further on, the presentations and the discussions among the attendees showed the substantial

interest of academia in transferring the results achieved so far to industry. On the other hand, industry seems to be well aware of these achievements and of the added value of using semantics for data and application integration purposes.

The main results of the SEBIZ06 workshop and associated discussions can be summarized as follows:

- The results achieved by the research community in the last decade provide the core building blocks for realizing the Semantic Web. France Telecom, HP, IBM or Vodafone provide first success stories for deploying semantic technologies within enterprizes, while companies such as Ontoprise, TopQuadrant, Cerbera, Oracle or Altova are established technology vendors.
- There was consensus among the workshop participants that the mainstream adoption of semantic technologies will take about five years from now.

Further on, the discussion revealed a series of open issues, which are crucial for the uptake of semantic technologies at industrial level:

- A major drawback when applying semantics within enterprizes is the lack of tools leveraging semantic data from existing legacy systems.
- Business people require means to evaluate the technology and the content.
- The business aspects of the development and deployment of semantic technologies are still marginally addressed, thus impeding their large scale adoption.

Berlin, Innsbruck and Karlsruhe
November, 2006

Elena Simperl
Martin Hepp
Christoph Tempich

Enhancing Data and Processes Integration and Interoperability in Emergency Situations: a SWS based Emergency Management System

Alessio Gugliotta[1], Rob Davies[2], Leticia Gutiérrez-Villarías[2],Vlad Tanasescu[1], John Domingue[1], Mary Rowlatt[2], Marc Richardson[3], Sandra Stinčić[3]

[1] Knowledge Media Institute, The Open University,
Walton Hall, Milton Keynes, MK7 6AA, UK
{v.tanasescu, a.gugliotta, j.b.domingue}@open.ac.uk
[2] Essex County Council, County Hall,
Chelmsford, CM1 1LX, UK
{Leticia.gutierrez, maryr}@essexcc.gov.uk
rob.davies@mdrpartners.com
[3] BT Group
Adastral Park Martlesham, Ipswich IP5 3RE, UK
{marc.richardson, sandra.stincic}@bt.com

Abstract. In this paper we describe a powerful use case application in the area of emergency situations management in which to illustrate the benefits of a system based on Semantic Web Services (SWS), through the automation of the business processes involved. After creating Web services to provide spatial data to third parties through the Internet, semantics and domain ontologies were added to represent the business processes involved, allowing: ease of access and combination of heterogeneous data from different providers; and automatic discovery, access and composition to perform more complex tasks. In this way, our prototype contributes to better management of emergency situations by those responsible. The work described is supported by the DIP (Data, Information and Process Integration with Semantic Web Services) project. DIP (FP6 – 507483), an Integrated Project funded under the European Union's IST programme.

1. Introduction

In an emergency response situation there are predefined procedures which set out the duties of all agencies involved. A very wide range of agencies is often involved in the management of an emergency situation, potentially involving a huge data provision and communication requirement between them. Needs and concerns are escalated through a specified chain of command, but the organisations are independent of one another and decisions have to be made rapidly, based on knowledge of the situation (e.g. the type of problem, the site, and the population affected) and the data available. Gathering all the data in a manual or semi-automated way takes time and resources that those responsible for emergency planning and incident response may not have.

Having data and resources available through the internet, companies and public organizations can easily and inexpensively share information with customers and partners. Web Services (WS) would allow emergency planning agencies and rescue corps to interoperate and share vital information easily. The supplied services are autonomous and platform-independent computational elements. They can be described, published, discovered, orchestrated, and programmed using XML artifacts for the purpose of developing massively distributed interoperable application. Unfortunately, despite progress in the use of standards for Web Service description (WSDL [9]) and publishing (UDDI [10]), the syntactic definitions used in these specifications do not completely describe the capability of a service and cannot be understood by software programs. A human developer is required to interpret the meaning of inputs, outputs and applicable constraints as well as the context in which services can be used.

Semantic Web Services (SWS) technology aims to alleviate these problems. It combines the flexibility, reusability, and universal access that typically characterize a WS, with the expressivity of semantic mark-up, and reasoning in order to make feasible the invocation, composition, mediation, and automatic execution of complex services with multiple paths of execution, and levels of process nesting. As a result, computers can automatically interoperate and combine information, creating a comprehensive and most relevant possible response which is seamlessly delivered to end-users in real time.

The Emergency Management System (EMS) envisaged within the DIP use case will provide a decision support system which will assist the emergency planning officer to automatically gather and analyze relevant information in a particular emergency scenario, through the adoption of SWS technology. This should improve the quality of the information available to emergency managers in all the phases of emergency management: before (planning), during (response), and after (evaluation and analysis); thereby facilitating their work and improving the quality of their decisions in critical situations.

Our work contributes to raise the awareness of potential SWS benefits in real-world applications - ease the creation of infrastructure in which new services can be added, discovered and composed continually, and the organization processes automatically updated to reflect new forms of cooperation - and promote the availability of working SWS platforms.

2. Integrated Emergency Management (IEM) Requirements

In the definition of the use case scenario, an attempt has been made to bring together the needs of all the groups that would be involved in case of an emergency occurring in Essex - a large region in South East England (UK). We have conducted interviews with emergency planning personnel in Essex County Council (ECC) and several other agencies which are involved in various types of emergency scenario (e.g. Meteorological Office; police, fire, ambulance emergency services; traffic control service; British Airport Authority; and other County Councils surrounding Essex). As a result of this work, the following main requirements were delineated:

R1. *In an emergency event all the authorities involved have to cooperate and provide relevant data to each others upon request. This data comes from many sources in many different formats.* As required in the Civil Contingencies Act 2004 [1]: "local responder bodies have to co-operate in preparing for and responding to emergencies through a Local Resilience Forum (LRF)". ECC is aware of the importance of multi-agency working and consequently has belonged for many years to several emergency groups and networks. All of these groups collaborate now under the Essex Resilience Forum. There is also in Essex an "Essex Emergency Services Coordinating Group (EESCG)" which is formed by representatives from Essex Police, Essex Fire and Rescue Service, British Transport Police, Essex Ambulance Service, Maritime Coastguard Agency, Military and Local Authorities.

R2. *Interoperation and collaboration among many agencies in an emergency situation follow predefined procedures which set out agency's duties.* As stated in the COPE (Combined Operational Procedures for Essex) document [2]: "The purpose of the group is to develop, maintain and improve effective co-ordination between the Emergency Services and the principal emergency Support Organizations and to identify the means to ensure effective co-ordination and regular liaison between those services in the planned response to emergencies."

R3. *Geographical Information Systems (GIS) applied to an IEM scenario can ease the integration, storage, querying, analysis, modeling, reporting and mapping of geographically-referenced data relevant for the emergency situation.* As stated in by the UK Emergency Planning College in their "Guide to GIS Applications in Integrated Emergency Management (IEM)" [4]: "Geography matters to IEM: hazards are spatially distributed, and generally very uneven in that distribution, vulnerable facilities are distributed and clustered in space, and resources may be sub-optimally located to deal with anticipated and actual emergencies".

R4. *Cross-border relationships are highly important in an emergency situation*, especially in the context of the Stansted area. The Airport is considered to be in its own 'territory' governed by British Airports Authority (BAA) and does not form part of a local government District. In the event of an emergency situation around Stansted, ECC needs to work closely with the other affected adjacent local government authorities, namely: Hertfordshire County Council and Uttlesford District Council and with BAA itself.

3. The Emergency Management System

We are developing an Emergency Management System (EMS), which is an end-user Web application providing e-Emergency services to customers. The system is intended to be used during the planning and response phases of an emergency. Provided services can cover all kinds of information concerned with emergencies - including information about hazardous weather, personnel involved in an emergency situation, rescue corps involved in the prevention response and recovery phases of an emergency situation, evacuation procedures, provision of supplies and help to affected people, location of damaged facilities and the consequences, assistance needed by vulnerable people, location of 'hotspots' etc.

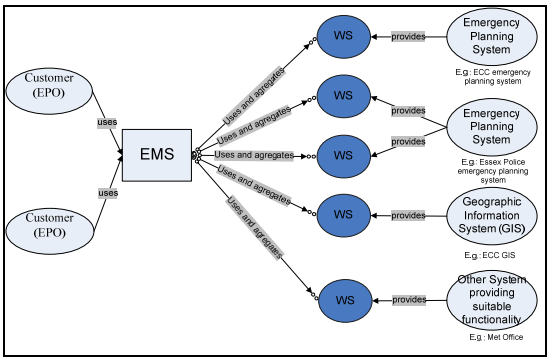


Figure 1 – Context Diagram

As depicted in Figure 1, there are three main actors in the general use case, which participate in this use case and with different roles. These are:

- *Customer (EPO):* The end user that requests the services provided by the EMS. They select and invoke services through a user-friendly emergency planning interface. We envisage this application will be used by the Emergency Planning Officers (EPO) in public organizations, and other emergency partners (Police, Fire & Rescue, Ambulance service, NHS, Rover Rescue, etc.). As a result we obtain a cross-border application (IEM requirement *R4*).
- *Emergency Planning and Geographical Information Service providers:* Governmental authorities, Ordnance Survey, Meteorological Office, emergency agencies, commercial companies, etc, which provide specific emergency planning services and spatially-related services through the Internet in the form of WS. They provide services to end users to improve collaboration in an emergency-based scenario (IEM requirements *R1*, *R3*).
- *EMS:* The intermediary between the customer and the providers. This management system holds all the functionalities for handling SWS - supporting automatic discovery, composition, mediation and execution. It exposes services to end-users, using existing emergency services and aggregating them into new high-level services in order to improve collaboration in an emergency-based scenario (IEM requirement *R2*). The EMS is considered to have a non-profit governmental basis and to serve public interests in case of an emergency. It interacts with customers (emergency planners and heads of rescue corps) via a user-friendly interface, allowing users to access and combine the different services provided by the service providers.

3.1 Use case

Several emergency-related scenarios were considered in order to pilot the prototype definition. With the collaboration of the ECC emergency planners, we finally decided to focus on a real past situation: *"Heavy snowstorm around the Stansted area and M11 corridor (Essex, UK)* on 31st January 2003*"*, in which thousands of motorists were trapped overnight on some of Britain's busiest motorways [3]. By focusing on a past event we ensure the availability of real data. An additional advantage is the

ability to compare the actions taken and the data available at that time, with the data and actions that would had been taken if a SWS-based emergency planning tool had been available.

3.2 Business process and data

The current version of the prototype focused on the planning phase. Figure 2 depicts the main goals to achieve (business processes) in a snowstorm hazardous situation before planning an adequate emergency response. The first step is to identify the affected area by analysing snow data. Then, the EPO has to locate suitable shelters for resting affected people and – not necessarily in this order - identify available relevant people (rescue corps) in the affected area. These goals are not merely retrieval operations, but involve sub-processes that select services and manipulate retrieved data according to situation-specific requirements. Semantics will be adopted to represent these decompositions. A detailed example is provided in Section 4.5.

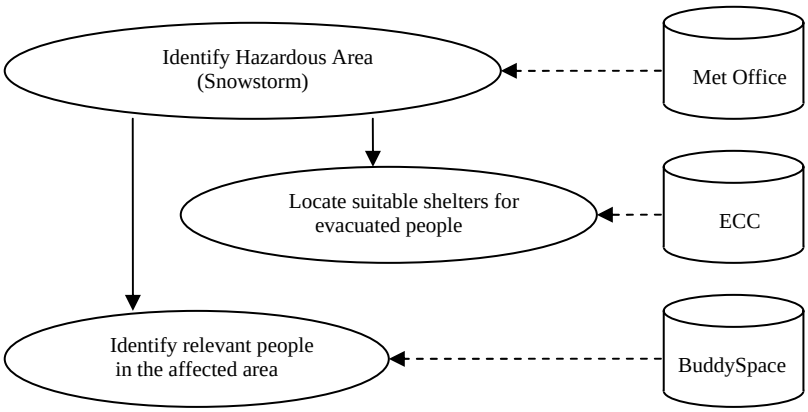


Figure 2 – Emergency procedure in a snowstorm hazardous situation.

The prototype will aggregate data and functionalities from the following three heterogeneous sources:

- *Meteorological Office*: a national UK organization which provides environmental resources and in particular weather forecast data. The prototype aggregates snow data related to the date of the snowstorm in question.
- *ECC geospatial and emergency data*: The prototype makes use of a wide range of geospatial data, such as administrative boundaries, buildings, Ordnance Survey maps, etc, as well as other data from the emergency department. Building related data is used to support searches for suitable rest centres.
- *BuddySpace* is an Instant Messaging client facilitating lightweight communication, collaboration, and presence management [5] built on top of the instant messaging protocol Jabber[1]. The BuddySpace client can be accessed on

[1] Jabber. http://www.jabber.org/

standard PCs, as well as on PDAs and on mobile phones (which in an emergency situation may be the only hardware devices available).

As many of the integrated real systems have security and access restriction policies, British Telecommunications (BT) has created a single corporate spatial data warehouse where all Meteorological Office and ECC data sources have been replicated in order to work with them in a safe environment, thereby providing suitable Web Services (WS) to work with. However, the prototype represents how this system would work in a distributed environment with heterogeneous and scattered data sources over Internet.

WS will provide a first level of interoperability by encapsulating functionality regardless of the specific technologies/protocols of the providers' legacy systems. Semantic descriptions will provide the final level of interoperability, allowing automation of all the stages of the WS use (mainly: discovery, composition and invocation). In Section 4, we will detail these aspects.

4. The Prototype

The main functional requirements of our SWS-enabled EMS are: (FR1) providing a graphic user interface (GUI) for customer interaction and displaying outputs: e.g. browser/visualization tool to display and select data layers on a map; (FR2) discovering, combining and invoking suitable Web Services for a user request; (FR3) providing a WS Execution Environment with control functions, error handling, and support for optional user interaction; (FR4) dealing effectively with heterogeneous resources, thus allowing for appropriate mediation facilities (Ontology-Ontology mediation has been identified in the earlier stages of the prototype, other kinds of mediation may be identified later); (FR5) providing interfaces for cooperation with GIS and emergency service providers.

In order to provide semantic and step toward the creation of added value services (FR2, FR3, FR4, FR5), we adopt WSMO [6] – a promising SWS framework – and IRS-III [7] – a tested implementation of this standard. The reference language for creating ontologies is OCML [8].

4.1 Semantic Web Services framework: WSMO and IRS-III

The Web Service Modeling Ontology (WSMO) [6] is a formal ontology for describing the various aspects of services in order to enable the automation of WS discovery, composition, mediation and invocation. The meta-model of WSMO defines four top level elements:

- *Ontologies*: provide the foundation for describing domains semantically. They are used by the three other WSMO components.
- *Goals:* define the tasks that a service requester expects a Web service to fulfil. In this sense they express the requester's intent.
- *Web Service* descriptions represent the functional behavior of an existing deployed Web service. The description also outlines how Web services communicate (*choreography*) and how they are composed (*orchestration*).

- *Mediators* handle data and process interoperability issues that arise when handling heterogeneous systems.

One of the main characterizing features of WSMO is that ontologies, goals and Web services are linked by mediators:

- *OO-mediators* enable components to import heterogeneous ontologies;
- *WW-mediators* link Web Services to Web Services;
- *WG-mediators* connect Web Services with Goals;
- *GG-mediators* link different Goals.

The incorporation of four classes of mediators in WSMO facilitates the clean separation of different mapping mechanisms. For example, an OO-mediator may specify an ontology mapping between two ontologies whereas a GG-mediator may specify a process or data transformation between two goals.

IRS-III, the *Internet Reasoning Service* [7], is a platform which allows the description, publication and execution of Semantic Web Services, according to the WSMO conceptual model.

Based on a distributed architecture communicating via XML/SOAP messages, it provides an execution environment for SWS; ontologies are stored by the server, and used in WSMO descriptions to support discovery, composition, invocation and orchestration of WS. It allows *one-click publishing* of "standard" program code to WS by automatically generating an appropriate wrapper. Standard WS or REST services can also be trivially integrated and described by using the platform.

Also, by extending WSMO goal and Web Service concepts, clients of IRS-III can invoke web services via goals. That is, IRS-III supports so called *capability-*, or *goal-driven* service invocation which allows the user to use only generic inputs, hiding the possible complexity of a chain of heterogeneous WS invocations.

4.2 Architecture

The general architecture of our semantically-enhanced prototype is depicted in Figure 3. As can be seen, it is a service oriented architecture (SOA) composed of the following four layers:

- *Legacy System layer:* consists of the existing data sources and IT systems available from each of the parties involved in the integrated application.
- *Service Abstraction layer:* exposes (micro-) functionality of the legacy systems as WS, abstracting from the hardware and software platforms. The adoption of existing Enterprise Application Integration (EAI) software facilitated the creation of required WS.
- *Semantic Web Service layer*: given a goal request, this layer, implemented in IRS-III will (i) discover a candidate set of Web services, (ii) select the most appropriate, (iii) mediate any mismatches at the data, ontological or business process level, and (iv) invoke the selected Web services whilst adhering to any data, control flow and Web service invocation requirements. To achieve this, IRS-III utilises the set of SWS descriptions, which are composed of goals, mediators, and Web services, supported by relevant ontologies.
- *Presentation layer:* is a Web application accessible through a standard Web browser. The goals defined within the SWS layer are reflected in the structure of

the interface and can be invoked either through the IRS-III API or as an HTTP GET request. The goal requests are filled with data provided by the user and sent to the Semantic Web Service layer. We should emphasise that the presentation layer may be comprised of a set of Web applications to support distinct user communities. In this case, each community would be represented by a set of goals supported by community related ontologies.

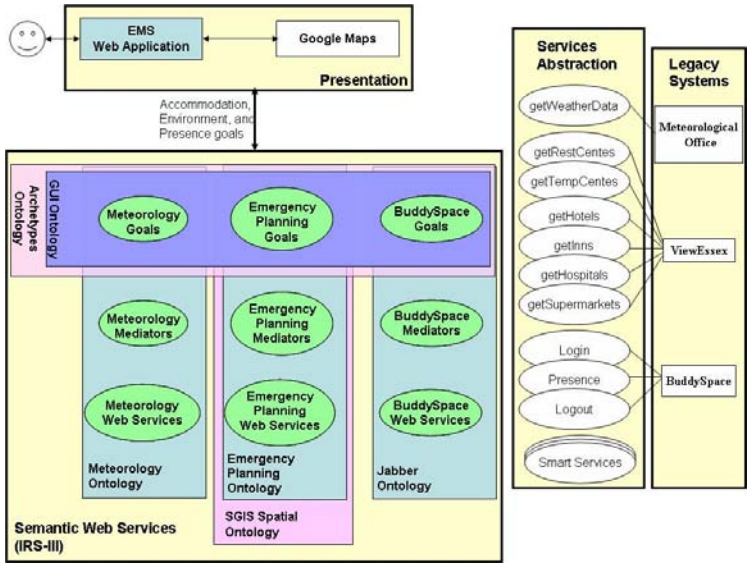


Figure 3. The EMS general architecture.

4.3 Services

We distinguish between two classes of services: *data* and *smart*. The former refers to the three data sources introduced in Section 3, and are exposed by means of WS:

- *Meteorological service:* this service provides weather information (e.g. snowfall) over a specific rectangular spatial area.
- *ECC Emergency Planning services*: using the ViewEssex data each service in this set returns detailed information on a specific type of rest centre within a given circular area. For example, the *'getHospitals'* Web service returns a list of relevant hospitals.
- *BuddySpace services*: these services allow presence information on online users to be accessed.

Smart services represent specific emergency planning reasoning and operations on the data provided by the data services. They are implemented in a mixture of Common Lisp and OCML and make use of the developed ontologies. In particular, we created a number of filter services that manipulate meteorological and GIS data according to emergency-specific requirements semantically described; e.g. rest centres with

heating system, hotels with at least 40 beds, easier accessible hospital, etc. The criteria used were gained from our discussions with the EPO's.

4.4 Semantic Web Services: Ontologies

In this and next section, we focus on the semantic description defined in the Semantic Web Services Layer. The following ontologies reflecting the client and provider domains were developed to support WSMO descriptions:

- *Meteorology, Emergency Planning and Jabber Domain Ontology*: representing the concepts used to describe the services attached to the data sources, such as *snow* and *rain* for Met Office, *hospitals* and *supermarkets* for ECC Emergency Planning, *session* and *presences* for Jabber. If a new source and the Web services exposing its data and functionalities are integrated, a new domain ontology has to be introduced - also reusing existing ontologies. The services, composed of the data types involved as well as its interface, have to be described in such a ontology usually at a level low enough to remain close from the data.

To get the information provided by WS up to the semantic level, we introduce *lifting operations* that allows the passage of data types instances from a syntactic level (xml) to an ontological one (OCML) specified in the domain ontology definitions. These lisp functions automatically extract data from SOAP messages and create the counterpart class instances. The mapping information between data types and ontological classes is defined at design time by developers.

- *HCI Ontology*: part of the user layer, this ontology is composed of HCI and user-oriented concepts. It allows to lowering from the semantic level results for the particular interface which is used (e.g. stating that Google Maps API is used, defining "pretty names" for ontology elements, etc.). Note that although the choice of the resulting syntactic format depends of the chosen lowering process, concepts from the HCI ontology are used in order to achieve this transformation in a suitable way.
- *Archetypes Ontology*: part of the user layer, this is a minimal ontological commitment ontology aiming to provide a cognitively meaningful insight into the nature of a specialized object; for example, by conveying the cognitive ("naïve") feeling that for example an hospital, as a "container" of people and provider of "shelter" can be assimilated to the more universal concept of "house", which we consider to be as an *archetypal* concept, i.e. based on image schemata and therefore supposed to convey meaning immediately. It is moreover assumed that any client, whilst maybe lacking the specific representation for a specific basic level concept, knows its archetypal representation.
- *Spatial Ontology*: a part of the mediation layer, it describes GIS concepts of location, such as coordinates, points, polygonal areas, and fields. It also allows describing spatial objects as entities with a set of attributes, and a location.

The purpose of the HCI, Archetypes and Spatial ontologies is the aggregation of different data sources on, respectively, a representation, a cognitive and a spatial level. Therefore we can group them under the appellation *aggregation* ontologies. They allow the different data sources to be handled and presented in a similar way. Inversely to the lifting operations, *lowering operations* transform instances of

aggregation ontologies into syntactic documents to be used by the server and client applications. This step is usually fully automated since aggregation ontologies are, by definition, quite stable and unique.

Context Ontology: the context ontology allows describing *context n-uples* which represent a particular situation. In the emergency planning application, context n-uples have up to four components, the use case, the user role, the location, and the type of object. Contexts are linked with goals, i.e. if this type of user accesses this type of object around this particular location, these particular goals will be presented. Contexts also help to inform goals, e.g. if a goal provides information about petrol stations in an area, the location part of the context is used to define this area, and input from the user is therefore not needed. Each time an object is displayed by a user at a particular location, a function of the context ontology provides the goals which need to be displayed and what inputs are implicit.

4.5 Semantic Web Services: WSMO descriptions

As depicted in Figure 3, the goals, mediators, and Web Services descriptions of our application currently link the UK Meteorological Office, ECC Emergency Planning, and BuddySpace Web services to the user interface. Correspondingly, the Web Service goal descriptions use SGIS spatial, meteorology, ECC Emergency Planning and Jabber domain ontologies whilst the goal encodings rely on the GUI and archetypes ontologies. Mismatches are resolved by the defined mediators. As introduced in the previous section, the inputs of the WS (XML in our particular scenario, but any other format could be provided) are lifted to the ontology, and, after invoking a Goal, the results are lowered back into XML so the results can be displayed back to the user. For illustration purposes, a small portion of the SWS descriptions are shown in Figure 4. The example details the main goal "*Locate suitable shelters for evacuated people*" introduced in Section 3.2.

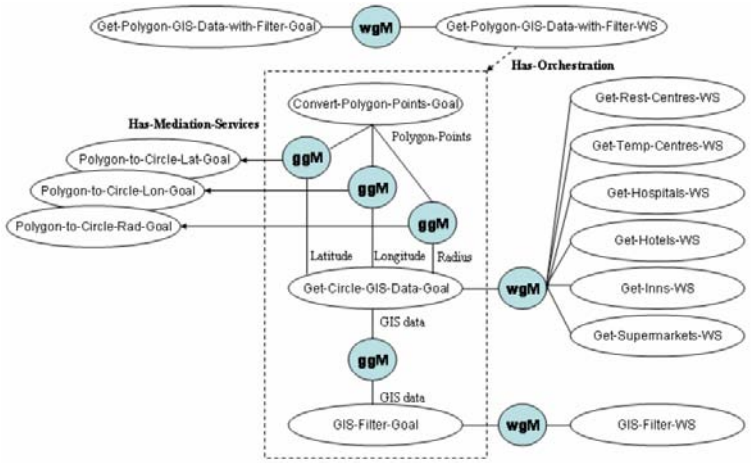


Figure 4. A portion of WSMO descriptions for the EMS prototype.

Get-Polygon-GIS-data-with-Filter-Goal represents a request for available shelters within a delimited area. The user specifies the requirements as a target area, a sequence of at least three points (a polygon), and a shelter type (e.g. hospitals, inns, hotels). As mentioned above the set of ECC Emergency Planning Web services each return potential shelters of a specific type with a circular query area. The obtained results need to be filtered in order to return only shelters correlated to emergency-specific requirements (for example a snowstorm). The process automated in our application is usually performed by EPO manually.

From a SWS point of view the problems to be solved by this particular portion of the SWS layer included: (i) *discovering* the appropriate ECC Emergency Planning Web service; (ii) *meditating* the difference in area representations (polygon vs. circular) between the goal and Web services; (iii) *composing* the retrieve and filter data operations. Below we outline how the WSMO representations in Figure 4 address these problems.

- *Web service discovery (FR2):* each SWS description of ECC Emergency Planning service defines, in its capability, the specific class of shelter that the service provides. Each definition is linked to the *Get-Circle-GIS-Data-Goal* by means of a unique WG-mediator (shown as wgM). The inputs of the goal specify the class of shelter, and the circular query area. At invocation IRS-III discovers through the WG-mediator all associated Web services, and selects one on the basis of the specific class of shelter described in the Web service capability.
- *Area mediation and orchestration (FR2, FR4, FR5)*: the *Get-Polygon-GIS-data-with-Filter-Goal* is associated with a unique Web service that orchestrates, by simply invoking three sub-goals in sequence. The first gets the list of polygon points from the input; the second is *Get-Circle-GIS-Data-Goal* described above; finally, the third invokes the smart service that filters the list of GIS data. The first two sub-goals are linked by means of three GG-mediators (depicted as ggM) that return the centre, as a latitude and longitude, and radius of the smallest circle which circumscribes the given polygon. To accomplish this, we created three mediation services invoked through: *Polygon-to-Circle-Lat-Goal*, *Polygon-to-Circle-Lon-Goal*, and *Polygon-to-Circle-Rad-Goal* (the related WG-mediator and Web service ovals were omitted to avoid cluttering the diagram). The results of the mediation services and the class of shelter required are provided as inputs to the second sub-goal. A unique GG-mediator connects the output of the second to the input of the third sub-goal. In this instance no mediation service is necessary.

It is important to note that if new WS – for instance providing data from further GIS are available, new Web Service descriptions will be simply introduced, and linked to the *Get-Circle-GIS-Goal* by the proper mediators (even reusing the existing ones, if semantic mismatches do not exist), without affecting the existing structure. In the same way, new GIS filter services (e.g. more efficient ones) may be introduced. The effective workflow – i.e. which services are invoked – is known at run-time only.

4.6 User Interface: usage example

The user interface has been developed using Web standards: XHTML and CSS are used for presentation, JavaScript (i.e. EcmaScript) is used to handle user interaction

and AJAX provides IRS-III goal invocation (*FR1*, *FR3*). One of the main components of the interface is a map, which uses the Google Maps API to display polygons and objects (custom images) at specific coordinates and zoom levels. These objects are displayed in a pop-up window or in a hovering transparent region over the maps.

When the application is launched, a goal is invoked for the Essex region, and snow hazard or storm polygons are drawn according to data from the meteorological office. The value from which snow values can constitute a hazard or a storm are heuristic and as emergency knowledge is gathered it can easily improved, by modifying the smart services which are composed with weather information, while the goal visible to the user remains the same. As an example of practical usage, we describe how an EPO describes and emergency situation, before trying to contact relevant agents. The procedure is as follows:

1. The EPO clicks within the displayed hazard region to bring up a menu of available goals. In this case (Figure 5a) three goals are available: show available shelters, login to BuddySpace and get the presence information for related staff.
2. The EPO asks for the available Rest Centres inside the region, and then inspects the detailed attributes for the Rest Centre returned (Figure 5b).
3. The EPO requests to see the presence status for all staff within the region and then initiates an online discussion the closest online agency worker (Figure 5c).

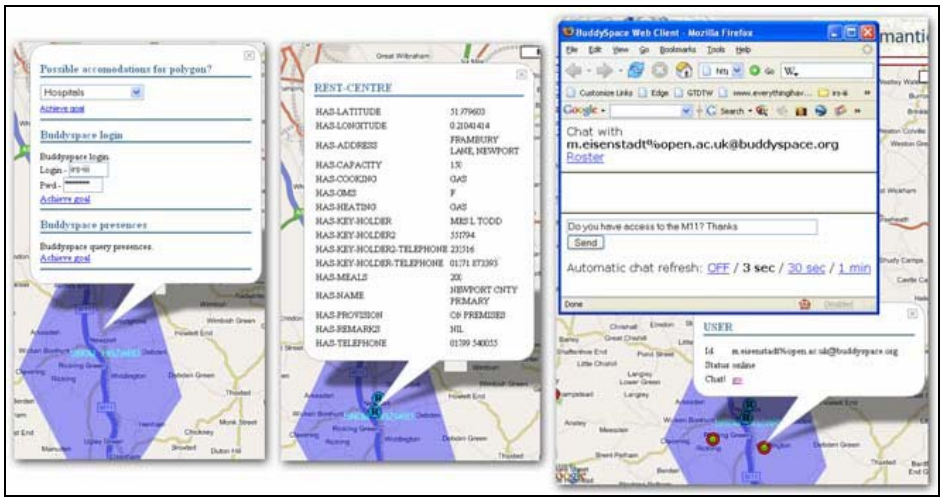


Figure 5 - Three views of the application in use: 5a) Goals available for the snow hazard, 5b) obtaining detailed information for a specific rest centre, 5c) initiating a discussion with an online emergency worker.

5. Related Work and Lesson Learned

Spatial-related data is traditionally managed with the help of GIS, which, by linking spatial algorithms and representation means to spatially extended databases, help supporting decision making by facilitating the integration, storage, querying, analysis,

modeling, reporting, and mapping of this data to analyze possible models. However, each agency tends to collect only data relevant for itself and organizes it in the way that suits it best, managing it according to particular business processes and sharing only what is not judged confidential information. In an emergency situation, such access and semantic barriers are unacceptable and the wish for more complete interoperability through the network is often expressed[2].

Maps available on the web, for identifying an address or getting transportation information, are popular but allow only simple queries. However, recently, a new type of mapping systems has emerged; highly responsive mapping frameworks providing API (Google[4], Yahoo[5], Mapquest[6], etc.). They are also usually enhanced with "reality effects" – e.g. seamless transition between maps, satellite and hybrid views, 2.5-3D visualisations, street level photography, etc. – which make them even more appealing. API allow developers to populate online maps with custom information – location of "events" or "things" –, by collecting data from standard documents such as RDF files, or simply by ad hoc "web scraping" of HTML resources. These embryonic but very agile *Web GIS*, called *mashups*, can merge more than one data sources and add functionality such as filtering and search features. However, although extremely popular, relatively easy to build and to enhance, Web GIS do not avoid traditional issues attached to non semantic applications; indeed (i) handling data heterogeneity still requires considerable manual work, (ii) the lack of semantics limits the precision of queries, and (iii) limited expressiveness usually drastically limits functionality.

Any information system can gain advantage from the use of semantics [14]. In GIS-related application, the use of semantic layers, although not yet firmly established, is being investigated in a number of research studies [11][12][13]. Having ontologies describing a spatial-related data repository and its functionalities is believed to make cooperation with other systems easier and to better match user needs.

In our approach, we adopted WSMO and IRS-III to provide an infrastructure, in which new services can be added, discovered and composed continually, and allow the automatic invocation, composition, mediation, and execution of complex services. The integration of new data sources results relatively simple; the steps involved in the process of adding new data sources can be summarized as follow: (i) ontological description of the service; (ii) lifting operations definition; (iii) mapping to aggregation ontologies; (iv) goal description; (v) mediation description; (vi) lowering definition; and (vii) context linking. Although this procedure may seem tedious, and can actually only be performed by a knowledge expert, it presents many advantages compared to standard based approaches as the one demonstrated in the OWS-3 Initiative[3]:

- *Framework openness*: standards are helpful but not necessary. For example, if querying sensor data, the use of standards – e.g. SensorML[4] – helps the reuse of service ontologies and lifting procedures since they can be applied to any service using a similar schema. However any other schema can be integrated with the same results.

[2] http://www.technewsworld.com/story/33927.html
[3] http://www.opengeospatial.org/initiatives/?iid=162
[4] http://vast.nsstc.uah.edu/SensorML/

- *High level services support*: since services are described as SWS, they inherit all benefits of the underlying SWS execution platform and are updated as more features are added to the platform (e.g. trust based invocation). In other solutions support for composition and discovery is imbedded in syntactic standards themselves, which implies specific parsing features and adding ad hoc reasoning capabilities to standard software applications, which is time consuming and error prone. Moreover, SWS introduce a minimalist approach in the description of a domain, by modeling the concepts used by Web Services only, and allowing on-the-fly creation of instances when Web Services are invoked (lifting).
- *Support of the Emergency Handling Process*: the conceptual distinction between goal and web services - introduced by WSMO – allows developers to easily design business processes known a priori (e.g. emergency procedure) in terms of composition of goals, and move the (automatic) identification of the most suitable service at run-time. Specifically, the constant use of context to link goals and situations greatly enhances the decision process. Indeed, actions are oriented depending on the use case, the object, user role and location. With the help of explanations of the utility of each goal in each context, the Emergency Officer's task is greatly simplified. A future development of the context ontology will include feedback from goal invocation history, and allow workflow definitions, i.e. this goal only appears after these two have been invoked. Note that all goals are also accessible independently of any context which allows non directed queries to occur, if needed.

6. Conclusions and Future Work

In the future, a new era of emergency management can be envisaged, in which EMS's 'collaborate' thorough the Internet to provide relevant information in emergency situations through SWS technology. In this way, involved agencies and emergency corps can extend their knowledge about a particular emergency situation making use of different functionalities based on data hold by other agencies which otherwise might not be accessible to them or slow to obtain.

The proposed EMS is a decision support system based on SWS technology, which assists the EPO in the tasks of retrieving, processing, displaying, and interacting with only emergency relevant information, more quickly and accurately.

In our approach, we aimed to obtain a development process that might be pragmatic - in order to quickly lead to a working outcome – as well as flexible - in order to easily respond to eventually changes/improvements and meet the multiple actors' viewpoints. We followed a prototyping approach that produced two main cycles; a third one is under way.

The first cycle rapidly defined the structure, processes and data sources of the EMS (Section 3) starting from the requirements of a real-world integrated emergency management (Section 2). The result has been valued by stakeholders (emergency planning department in ECC) before advancing with the application development.

The second cycle actualized the required EMS functional requirements (Section 4) by adopting semantic technologies. Specifically, WSMO and IRS-III have been used to implement the SWS infrastructure, which has been linked to the user interface

(based on Google Maps) through AJAX approach. As a result, we obtained a working prototype that has been shown to the EPO's and other people dealing with emergency situations in the ECC area (i.e. potential end-users).

On the basis of their feedback, the third cycle has been planned. Future improvements involve integrating demographic, highways and transport data from ECC. Moreover, we are seeking to use real time data (e.g.: real time RADAR data instead of the weather forecast). Assuming the availability of this data, the system could also be used in the response phase of the designed EMS.

References

1. Essex Resilience Forum (2006). (http://www.essexcc.gov.uk/microsites/essex_resilience/)
2. Combined Operational Procedures For Essex (2006). (http://www.essex.police.uk/pages/about/mipm.pdf)
3. BBC news web site (2006). (http://news.bbc.co.uk/2/hi/talking_point/2711291.stm)
4. A Guide to GIS Applications in Integrated Emergency Management – Cabinet Office - Emergency Planning College (2006). (http://www.epcollege.gov.uk/training/events/gis-guide_acro6.pdf)
5. Eisenstadt, M., Komzak, J., Dzbor, M. Instant messaging + maps = powerful collaboration tools for distance learning. (2003).
6. WSMO Working Group, D2v1.0: Web Service Modeling Ontology (WSMO). WSMO Working Draft, (2004). (http://www.wsmo.org/2004/d2/v1.0/)
7. Cabral, L., Domingue, J., Galizia, S., Gugliotta, A., Norton, B., Tanasescu, V., Pedrinaci, C.: IRS-III: A Broker for Semantic Web Services based Applications. In proceedings of the 5th International Semantic Web Conference (ISWC 2006), Athens, USA (2006).
8. Motta, E.: An Overview of the OCML Modelling Language. (1998).
9. WSDL: Web Services Description Language (WSDL) 1.1, (2001). (http://www.w3.org/TR/2001/NOTE-wsdl-20010315)
10. UDDI Consortium. UDDI specification. (2000). (http://www.uddi.org/)
11. Casati, R., Smith, B., Varzi, A. C.: Ontological tools for geographic representation. (1998) 77–85.
12. Peuquet, D., Smith, B., Brogaard B.: The ontology of fields. (1999).
13. Fonseca, F. T., Egenhofer, M. J.: Ontology-Driven Geographic Information Systems. ACM-GIS (1999) 14-19.
14. Semantic Interoperability Community of Practice (SICoP). Introducing Semantic Technologies and the Vision of the Semantic Web. (2005).

Building Ontology in Public Administration: A Case Study

Graciela Brusa[1], Ma. Laura Caliusco[2], Omar Chiotti[3]

[1] Dirección Provincial de Informática, San Martín 2466
Santa Fe, Santa Fe, Argentina, gracielabrusa@santafe.gov.ar
[2]CIDISI, CONICET-UTN-FRSF, Lavaise 610,
Santa Fe, Santa Fe, Argentina, mcaliusc@frsf.utn.edu.ar
[3]INGAR, CONICET-UTN-FRSF, Avellaneda 3657
Santa Fe, Santa Fe, Argentina, chiotti@ceride.gov.ar

Abstract. The inclusion of Semantic Web technologies into some areas, particularly in the public sector, has not been as expected. That is, among others, because government processes require a large amount of information and its semantic is impossible to carry across organizations. Hence, public servers depend on technical and specifics areas to incorporate knowledge about information that crosses the organization structure government. It succeeds too when government administrations aim at web services and people needs access to semantic of services. In this public services transformation, it is necessary incorporate new tools to be used by community whom this services are addressed. Ontologies are important to share information in internal activities of government administration and to facilitate information access in e-government services. This work presents the experiences during the ontology building in a local public sector: the budgetary and financial system of Santa Fe Province (Argentine). Software engineering techniques were use in manner of minimize the impact of technical knowledge required. At last, architecture is proposed in order to show ontologies applications in government areas and their advantages.

Keywords: ontology, public sector, budgetary and financial system.

1 Introduction

During the last years, an important progress on achieving information interoperability between heterogeneous applications in business sector has been made. Public administrations are facing the same problems than business organizations with respect to information integration. In the public sector, however, the direct replication of the experiences from business sector drives several problems [20], mainly since the complexity of the public sector.

The main difference between the business sector and the public sector is not only the complexity but also the bureaucracy and idiosyncrasy. To comprehend the public sector idiosyncrasy it can be adequate to consider the holistic reference model presented by [26], which, based on a socio-technique approach, makes a consideration of the public sector, showing different views, progress of public services and

abstraction layers. From a technologic point of view, a main government challenge is to get a set of capabilities to facilitate the interoperability, needed for integration as well as the suitable interpretation of information to make decisions.

The interpretation of information without misunderstanding require to make its meaning explicit. To this aim, ontology can be used. Ontology provides a shared vocabulary for common understanding of a domain.

There are several works on how to develop ontologies methodologically. As example can be mentioned: Grüninger and Fox [9], METHONTOLOGY [8][22], and 101 Method [16], among others. These methodologies were successfully used to define ontologies in different domains [4]. Each of them presents different intermediate representations.

Concerning software platforms that aid in ontology development can be mentioned Protégé 3.1, and WebODE [1], among others.

In this paper we present how to develop a budgetary ontology following different development ontology methodologies and using Protégé 3.1. To this aim, the paper is organized as follow. Section 2 describes budgetary and financial system of the Santa Fe Province. Section 3 discusses the tasks carried out to build the budgetary ontology. Section 4 presents the ontology implementation using Protégé 3.1. Section 5 introduces architecture to support information integration using the implemented ontology. Finally, Section 6 presents our conclusions.

2 Budgetary and Financial System: Domain Description

The budget of a government is a plan of the intended revenues and expenditures of that government. The budget is prepared by different entities in different governments. Particularly, in the Santa Fe Province (Argentine) the entities are actors participate:

- Executive Power: it carries out the Provincial Budget Draft. A Rector Organism that conducts all activities and all the Executors Organisms existing in government compounds it that formulates their own budgets.
- Legislative Power: it sanctions the annual budget law.

The interaction among these actors leads different budget states: In Formulation, in Parliamentary Proceeding and Approved. This iterative process is shown in Fig. 1.

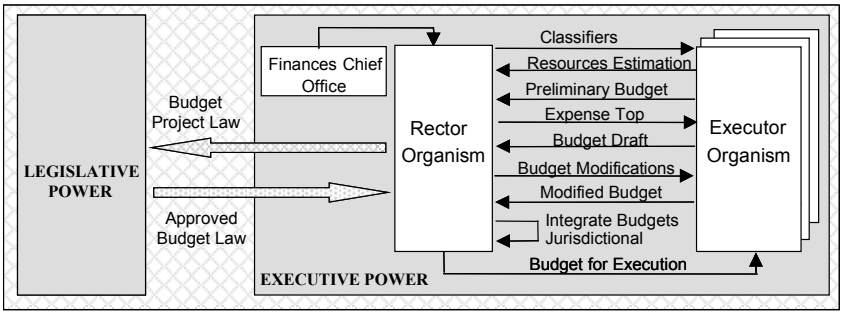


Fig. 1. Iterative process until budget is ready for execution.

In the Executive Power exists a Rector Organism that is responsible for all the budgetary formulation process. This Rector Organism sets the budgetary policies and drives the jurisdictional interactions to complete and integrate its own expenses and resources estimates through this formulation process. Each jurisdiction as Health or Production Ministries has Executor Organism, which are responsible to formulate and execute budget. Formulation process results in the Project of Budget Law issued to Legislative Power for approving.

Local budget life cycle (Fig. 2) is complex because involve a sequence of different instances with a lot of data to each other and a great and specific knowledge is required to operate with them.

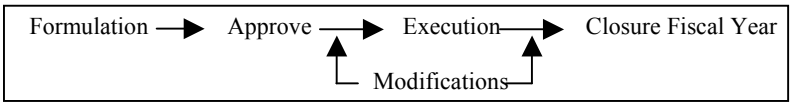


Fig. 2. Local Budget Life Cycle.

Along this life cycle the evaluation and control of actual and financial resources is made, and all of them are assigned to good and services production. Table 1 shows the detail steps.

Table 1. Budget Life Cycle Steps

1.To Initiate Fiscal Year and Distribute Classifiers	8. To Elaborate new budget according to Budget Law
2.To Prepare Preliminary Budget and Resources Estimation	
3. To Define Budgetary Policy and Expenses Projection	9. To Distribute Budget for executing
4. To Determine Expenses Top	10. To Elaborate Budgetary Modifications
5. To Formulate Budget Project Draft	11. To Program Budget executing
6. To Present Budget Project Draft to Legislature	12. To Reconduct Budget
7. To Approve Budget in Legislature	13. To Closure Fiscal Year

There is common information for all budget life cycle stages: Expenses and resources classifiers. They carry over all budgetary life cycle states bringing a thematic classification for its imports. Primary classifiers used in this work are: Institutional, Expense Object, Geographic Locate, Finality Function, Resource Item, Financing Source, and Programmatic Categories.

There are two situations where the availability of semantic information associated to budgetary data is critical: budget formulation and approval tasks. In first case, only government staff with specific knowledge can be involved in this task, concentrating a high responsibility in few persons with much difficult to knowledge transference. In second case, semantics information it is necessary to analyze budgetary data and then to sanction budget law. Here, it is more complex because all the legislators must vote and the majority has not the specific knowledge. For simplicity, the Formulation stage for expenses budget was considered to this study case.

3 Building the Budgetary Ontology

The objective of this section is to discuss the steps we have carried out in order to define an ontology that describes the semantics of the budgetary system domain.

3.1 A Methodology Selection

Before starting to define the ontology, different development methodologies were studied [5][14][24]. From this study, could be identified two main groups. On the one hand, there are experience-based methodologies, such as the methodology proposed for Grüninger y Fox [9], based in TOVE Project or the other exposed by Uschold y King [21] [24] from Enterprise Model. Both issue in 1995 and both belong to the enterprise modeler domain. On the other hand, there are methodologies that propose evolutive prototypes models, such us METHONTOLOGY [8] that proposes a set of activities to develop ontologies based on its life cycle and the prototype refinement; and 101 Method [16] that proposes an iterative approach to ontology development.

There is no one correct way or methodology for developing ontologies. Usually, the first ones are more appropriate when purposes and requirements of the ontology are clear, the second one is more useful when the environment is dynamic and difficult to understand exists [5]. Moreover, it is common to merge different methodologies since each of them provides design ideas that distinguish it from the rest. This merging depends on the ontology users and ontology goals.

At this work, both approaches were merged because in one hand, requirements core are clear but in the other, domain complexity drives to adopt an iterative approach to manage refinement and extensibility.

In general, the ontology development can be divided into two main phases: specification and conceptualization. The goal of the specification phase is to acquire knowledge about the domain. The goal of the conceptualization phase is to organize and structure this knowledge using external representations that are independent of the implementation languages and environments. In order to define the ontology for the budget domain we have followed the 101 Method (OD101) guides for creating a first ontology [16] and used the analysis steps from METHONTOLOGY in the conceptualization process. Both consider an incremental construction that allows refining the original model in successive steps and they offer different representations for the conceptualization task.

3.2. Specification: Goal and Scope of the Ontology

The scope limits the ontology, specifying what must be included and what must not. In OD101, this task is proposed in a later step but we considered appropriate to include it at this point for minimizing the amount of data and concepts to be analyzed, especially for the extent and complexity of the budgetary semantic. In successive iterations for verification process, it will be adjusted if necessary.

This ontology only considers the needs to elaborate a project of budget law with concepts related to expenses. It is a first prototype and then, it does not consider the

concepts related to other stages as budgetary executing, accounting, payments, purchases or fiscal year closure. Therefore, it includes general concepts for the budget life cycle and specifics concepts for the formulation.

3.3. Specification: Domain Description

Taking into account that this work was made from scratch it was necessary to make a previous domain analysis. In this analysis, the application for formulating the provincial budget and its related documentations were studied and revised. Furthermore, meetings with a group of experts were carried out. This group was conformed by public officials responsible for whole budget formulation process in Executive Power, expert professionals of Budget Committee in Legislative Power, public agents of administrative area taking charge of elaborate own budget and software engineers whom bring informatics supports for these tasks.

3.4. Specification: Motivating Scenarios and Competence Questions

We included this step taking into account the opinion of Gruninger y Fox [9] who considers that for modeling ontologies, it is necessary an informal logic knowledge model in addition to requirements resulting from different sceneries. The motivation scenerios show problems that arise when people needs information that the system does not provide. Besides, it contains a set of solutions to these problems in which the semantic aspects to resolve them are. In order to define motivation scenarios and communicate them to involved people, templates have been used. These templates were based on those proposed to specify case uses in object oriented methodology [23]. An example is shown in Table 2. In this template, the main semantic problems and a list of key terms were included.

Table 2. Scenario description.

SCENARIO N°	**1**	
NAME	**Local Budget Formulation**	
DESCRIPTION	**Necessary tasks to estimate expenses for next year, which will be integrate with the other government jurisdictions for elaborating Draft Local Budget. ……**	
SITE	**Executor Organism of a Jurisdiction**	
ACTORS	▪ **Public agents uncharged jurisdictional budget** ▪ **Rector Organism agents** ▪ **Public agents from areas of a jurisdiction**	
PRE-REQUIREMENTS	▪ **Budgetary Policy defined** ▪ **Expenses Classifiers received from Rector Organism** ▪ **Reference documentation**	
ASSOCIATES REQUIREMENTS	▪ **Prepared agents in Budget Formulation tasks.** ▪ **Advisory agents from Rector Organism**	
NORMAL SEQUENCE	**STEP**	**ACTION**
	1	**To receive expenses estimations from jurisdiction areas**
	2	**To bring support to this areas for elaborating own expenses programs.**
	3	**To integrate all expenses programs for jurisdiction.**
	4	**To create Programming Categories and send it to Rector Organism**
	5	**To create the Jurisdictional Budget Project**

	6	To load budget in informatics system and send it to Rector Organism
	7	To receive approved jurisdictional budget from Rector Organism
POST-CONDITION	▪ Jurisdictional Expenses Budget Project ▪ Jurisdictional Programmatic categories	
EXCEPTIONS	STEP	ACTION
	5	To consult the Rector Organism if it is not understands different aspects to formulate budget.
	7	To modify budget if it is not approved
PERMANENT TASKS	▪ To interact with Rector Organism to clarify the knowledge of conceptual domain aspects ▪ To bring support to different areas of jurisdiction	
MAIN PROBLEMS	▪ A lot of time loosed in clarify conceptual doubts ▪ Great problems when an agent must be replaced in key places of work. ▪ The whole process is highly dependent of few persons knowledge. ▪ …….	
MAIN TERMS	Budgetary classifier, expense a classifier, Institutional, Programmatic Category, Geographic, Expenses Object, Financing Source and Finality Function Classifiers, among others, for working into the budget draft task.	

The competency questions proceed from motivation sceneries, allowing deciding the ontology scope, to verify if it contains enough information to answer these questions and to specify the detail level required for the responses. Besides, it defines expressivity requirements for the ontology because it must be able to give answers using its own terms, axioms and definitions. The scope must define all the knowledge that should be in the ontology as well as those that must not be. It means that a concept must not be included if there are not competency questions to use its. This rule is also used to determine if an axiom in ontology must be included or not.

Moreover, the competency questions allow defining a hierarchy so that an answer response to a question may also reply to others with more general scope by means of composition and decomposition processes. Table 3 shows some of them.

Table 3. Samples of Competency Questions

Simple Questions	Complex Questions
Which are budget states?	Which is the institutional code for Department of Labor?
Which are budgetary classifiers?	Which are sector and subsector for Central Administration?
Which are expenses classifiers?	What is the character code for "Decentralized Organism"?
Which are resources classifiers?	Which properties have an Institution?
Which are the executor organisms for Health Minister?	Which is the institutional code for "Pharmacological Producer Laboratory" SAF?

3.5. Specification: Ontology Granularity and Type

According to purpose and level of granularity [8], the ontology proposed here was defined as a domain ontology. Domain ontology describes the vocabulary related to a specific domain. In this case study the ontology describe the budgetary domain of the Santa Fe Province. And, the ontology objective is to facilitate the communication between central administration staff that must deal with the local budget, bringing adequate terminology to non-expert users.

The term ontology can be used to describe models with different degrees of structure. Particularly, the ontology defined in this paper is a formal structure expressed in artificial formally defined languages.

3.6. Conceptualization: Conceptual Domain Model Determination

In this step, a list of more important terms was elaborated according to OD101. To this aim, the middle-out strategy [19] was used. With this strategy, the core of basic terms is identified first and then they are specified and generalized if necessary.

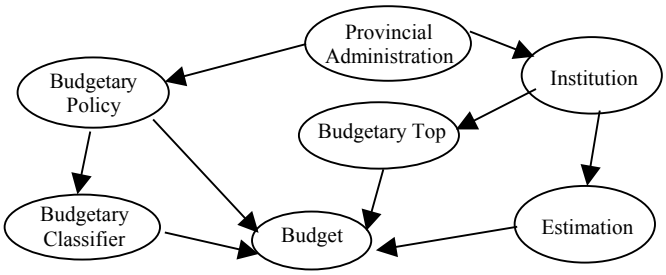


Fig. 3. Basic terms of the budget domain.

Then with these concepts as reference, the key term list was defined. List shown in Table 4 does not include partial or total overlapping of concepts, synonyms, properties, relations and attributes.

Table 4. Key Terms

Activity	Expense	Subpartial Item
Budget	Expenses Classifier	Subprogram
Budget Analytic	Expense Object	Program Executer Unit (UEP)
Budget Approved	Finality Function	Programmatic Category
Budget Project Draft	Financial Administration	Project
Budget Synthetic	Financing Source	Public Funds Administrative Service (SAFOP)
Budget States	Geographic Locate	
Budgetary Classifier	Institutional	Rector Organism
Budgetary Fiscal Year	Institution	Resource
Budgetary Policy	Jurisdiction	Resources Estimation
Budgetary Top		Year Financial Administrative Service (SAF)
Executor Organism	Program	

To properly understand the conceptual aspects in the context, we elaborated a Unified Modeling Language (UML) diagram [23], (Fig. 4), with the main relations among defined concepts. UML is a useful tool for ontology modeling though it was not designed for this task [3].

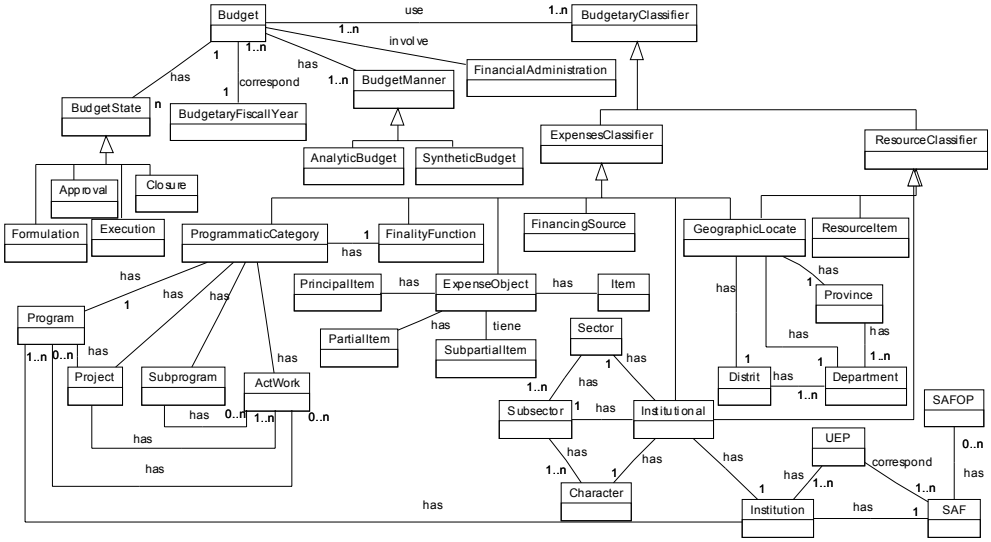


Fig. 4. Domain Model in UML.

This information was the base for building the ontology term glossary, trying to include other concepts by means of generalization and specialization techniques. The conflictive assertions over the same entity may be discovered if the concepts are described as completely as possible [12], to this aim, definitions were made as complete as possible to contribute to define rules and axioms.

This UML model was useful to verify the ontology scope and to take an important design decision: working with two ontologies. One of them is the Domain Ontology that contains the general concepts for the budget life cycle and the other, Formulation Ontology, contains the semantic specific for formulating it. This is task ontology [10] since it defines concepts related to a specific task, the budget formulation. So, we have to modify the list of key terms, hierarchical relations, and to group competency questions depending on the ontology concepts they were related with. As Guarino sets [10], it exists ontology types accord with dependence level of task or viewpoint. Hence, ontologies construction implies two different strategies [6]. In one hand, a domain ontology with an application-independent strategy because its general concepts must be available all time. In other hand, task ontology is application-semidependent because different use scenerios can be identified and its conceptualization is associated to real activities.

Working with different ontologies allows the term reusability and usability. These concepts are important goals in ontologies construction [13] and differ finely. While reusability implies to maximize the ontology use among different task types, usability maximizes the number of different applications using the same ontology. From now,

the work is concentrated on Domain Ontology development. This Domain Ontology will be able for using in all budget states facilitating term reusability.

3.7. Conceptualization: Identification of Classes, Relations and Attributes

At this step, we considered OD101 recommendations. Besides, we used representations proposed by METHONTOLOGY to knowledge organization as concepts classifier trees (Fig. 5) to analyze hierarchies and attributes, binary relations, axioms and instances tables. For determining classes, we identified those terms of independent existence from the key terms list and the glossary.

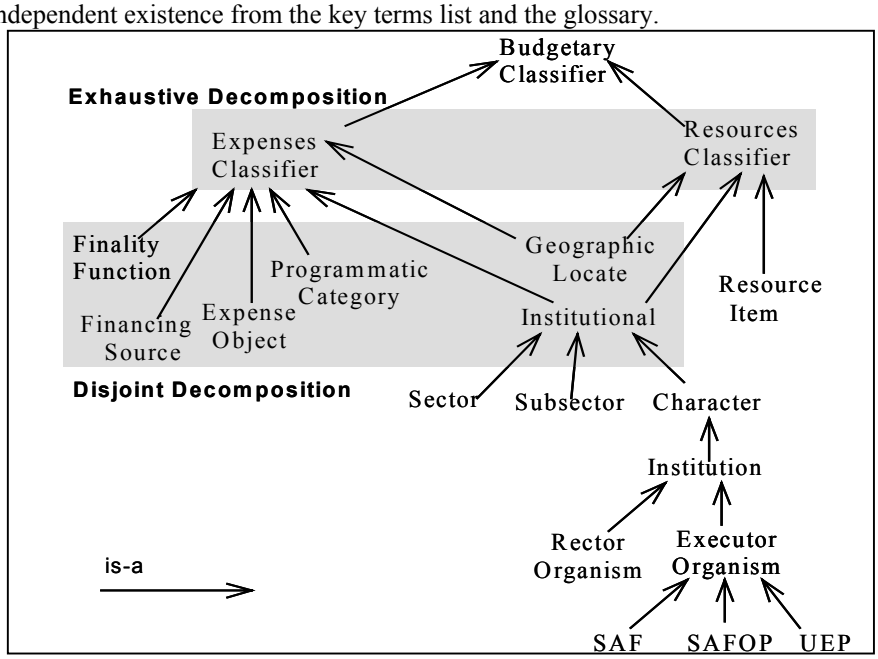


Fig. 5. Taxonomy of Budgetary Ontology Concepts.

Disjoint classes, exhaustive decompositions and partitions [12] may be identified in these graphic representations. A Disjoint-Decomposition of a concept C is a set of subclasses of C that do not have common instances and do not cover C, that is, there can be instances of the concept C that are not instances of any of the concepts in the decomposition. As example (see Fig. 5), Finality Function, Financing Source, Expense Object, Programmatic Category, Geographic Locate and Institutional can be mentioned as disjoints. An Exhaustive-Decomposition of a concept C is a set of subclasses of C that cover C and may have common instances and subclasses, that is, there cannot be instances of the concept C that are not instances of at least one of the concepts in the decomposition. For example (see Fig. 5), the concepts Expenses Classifier and Resource Classifier make up an exhaustive decomposition of the concept Budgetary Classifier because there are no classifier that are not instances of at least one of those concepts, and those concepts can have common instances. A Partition of a concept C is a set of subclasses of C that do not share common instances

and that cover C, that is, there are not instances of C that are not instances of one of the concepts in the partition. In this scenario there are no partitions.

It is always convenient to begin with primitive classes, examining which of them are disjoint and verifying if that condition does not produce instances absents.

Once the hierarchies and their features have been identified a table to reflect bidirectional relations may be elaborated by means of assigning names with an uniform criteria. An example is shown in Table 5. Shades rows are self-evident relations between concepts shown in the Concepts Classifier Tree (see Fig. 5) that it results bidirectional relation after analyzing them.

Table 5. Bidirectional Relations

CONCEPT	RELATION	CARDINALITY	CONCEPT	INVERSE RELATION
Institutional	inst-include-sec	1	Sector	sec-isPartOf-Inst
Institutional	inst-include-sbsec	1	Subsector	sbsec-isPartOf-Inst
Institutional	inst-include-char	1	Character	Char-isPartOf-Inst
Sector	sec-isPartOf-Inst	1,n	Institutional	inst-include-sec
Subsector	sbsec-isPartOf-Inst	1,n	Institutional	inst-include-sbsec
Character	char-isPartOf-Inst	1,n	Institutional	inst-include-char
Character	char-has-Inst	1,n	Institution	inst-correspond-char
Institution	ins-has-SAF	1	SAF	SAF-correspond-inst

The relation direction depends on competence questions to be solved and the possible conflicts with other defined classes restrictions. A restriction list identifies those necessary and sufficient conditions and those only necessary to work later in their formalization. We individually analyzed the axioms but also in a group of classes to verify if closure restrictions are required.

3.8 Conceptualization: Instances Definition

Once the conceptual model of the ontology has been created, the next step is to define relevant instances inside an instance table. According to METHONTOLOGY, for each instance should be defined: its name, the name of the concept it belongs to, and its attribute values, if known, as Table 6 shows.

Table 6. An excerpt of the Instance Table of the Budgetary Ontology.

CONCEPT NAME	INSTANCE NAME	PROPERTY	VALUE
Institutional	Institutional_111	cod-institutional	1.1.1
		has-fiscal-year	2004
		inst-include-sec	1-No Financial Local Public Sector
		inst-include-sbsec	1- Local Administration
		inst-include-char	1- Main Administration
	Institutional_212	cod-institutional	2.1.2
		has-fiscal-year	2004
		inst-include-sec	2-Financial Local Public Sector
		inst-include-sbsec	1-Offcial Banking System
		inst-include-char	2- Official Banks

4 Implementing the Budget Ontology with PROTÉGÉ 3.1

In order to implement the ontology, we chosen Protégé 3.1 due to it is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development. Protégé ontologies can be exported into different formats including RDF Schema (RDFS) [2] and Web Ontology Language (OWL) [19].

Particularly, we have implemented the Budgetary Ontology in OWL. The first challenge during this task was how to transform the UML diagram from conceptualization phase into the OWL formalism. This task was hard and time consuming. Modeling in OWL implied to transform composition relations into bidirectional relations. In addition, some concepts modeled as classes in UML were properties in ontology. And not all relations in UML were modeled in ontology but only those relations that were necessary to answer competence questions. Moreover, the granularity of domain ontology is coarse and it was adequate select a flat structure for its.

Then, we verified the ontology by using Racer [11]. During the verification process, we have taken into account experiences of CO-ODE Project [15] and [17]. We verified consistency validation and classification. During process for charging classes and attributes, the verification was incremental and continuous to avoid future propagation errors. When a class is unsatisfiable, Racer shows it with a red bordered icon and there are different categories of causes [25] and can be exists propagated errors. At this point is very important how are classes defined (disjoint, isSubclassOf, Partial Class, Defined Class, etc.) and their restrictions (unionOf, allValuesFrom, etc.). Classification process can be invoked either for the whole ontology, or for selected subtrees only. When the test is over whole ontology, errors were searched beginning with minor level class in the hierarchy for minimizing propagation errors.

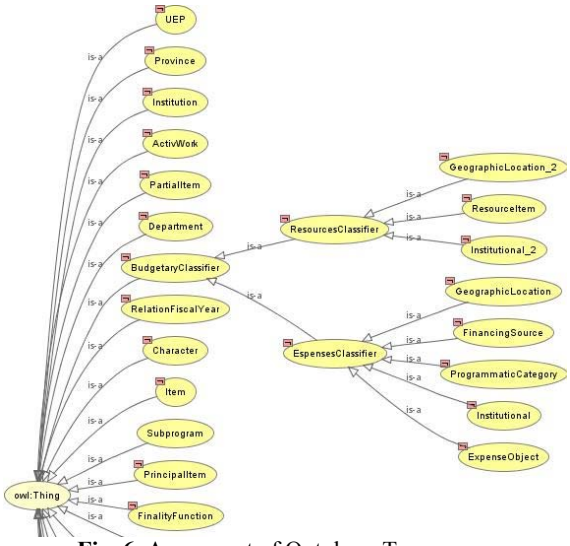


Fig. 6. An excerpt of Ontology Taxonomy.

To compare the ontology implementation with its conceptualization, graphics by using the OWLViz and Ontoviz plug-ins were generated and compared with UML diagrams. Fig. 6 shows an excerpt of the General Ontology taxonomy.

On the one hand, OWLViz enables the class hierarchies in OWL Ontology to be viewed, allowing comparison of the asserted class hierarchy and the inferred class hierarchy. With OWLViz primitive and defined classes can be distinguished, computed changes to the class hierarchy may be clearly seen, and inconsistent concepts are highlighted in red. In the taxonomy shown here, can be seen how to represent a multiple inheritance with twice terms defined for Location Geographic. Another form is to use axioms and lets to reasoner generates inferred classes.

On the other hand, OntoViz generates a graphics with all relations defined in the ontology instances and attributes. It permits visualizing several disconnected graphs at once. These graphs are suitable for presentation purposes, as they tend to be of good clarity with none overlapping nodes. Besides, these graphics are very useful for analyzing when a concept must be modeled as a class and when must be modeled as an attribute. An example of them is shown in Fig. 7.

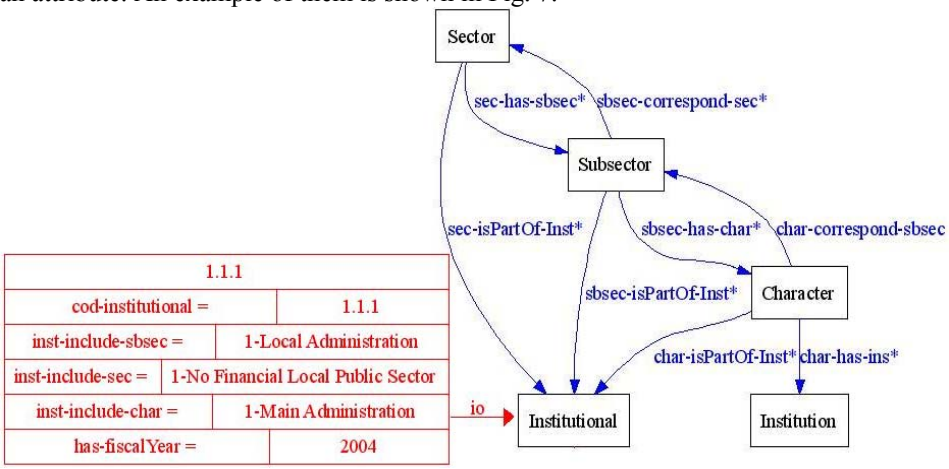


Fig. 7. Main Relations Between Concepts of Institutional Classifier.

4.1 Ontology Querying

In order to verify and validate the ontology regards to competency questions, we used the RDF Data Query Language (RDQL) [18]. RDQL is an implementation of an SQL-like query language for RDF. It treats RDF as data and provides query with triple patterns and constraints over a single RDF model. Another query language is OWL-QL [7], which was designed for query-answering dialogues among agents using knowledge in OWL. Then, OWL-QL is suitable when it is necessary to carried out an inference in the query. This is not the case of the major competency questions, then, RDQL is enough. To implement these queries we used the Jena framework, which provides an API for creating and manipulating RDF models.

Following the RDQL query that models the competency question “What are sector and subsector for Main Administration?” is shown.

```
SELECT ?x ?y ?z ?nsec ?nsbsec
WHERE (x,<adm:rdfsec-hassbsec>,?y)
       (?y,<adm:rdfsbsec-has-char>,?z)
       (?z,<rdfn:label>, '1-Main Administration')
       (?x,<rdfn:label>, ?nsec ),
       (?y,<rdfn:label>, ?nsbsec )
USING rdfn FOR http://www.w3.org/2000/01/rdf-schema#
      adm FOR http://protege.stanford.edu/
```

5 Using the Budget Ontology

The main ontology goal is to provide a mechanism for information sharing between people and applications without misunderstanding, independent of its implementation. Then, the final step to achieve the ontology goal is to design and implement architecture as one shown in Fig. 8. The architecture components are described next.

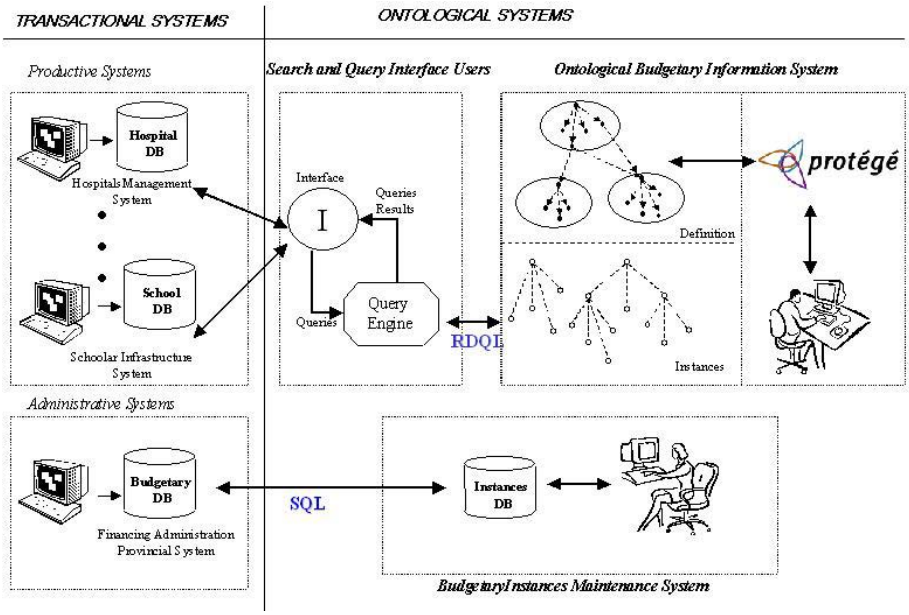


Fig. 8. Ontology-based Architecture for Budget Content Integration in Public Sector.

Ontological Budgetary Information System: Ontology designer team carry out the design, implementation and maintenance tasks using Protégé. This architecture proposes a general ontology for the common concepts for all the budgetary life cycle and specific ontologies for each stage of budget as formulation, approve, execution and closure.

Budgetary Instances Maintenance: expert persons realize the maintenance of instances for general and specific ontologies, requiring the necessary adjusts to ontological designer through the interaction with budgetary system and users.

Search and Query Interface Users: receive queries and return results of them through a friendly interface user. Applications or persons can issue queries through this interface that it uses RDQL as query language support.

Transactional Systems: both administrative and productive government systems. In this study case, a productive system as Hospitals or School Infrastructure Administrative System can access simply to budgetary information for own each interests through Ontological Systems.

6 Conclusions

In this paper, we have shown how domain experts in public sector can develop their own ontologies merging two different methodologies and software engineering techniques taking advantages of them. Particularly, this approach has been used to define General Ontology for a Budgetary and Financial System, which could be extended by Task Ontologies and used by different government applications.

The main conclusions that can be transmitted to the reader are:

- To assign all the necessary time to do a good conceptual analysis because it is considered the most important task during development ontology.
- To modularize the ontology while it is possible for giving it more flexibility and permitting extensibility and reuse. It can be made through relations and attributes observation of conceptual aspects involved.
- To take into account that there are steps that can be applied during the development of any ontology whereas there are steps that are domain-dependent.
- To realize a permanent and iterative validation process, taking into account that partial verifications allow to identify errors propagation between sets of classes.
- To define graphics to transmit the domain conceptualization to the domain experts. Some software engineering techniques that could be familiar for the domain experts, such as UML, can be useful.
- To consider for development who is maintenance responsible expert user and it anticipates a friendly interface user.

References

1. Arpírez JC, Corcho O, Fernández-López M, Gómez-Pérez A (2003) WebODE in a nutshell. AI Magazine, 24(3)-37-47.
2. Brickley, D., Guha, R.V. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004. http://www.w3.org/TR/rdf-schema/
3. Caliusco M. L., A Semantic Definition Support of Electronic Business Documents in e–Colaboration. PhD Thesis. (Universidad Tecnológica Nacional, F.R.S.F., 2005.
4. Corcho O, Fernández-López M, Gómez-Pérez A, López-Cima A. Building legal ontologies with METHONTOLOGY and WebODE. Law and the Semantic Web. Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications. March 2005.
5. Cristani M. and Cuel R. Methodologies for the Semantic Web: state-of-the-art of ontology methodology. SIGSEMIS Bulletin. SW Challenges for KM. July 2004 Vol. 1 Issue 2.
6. Fernandez Lopez M. (1999) Overview of methodologies for building ontologies. In: Benjamins VR, editor. IJCAI-99 Workshop on Ontologies and Problem-Solving Methods; Stockholm, Sweden: CEUR Publications: 1999.

7. Fikes, R., Hayes, P., Horrocks, I., OWL-QL - A Language for Deductive Query Answering on the Semantic Web. KL Laboratory, Stanford University, Stanford, CA, 2003.
8. Gómez-Pérez A., Fernández López M. and Corcho O. Ontological Engineering with examples from the areas of knowledge management, e-commerce and the semantic web. London: Springer, 2004.
9. Gruninger M. and Fox M. S., Methodology for the Design and Evaluation of Ontologies, IJCAI Workshop on Basic Ontological in Knowledge Sharing, Montreal, Canada, 1995.
10. Guarino N. (1998) Formal Ontology and Information Systems. In Proceedings of FOIS'98, Trento, Italy. Amsterdam, IOS Press.
11. Haarslev V. and Möller R. 2001. RACER System Description. In Proceedings of the First international Joint Conference on Automated Reasoning. IJCAR, June 2001.
12. Horridge M., Knublauch H., Rector A., Stevens R., Wroe C., A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0, The University Of Manchester Stanford University, August 27, 2004.
13. Jarrar M., Towards Methodological Principles for Ontology Engineering. PhD Thesis, Vrije Universiteit Brusell, 2005.
14. Jones D., Bench-Capon T. y Visser P., Methodologies for Ontology Development, en Proc. IT&KNOWS Conference, XV IFIP World Computer Congress, Budapest, August 1998.
15. Knublauch H., Horridge M., Musen M., Rector A., Stevens R., Drummond N., Lord P., Noy N., Seidenberg J., Wang H., The Protégé OWL Experience, Workshop on OWL: Experiences and Directions, Fourth International Semantic Web Conference (ISWC2005), Galway, Ireland, 2005.
16. Noy, N., McGuinness D., Ontology Development 101: A Guide to Creating Your First Ontology, 2001.
17. Rector A., Drummond N., Horridge M., Rogers J., Knublauch H., Stevens R., Wang H., Wroe C., OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns.
18. Seaborne A., RDQL - A Query Language for RDF, W3C Member Submission 9 January 2004, http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/
19. Smith M., Welty C., McGuinness D., OWL Web Ontology Language Guide, W3C Recommendation 10 February 2004, http://www.w3.org/TR/owl-guide/
20. Traunmüller R., Wimmer M., Feature Requirements for KM in Public Administration, 2002, http://www.lri.jur.uva.nl/~winkels/eGov2002/Traunmuller.pdf
21. Uschold, M., Building Ontologies: Towards a Unified Methodology, 16th Annual Conference of the British Computer Society Specialists Group on Expert Systems, Cambrige, UK, 16-18 December 1996.
22. Uschold, M., Gruninger M., Ontologies: Principles, Methods and Applications, Knowledge Engineering Review, 1996.
23. Unified Modeling Language. http://www.uml.org/
24. Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner S., Ontology-Based Integration of Information –A Survey of Existing Approaches, in Proc. IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, pp. 108-117, 2001.
25. Wang H., Horridge M., Rector A., Drumond N., Seidenberg J. (2005) Debugging OWL-DL Ontologies: A Heuristic Approach. 4th International Semantic Web Conference (ISWC'05), Galway, Ireland.

Personalized Question Answering: A Use Case for Business Analysis

VinhTuan Thai[1], Sean O'Riain[2], Brian Davis[1], and David O'Sullivan[1]

[1] Digital Enterprise Research Institute,
National University of Ireland, Galway, Ireland
{VinhTuan.Thai,Brian.Davis,David.OSullivan}@deri.org
[2] Semantic Infrastructure Research Group,
Hewlett-Packard, Galway, Ireland
sean.oriain@hp.com

Abstract. In this paper, we introduce the Personalized Question Answering framework, which aims at addressing certain limitations of existing domain specific Question Answering systems. Current development efforts are ongoing to apply this framework to a use case within the domain of Business Analysis, highlighting the important role of domain specific semantics. Current research indicates that the inclusion of domain semantics helps to resolve the ambiguity problem and furthermore improves recall for retrieving relevant passages.

Key words: Question Answering, Business Analysis, Semantic Technology, Human Language Technologies, Information Retrieval, Information Extraction

1 Question Answering Overview

Question answering (QA) research originated in the 1960s with the appearance of domain specific QA systems, such as BASEBALL which targeted the American baseball games domain and LUNAR which in turn focused on the lunar rock domain [1]. These early systems were concerned with answering questions posed in natural language, against a structured knowledge base of a specific domain [1]. They are commonly known as natural language front ends to databases. Research within this field remains active with the introduction of new approaches and techniques to enhance QA performance in more real-life, complex settings (e.g. [2–4]).

With the advent of Semantic Web technologies, domain specific knowledge can also be encoded within formal domain ontologies. This in turn has motivated the growth of another branch of QA research; focusing on answering natural language questions against formal domain ontologies. Attempto Controlled English (ACE) [5] and AquaLog [6] are recent research additions to this area. Rather than translating natural language questions into SQL statements, these systems translate them into variants of first-order predicate logical such as Discourse Representation Structure (DRS) in the context of ACE and Query-Triples for

AquaLog respectively. Consequently this permits answer derivation as an outcome of the unification process of both question and knowledge-based logical statements.

The application of domain specific QA extends further than systems that have their knowledge sources completely encoded in relational database or formal ontologies. Many research initiatives have investigated the use of domain ontologies or thesaurus in assisting finding answers for questions against a small volume of collections of unstructured texts contained within terminology-rich documents. A variety of approaches have been proposed, ranging from a pure Vector Space Model based on traditional Information Retrieval research, extending this approach a with domain specific thesaurus in [7], to a template-based approach for medical domain systems [8, 9], or a computational intensive approach in [10], the goal being to convert both the knowledge source and the question into Minimal Logical Form.

Apart from domain specific QA research, the introduction of the QA track at Text Retrieval Conference TREC-8 in 1999 involved researchers focusing on combining tools and techniques from research fields such as Natural Language Processing, Information Retrieval, and Information Extraction in an attempt to solve the QA problem in open-domain setting: the main knowledge source being a predefined large newswire text corpus, with the World Wide Web acting as an auxiliary source of information. The questions being asked consist mainly of: factoid questions, list questions, definition questions and most recently, the relationship task type question [11]. A review of participating systems in TREC QA track is beyond the scope of this paper. Interested readers are referred to [12] for further details. Of crucial importance however is that the existing QA track does not target domain specific knowledge.

QA, in itself, remains an open problem. The research overview has highlighted the diversity of QA systems under development. Each system is designed to address the problem in a particular usage scenario, which imposes certain constraints on available resources and feasible techniques. Nevertheless, there remain usage scenarios for QA systems that require addressing, one of which is Personalized Question Answering. We discuss our motivation for using Personalized Question Answering in Section 2.

The remainder of this paper is structured as follows: Section 3 describes our use case of Personalized QA within the Business Analysis domain. Section 4 presents a proposed framework for Personalized QA; Section 5 concludes this paper, reaffirms our goals and identifies future work.

2 Personalized Question Answering

Our motivation towards Personalized Question Answering stems from existing shortcomings within current QA systems designed for extracting/retrieving information from unstructured texts. The shortcomings are categorized below:

Authoritative source of information: In an open-domain QA setting, end-users have little control over the source of information from which answers are sought. The reliability of answers is based mostly on the redundancy of data present on the WWW [13]. Similarly, existing domain specific QA systems also limit the source of information to a designated collection of documents. To our knowledge, no QA system is designed in such a way that allows end-users to flexibly specify the source of information from which the answers are to be found. This is of importance with respect to the design of a QA solution. For the majority of existing work, the collection of documents must initially undergo pre-processing. This pre-processing is performed only once, the results being stored later for retrieval. This offline processing strategy makes a computational intensive approach (such as in [14, 10]), feasible because all the necessary processing is already performed offline before any questions can be asked, and therefore reduces significantly the time required to find answers at run time. A QA system that has a dynamic knowledge source will therefore need to take this amount of necessary processing into consideration.

Contextual information: The use of contextual information in QA has not received adequate attention yet. Only the work of Chung et al. [3] highlights the fact that while forming the question, users may omit important facts that are necessary to find the correct answer. User profile information is used in their work to augment the original question with relevant information. The design of QA systems therefore needs to take into account how and to what degree a given question can be expanded to adequately reflect the context in which it is asked.

Writing style of documents: Current domain specific QA systems are usually targeted to scientific domains, with the knowledge source, such as technical, medical and scientific texts [8, 14, 7], written in a straight-forward, declarative manner. This characteristic reduces the ambiguity in these texts. However, this is not always the case with other types of documents, for example business reports. Therefore, QA system should be able to utilize the domain and/or personal knowledge to resolve ambiguity in texts that are written in a rhetorical way.

Targeting to address the above limitations, we propose Personalized Question Answering, which:

- is domain specific, therefore avails of a formal domain ontology
- can cope with dynamic collection of unstructured texts written in rhetorical style
- can handle various question types
- resolves implicit context within questions
- provides an answer-containing chunk of texts rather than the precise answer

Before discussing details of the proposed framework, we first outline in Section 3, a use case for Personalized QA within the domain of Business Analysis.

3 Business Analysis use case

Business Analysis is largely performed as a Business Intelligence[3] (BI) activity with data mining and warehousing providing the information source for monitoring, identification and gathering of information. On-line analytical processing then allows differing data views and report generation possible from which further BI analysis may then be performed. Data excluded from the extract, transform and load phase passes through the process unaltered as unstructured information. Subsequent mining efforts on this information compound the problem by their reliance upon problematic document level technologies such as string-based searching resulting in information being missed.

It is this type of mining activity that enterprises performing customer analysis as a means to identify new business opportunities currently rely upon. The problem becomes more complex when it is considered that business analysts in performing company health checks depend largely upon the consolidated financial information and management statements found in the free text areas of the Form 10-Q. Management statements are those from the companies' CEO and are concerned with the companies' performance. They are viewed as a promotional medium for presentation of corporate image and are important in building credibility and investor confidence. Despite analysts having a clear understanding of the information content that the statements may contain, the searching, identification and extraction of relevant information remains a resource intensive activity.

Current BI technologies remain limited in this type of identification and extraction activities when processing information contained in unstructured texts written in a rhetorical manner. For example, part of performing a company health check involves building an understanding of that company's sales situation mentioned in Form 10-Q[4]. Sales[5] performance is in turn partially dependent upon product and services revenue. An intuitive and time-saving way to gain understanding on these areas is to pose in natural language non-trivial questions such as "*What is the strategy to increase revenues?*", "*Are there are plans to reduce the cost of sales versus revenue?*" and retrieve chunks of text that contain answers to these questions.

Our Personalized QA framework is based upon semantic technology and when applied to the Business Analysis domain, will offer business analysts the ability to expedite the customer analysis process by having potentially relevant information presented in a timely manner. This can be achieved by having the business analysts associate their knowledge in the form of a formal ontology and a set of

[3] Term introduced by Gartner in the 1980s that refers to the user-centered process of data gathering and analysis for the purpose of developing insights and understanding leading to improved and informed decision making.

[4] Quarterly report filed to the Securities and Exchange Commission (SEC) in the US. It includes un-audited financial statements and provides a view of the company's financial position. Relied upon by Investors and financial professionals when evaluating investment opportunities

[5] Discussion context is the software services domain

domain specific synonyms to the QA system, specify the source document (Form 10-Q), pose their questions, and retrieve chunks of text that contain answers to conduct further analysis. The framework is described in the next section.

4 Personalized Question Answering Framework

The proposed framework for Personalized QA, as shown in Fig. 1, consists of two main modules: Passage Retrieval and Answer Extraction. The Passage Retrieval module performs text processing of documents and analysis of questions on-the-fly to identify passages that are relevant to the input question. This coarse-grained processing reduces the search space for the answer significantly as only relevant passages are fed to Answer Extraction (which is a more computationally intensive module), to perform further fine-grained processing to identify chunks of texts containing the correct answer. The details of these modules are discussed below.

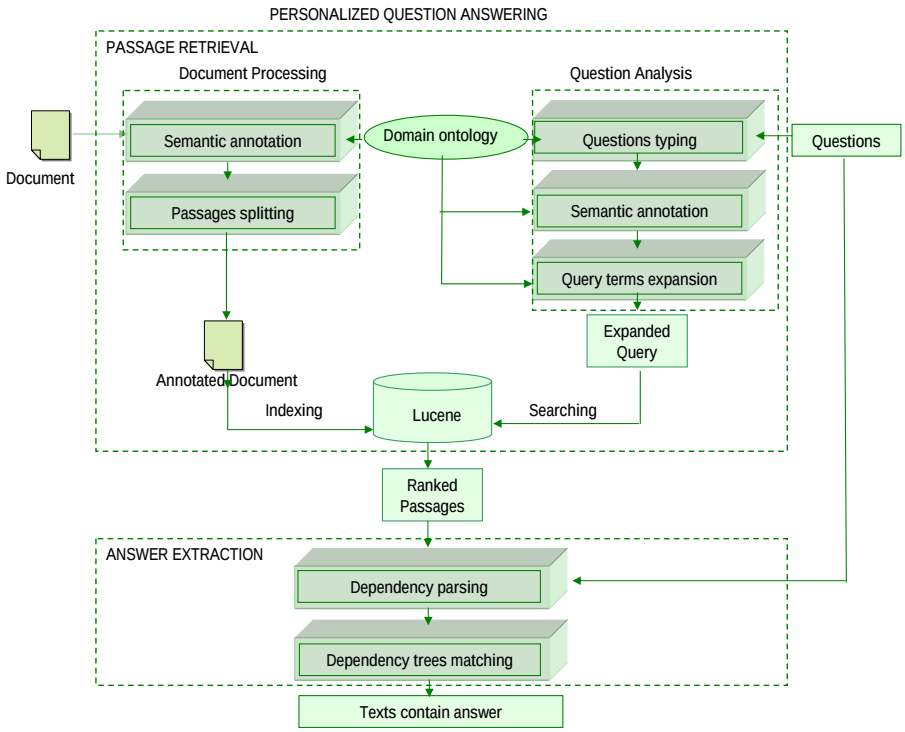


Fig. 1. Personalized Question Answering Framework

4.1 Passage Retrieval

Passage Retrieval serves as an important module in the whole QA process. If this module cannot locate any passage that possibly contains the answer when one actually exists, an answer cannot be found. On the other hand, as noted by Cui et al. [15], too many irrelevant passages returned by this module could hinder the Answer Extraction module in locating the correct answer. It is worth noting that many research works in open-domain QA have studied the Passage Retrieval problem and proposed different density-based algorithms, which are quantitatively evaluated by Tellex et al. [16]. The lack of domain specific knowledge makes these works different from ours significantly because no semantics is taken into consideration; the similarity between the question and the documents is statistically measured based only on the original words. However, this is rather an advantage of domain specific QA systems in terms of the resources available to them, than a limitation of current approaches being used for open-domain QA systems. The work of Zhang et al. [7] takes semantics into consideration while weighting similarity between the question and passages. This is similar to our Passage Retrieval approach described below; however, this work lacks the fine-grained processing performed by our Answer Extraction module to filter out passages that contain the query terms but not in the right syntactic structure to answer the question. The following paragraphs below describe each component of the Passage Retrieval module.

Document Processing: Document Processing involves two text processing tasks: Semantic annotation and Passage splitting.

Although to date there is no formal definition of "Semantic Annotation", this concept is generally referred to as "a specific metadata generation and usage schema, aiming to enable new information access methods and to extend the existing ones" [17]. In other words, the semantic annotation task is performed based on the domain ontology, in order to associate the appearances of domain specific terms or named entities with the respective ontological concepts, therefore anchoring those terms or named entities within contents to their corresponding semantic information. There have been a lot of research efforts within the field of semantic annotation with respect to discerning what to annotate, what additional information users expect to have, whether to embed annotation or not, and how to perform automatic annotation etc. [17]. It is our belief that a semantic annotation strategy should be tailored to the specific task at hand. In the context of Personalized QA, we employ a similar strategy used in the Knowledge and Information Management (KIM) platform [17], to link annotations to concepts in the domain ontology. The General Architecture for Text Engineering - GATE platform [18] provides a type of Processing Resource called a Gazetteer, which performs gazetteer lists lookup, furthermore linking recognized entities to concepts in ontology based on a mapping list manually defined by users. However, instead of embedding an alias of the instance's URI (Uniform Resource Identifier) as in KIM, we directly embed the label of the most specific class associated with the recognized terms or named entities inline with

the texts. For example, the following sentence "*CompanyX releases a new operating system.*" is annotated with "*CompanyX BIOntoCompany releases a new operating system BIOntoSoftware*" whereby *BIOntoCompany*, *BIOntoSoftware* are the labels of ontological concepts *http://localhost/temp/BIOnto#Company*, *http://localhost/temp/BIOnto#Software* respectively. Care is taken while naming the labels by prefixing the concept name with the ontology name, which is "*BIOnto*" in our use case, to make them unique. The rationale for this semantic annotation strategy is as follows:

- Preserving the original terms or named entities e.g. "*CompanyX*" ensures that exact keyword-matching still matches, and, avoids generating noise by over-generation if the original term is completely replaced by its annotation. This ensures that such question as "*Which products did CompanyX release?*" does not get as answer a sentence referring to products related to other companies.
- Embedding the class label directly in the text adds semantic information about the recognized terms or named entities. The annotation BIOntoCompany that follows "*CompanyX*" provides an abstraction that helps to answer such question as "*Which products did the competitors release?*". In this case, the term "*competitors*" in the question is also annotated with *BIOntoCompany*, therefore, a relevant answer can be found even though the question does not mention the company name specifically.
- Based on the concept label, the system can derive the URI of the concept in the domain ontology, query the ontology for relevant concepts and use them to expand the set of query terms of the original question.

Once the documents are annotated, they are split into passages based on the paragraph marker identified by GATE. Each passage is now considered as one document on its own and is indexed in the next step. Before indexing is carried out, stop-word removal is applied to each of the documents. Stop-words are words that do not carry any significant meaning, such as "*the*", "*a*", etc. They are used frequently but do not help to distinguish one document from the others and therefore do not help in searching [19]. Removing these insignificant words makes the indexing and searching task more effective. Porter stemming is also applied to convert the morphological variants of words into their roots.

Document Indexing: The texts within processed documents are fully indexed using Lucene[6], a freely available Information Retrieval (IR) library. Lucene supports Boolean queries based on the well-known *tf.idf* scoring function in IR research. Interested readers are referred to [19] for more details on the scoring formula being used in Lucene.

Question Analysis: Question Analysis involves three text processing tasks: Question typing, Semantic annotation, and Query terms expansion.

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

Question typing is a common process used in many QA systems. For instance, in [20], question type taxonomy is created to map the questions into their respective types. This helps to bridge the gap between wordings used in the question and those used in the texts, for example, the system is aware that question starting with "*Where*" asks about places so it is typed as "*Location*". However, since domain specific QA system already has the domain ontology in place, instead of building a separate taxonomy for a question type as in [20], a set of pattern-matching rules is built to map the question type to one of the concepts in the domain ontology. Therefore, for a question such as: "*Which products did CompanyX release?*", the question type is BIOntoProduct. The Wh-word and the pronoun that follow are replaced by the question type; and the question becomes "*BIOntoProduct did CompanyX release?*".

There are, however, some special cases in question typing, for instance, from the sample questions from business analysts in our use case, we observe that for "*Yes/No*" questions such as "*Are there any CompanyX's plans to release new products?*" end-users actually do not expect to receive "*Yes*" or "*No*" as an answer but instead the proof that the entities/events of interest exist if they do. Therefore, a set of pattern-matching rules is in place to reform this type of questions to the form of "*What*" question, for the above example it is reformed to "*What are CompanyX's plans to release new products?*" and then the question typing process is carried out as mentioned above. There are also cases whereby the questions cannot be typed to one of the domain concepts. In these cases, question words are removed and the remaining words are treated as a set of query terms.

Once the question is typed, it is also annotated, but in a different manner from the semantic annotation performed on documents. Care is taken so that specific named entities are not annotated with their ontological concepts' label to avoid noise, e.g. attaching a label *BIOntoCompany* after the word "*CompanyX*" in the question will match any terms annotated with *BIOntoCompany* in the document.

Before splitting the questions into query terms and submitting to IR engine, Query terms expansion is performed based on the domain ontology and a set of domain specific synonyms. Initial analysis of the sample questions from the business analyst in the use case indicates two phenomena:

- When the question is typed into a concept in the ontology and that concept has sub-concepts, the question needs expanding with all the sub-concepts in the ontology. Assuming that concept *http://localhost/temp/BIOnto#Product* has sub-concepts *http://localhost/temp/BIOnto#Software* and *http://localhost/temp/BIOnto#Hardware*, the first example question in this section needs to include those two sub-concepts as query terms. This ensures that those entities or terms annotated as *BIOntoSoftware* or *BIOntoHardware* can also be matched during the searching stage.
- End-users tend to use synonyms of verbs specifically to express the same meaning. For example, "*reduce*" and "*lower*" are used interchangeably. There-

fore, synonym lookup is performed against the available synonym set to include them in the set of query terms sent to the search engine.

Performing query terms expansion based on the domain ontology and synonym sets in effect addresses the issue of ambiguity caused by rhetorical writing style used in the source document.

Searching: Lucene is used to search for indexed documents containing the expanded query terms. Boolean query type is used, with AND operator between original terms and OR operator used between expanded terms. Ranked relevant passages returned from the search are fed into the next module, Answer Extraction, to filter out sentences containing query terms whose syntactic structures do not match that of the original question.

4.2 Answer Extraction

In this module, the question is matched with a given candidate answer, which is a sentence derived from passages selected by the Passage Retrieval module. A good review of previous works on Answer Extraction is provided in [1]. Typically, once the sentence has been found by some coarse-grained processing, a set of constraints is applied to check if the candidate sentence is actually the answer. A drawback of the majority of answer extraction techniques is that those techniques such as typical word overlap or term density ranking fail to capture grammatical roles and dependencies within candidate answer-sentences, such as logical subjects and objects [15, 1]. For instance, when presented with the question "*Which company acquired Compaq?*", one would expect to retrieve "*HP*" as an answer to the question. However, typical term density ranking systems would have difficulty with distinguishing the sentence "*HP acquired Compaq*" from "*Compaq acquired HP*". It is concluded that neglecting relations between words can result in a "major source of false positives within the IR" [16].

Answer Extraction modules within systems that involve processing beyond the lexical level, typically conduct pre-annotation of grammatical constraints/ relations of matching questions and candidate sentences [1]. The set of constraints can either be regarded as being "absolute" or as a set of "preferences" [1]. However, the degree of constraint plays an important role in determining the robustness of the system. It is recommended that grammatical constraints must "be treated as preferences and not as being mandatory" [1]. Previous work, such as PiQASso [21] system, shows that strict relations matching suffers substantially from poor recall. Cui et al. [15] propose a solution to the above "strict matching problem" by employing fuzzy or approximate matching of dependency relation using MiniPar[22]. To make this paper self-contained, we provide an overview of the MiniPar Dependency Parser and the dependency tree generated by Minipar.

MiniPar Dependency Parser MiniPar[22] is a fast robust dependency parser which generates dependency trees for words within a given sentence. In a de-

pendency tree, each word or chunked phrase is represented by a node. Each node is linked: one node corresponding to the governor and the other daughter node corresponding to the modifier. The label associated between each link is regarded as a dependency relation between two nodes i.e. *subj, obj, gen* etc. Fig. 2 is generated by MiniPar and illustrates the output dependency parse trees of a sample question-Q1 and sample candidate answer-S1 taken from an extract of Form 10-Q.

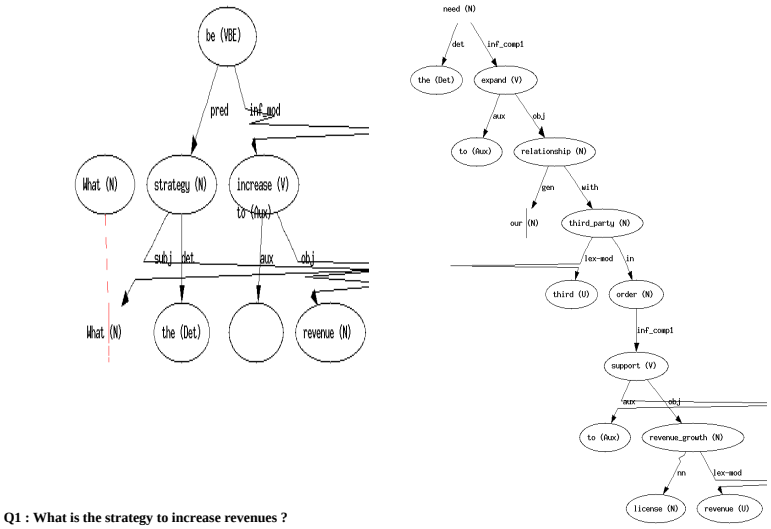


Fig. 2. Dependency trees for Question Q1 and Answer candidate S1

Approximate/Fuzzy relation matching The work of Cui et al. [15] addresses the setback of strict matching between dependency relations. In this work, the authors extract relation paths between two given nodes based on previous work in [23]. A variation of a Statistical Machine Translation model [24] is applied to calculate the probability given candidate sentence and question terms resulting in a match given a combination of relation paths. Mapping scores between relation paths are learned based on two statistical techniques: (1) Mutual Information in order to learn pair-wise relation mappings between questions and candidate answers and (2) Expectation maximization as an interactive training process [24]. Question-answer pairs are extracted from the TREC 8 and 9

QA tasks in order to provide training data. Quantitative evaluation shows that their approach achieves significant retrieval performance when implemented on a range of current QA systems, achieving a MMR 50-138% and over 95% for the top one passage. It is therefore concluded that the approximate dependency relation matching method can boost precision in identifying the answer sentence.

Applications for Personalized Question Answering It is our intention to adapt the above approach to the specific domain of Business Analysis and to integrate it as part of Answer Extraction module for the framework described in Fig. 1. We can utilize questions collected from business analysts and furthermore corpora of Form 10-Q to extract dependency relations using the MiniPar parser in order to generate sample questions-answer pairs for statistical modeling similar to [15]. We believe that this approach in combination with large samples of restricted domain training data will yield high precision while still maintaining high recall.

5 Conclusion and Future work

In this paper we introduce the idea of "Personalized Question Answering" and propose a framework for its realization. A usage scenario within the Business Analysis domain is also presented. Investigative analysis has highlighted that domain knowledge in the form of formal ontology plays an important role in shaping the approach and design of the aforementioned framework. This is particularly true for semantic annotation and query expansion whereby semantics are needed to address the issue of ambiguity caused by rhetorical writing style used in the source document. Current research indicates that: (1) the inclusion of domain semantics leads to better recall in passage retrieval; (2) in a domain-specific QA system, certain types of questions may require specific analysis (e.g. "*Yes/No*" questions in this business analysis domain); (3) the use of approximate dependency matching between questions-candidate answer pairs may yield higher precision for answer extraction without impacting on recall.

An application prototype applying Personalized Question Answering framework into Business Analysis use case is being implemented. Once the fully functional prototype is available, a quantitative evaluation scheme will be implemented to gauge the effectiveness of the system within the Business Analysis context. As the system is domain specific, the TREC QA track training data is not suitable for benchmarking, since it is targeted exclusively towards open-domain systems. The evaluation scheme will therefore involve the manual creation of test corpus of business reports, from which given a collection of test questions, Business Analysts will manually extract corresponding question answer pairs. This derived set of question/answer pairs will be used to benchmark the performance of our Personalised Question Answering System.

Future work will also involve prototype functionality enhancement to cater for complex questions whose answers are not explicitly stated, and those that

contain implicit contexts. Additional functionality will focus on a caching mechanism for the Question Analysis component to address performance measures for domain frequently asked questions. Last but not least, it is our goal to integrate the QA system with the Analyst Workbench [25] to provide business analysts with an integrated environment to perform business intelligence activity in an effective and timely manner.

Acknowledgments. We would like to thank John Collins, Business Development & Business Engineering Manager, HP Galway and Mike Turley, CEO of DERI, for discussions on business analysis problem. We also thank the anonymous reviewers for their constructive comments. This work is supported by Science Foundation Ireland(SFI) under the DERI-Lion project (SFI/02/CE1/1131).

References

1. Hirschman, L., Gaizauskas, R.: Natural language question answering: the view from here. Nat. Lang. Eng. **7** (2001) 275–300
2. Berger, H., Dittenbach, M., Merkl, D.: An adaptive information retrieval system based on associative networks. In: APCCM '04: Proceedings of the first Asian-Pacific conference on Conceptual modelling, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2004) 27–36
3. Chung, H., Song, Y.I., Han, K.S., Yoon, D.S., Lee, J.Y., Rim, H.C., Kim, S.H.: A practical qa system in restricted domains. In Aliod, D.M., Vicedo, J.L., eds.: ACL 2004: Question Answering in Restricted Domains, Barcelona, Spain, Association for Computational Linguistics (2004) 39–45
4. Sneiders, E.: Automated question answering using question templates that cover the conceptual model of the database. In: NLDB '02: Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems-Revised Papers, London, UK, Springer-Verlag (2002) 235–239
5. Bernstein, A., Kaufmann, E., Fuchs, N.E., von Bonin, J.: Talking to the semantic web a controlled english query interface for ontologies. In: 14th Workshop on Information Technology and Systems. (2004) 212–217
6. Lopez, V., Pasin, M., Motta, E.: Aqualog: An ontology-portable question answering system for the semantic web. In: ESWC. (2005) 546–562
7. Zhang, Z., Sylva, L.D., Davidson, C., Lizarralde, G., Nie, J.Y.: Domain-specific qa for construction sector. In: Proceedings of SIGIR 04 Workshop: Information Retrieval for Question Answering. (2004)
8. Demner-Fushman, D., Lin, J.: Knowledge extraction for clinical question answering: Preliminary results. In: Proceedings of the AAAI-05 Workshop on Question Answering in Restricted Domains, Pittsburgh, Pennsylvania. (2005)
9. Niu, Y., Hirst, G.: Analysis of semantic classes in medical text for question answering. In: Proceedings of 42st Annual Meeting of the Association for Computational Linguistics, Workshop on Question Answering in Restricted Domains. (2004) 54–61
10. Molla, D., Schwitter, R., Rinaldi, F., Dowdall, J., Hess, M.: Extrans: Extracting answers from technical texts. IEEE Intelligent Systems **18** (2003) 12–17
11. TREC: Text retrieval conference trec http://trec.nist.gov (2005)

12. Andrenucci, A., Sneiders, E.: Automated question answering: Review of the main approaches. In: ICITA '05: Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2, Washington, DC, USA, IEEE Computer Society (2005) 514–519
13. Dumais, S., Banko, M., Brill, E., Lin, J., Ng, A.: Web question answering: Is more always better? In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002), Tampere, Finland. (2002)
14. Diekema, A.R., Yilmazel, O., Chen, J., Harwell, S., Liddy, E.D., He, L.: What do you mean? finding answers to complex questions. In: Proceedings of the AAAI Spring Symposium: New Directions in Question Answering. Palo Alto, California. (2003)
15. Cui, H., Sun, R., Li, K., Kan, M.Y., Chua, T.S.: Question answering passage retrieval using dependency relations. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2005) 400–407
16. Tellex, S., Katz, B., Lin, J., Fernandes, A., Marton, G.: Quantitative evaluation of passage retrieval algorithms for question answering. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM Press (2003) 41–47
17. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic annotation, indexing, and retrieval. Journal of Web Semantics **2** (2005) 39
18. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: A framework and graphical development environment for robust nlp tools and applications. In: Proceedings of the 40th Annual Meeting of the ACL. (2002)
19. Hatcher, E., Gospodnetic, O.: Lucene in Action. Manning Publications Co. (2005)
20. Hovy, H., Gerber, L., Hermjakob, U., Lin, C., Ravichandran, D.: Towards semantic-based answer pinpointing (2001)
21. Attardi, G., Cisternino, A., Formica, F., Simi, M., Tommasi, A.: Piqasso: Pisa question answering system. In: Text REtrieval Conference. (2001)
22. Lin, D.: Dependency-based evaluation of minipar. In: Proc. Of Workshop on the Evaluation of Parsing Systems, Granada, Spain. (1998)
23. Gao, J., Nie, J.Y., Wu, G., Cao, G.: Dependence language model for information retrieval. In: SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2004) 170–177
24. Brown, P.F., Pietra, S.D., Pietra, V.J.D., Mercer, R.L.: The mathematic of statistical machine translation: Parameter estimation. Computational Linguistics **19** (1994) 263–311
25. O'Riain, S., Spyns, P.: Enhancing business analysis function with semantics. In Meersma, R., Tari, Z., eds.: On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA and ODBASE; Confederated International Conferences CoopIS, DOA, GADA and ODBASE 2006 Proceedings. LNCS 4275, Springer (2006) 818–835

OntoCAT: An Ontology Consumer Analysis Tool and Its Use on Product Services Categorization Standards

Valerie Cross and Anindita Pal

Computer Science and Systems Analysis, Miami University,
Oxford, OH 45056
{crossv, pala}@muohio.edu

Abstract. The ontology consumer analysis tool, OntoCAT, provides a comprehensive set of metrics for use by the ontology consumer or knowledge engineer to assist in ontology evaluation for re-use. This evaluation process is focused on the size, structural, hub and root properties of both the intensional and extensional ontology. It has been used on numerous ontologies from varying domains. Results of applying OntoCAT to two Product and Service Categorization Standards, UNPSCS and ecl@ss ontologies are reported.

Keywords: ontology evaluation, ontology metrics, ontology ranking, hubs

1 Introduction

The need for domain ontology development and management is increasingly more and more important to most kinds of knowledge-driven applications. Development and deployment of extensive ontology-based software solutions represent considerable challenges in terms of the amount of time and effort required to construct the ontology. These challenges can be addressed by the reuse of ontologies. The extent to which reuse of ontologies could contribute cost and time savings parallels that obtained in software reuse [17] because acquiring domain knowledge, constructing a conceptual model and implementing the model require a huge effort. As with any other resource used in software applications, ontologies need to be evaluated before use to prevent applications from using inconsistent, incorrect, or redundant ontologies. Even if an ontology has none of these problems and is formally correct, users must decide whether the content of ontology is appropriate for the requirements of their project, that is, they must determine how well the metadata and instances meet the requirements for their problem domain. Knowledge engineers need an ontology analysis tool to help in the process of ontology assessment for reuse.

Much ontology research has focused on new methodologies, languages, and tools [4]; recently, however, since the OntoWeb 2 position statement stressed the insufficient research on ontology evaluation and the lack of evaluation tools [11], much attention has been directed towards ontology evaluation [3,8]. Initially this attention concentrated on a formal analysis approach to evaluating ontologies [9]. Others have created taxonomies of ontology characteristics [12] to quantify the

suitability of ontologies for users' systems. Knowledge engineers must analyze these characteristics for the prospective ontologies in order to compare them and select the appropriate ontology for the system. More recent efforts address evaluating ontologies for reuse, not by ontology developers and experts, but by ontology consumers [14] who are users such as system project managers hoping to find existing ontologies on the Semantic Web which can be reused and adapted for their systems.

The objective of this research is to describe an ontology consumer analysis tool, OntoCAT [16], that summarizes essential size, structural, root and hub properties for both an intensional ontology and its corresponding extensional ontology. An intensional ontology only includes the ontology schema or definitions. An extensional ontology includes the instances, i.e., occurrences of classes and relationships. OntoCAT supports the ontology consumer by performing an analysis on the graph-like properties of an ontology. First, a brief overview of the variety of approaches to evaluating ontologies is presented in Section 2. Included in more detail in this presentation are those methods which take the ontology consumer perspective on evaluation. Section 3 describes some of the metrics included in OntoCAT. OntoCAT has been created as a plug-in for the Protégé Ontology Editor (http://protege.standford.edu/overview/). The OntoCAT user interface is presented in Section 4. The results of performing an ontology consumer analysis on two product services and categorization standards (PSCS), UNSPSC (United Nations Standard Products and Services Code) and ecl@ss are discussed in Section 5 along with a brief description of the UNSPC and eCl@ss ontologies. Conclusions and future planned work are presented in Section 6.

2 Ontology Evaluation

A variety of approaches to ontology evaluation have been proposed depending on the perspectives of what should be evaluated, how it should be evaluated and when it should be evaluated. As such, ontology evaluation has become a broad research area [3] with numerous frameworks proposed for evaluating how "good an ontology is". These frameworks can be classified into various categories depending on what qualities are considered most relevant to an ontology and how they are being evaluated. These qualities may also have an importance factor. For example, is the quality of the design more important than the quality of the content [15] and can a gold standard be used or is an assessment by a human expert required [3]? In addition some evaluation methods make specific recommendations about when in the ontology development lifecycle the evaluations should be performed. Others suggest developing methodologies to evaluate an ontology during the development process and throughout its entire lifetime [9].

Another distinction made in ontology evaluation is that of selection versus evaluation. In [18], ontology selection is defined as "ontology evaluation in the real Semantic Web." Their survey of existing selection algorithms reveals that few ontology evaluation methods are incorporated except for similarities in topic coverage. They conclude that although evaluation and selection consider different requirements, they are complementary. In [7] a holistic view of ontology evaluation

is considered by viewing an ontology as a communication object. The Qood grid method permits parametric design for both evaluation and selection (diagnostic) tasks. Ontologies are analyzed in their graph and formal elements, functional requirements, and annotation profile. The Qood evaluation based on graph analysis parallels that of the OntoCAT metrics proposed in [5].

Due to space limitations, not all of these various evaluation methods can be discussed. The following sections briefly describe ontology consumer evaluation methods and overview the two tools most closely related to OntoCAT.

2.1 Ontology Evaluation from the Consumers' Perspective

To make reusing ontologies easier, more research needs to address the evaluation of an ontology from the consumer point of view [14]. Ontology consumers needs tools to help with two different tasks, selecting from the enormous number of available ontologies the most appropriate ontology candidates for their applications and quality evaluation of ontologies. As pointed out previously, ontology evaluation and ontology selection are complementary. The question is in what order should these two tasks be performed. The answer might depend on the individual ontology developer, but typically the answer is that selection is performed first for filtering purposes and then followed by a more time consuming quality evaluation. Selection or filtering methods typically employ topic coverage, popularity, and richness of conceptualized knowledge [18].

Consumer analysis tools could be useful to both selection and evaluation tasks. One approach suggested in [14] for consumer analysis is ontology summarizations. Ontology summarizations are analogous to approaches used by researchers in reviewing the usefulness of papers or deciding on whether to purchase a book. Just as researcher examines a book's summary or a paper's abstract when deciding on its usefulness, similarly there should be some abstract or summary of what an ontology covers to help consumers decided if it fits their application requirements. The summary might include the top-level concepts and links between them as a graphical representation and a listing of hub concepts – concepts that have the largest number of links in and out. It could also include metrics similar to Google's page rank that determine that a concept is more important if other important concepts are linked to it.

OntoCAT metrics are based on these analogies and fall into both the structural and functional types of measures for ontology evaluation [7]. OntoCAT metrics can be valuable to both the selection and evaluation tasks performed on ontologies. The summaries which provide the consumer with a high level view of the topic coverage are functional types of measure and important to the selection task. The OntoCAT metrics analyzing an ontology as a graph structure are structural metrics that can be used to evaluate the quality of the ontology design similar to those used for software metrics [17]. Two of the more recent related approaches to OntoCAT are presented below. The ones selected either focus on quantitative size and structural metrics for ontology selection or evaluation or they have a component that includes such metrics. Structural types of measures in [7] correspond closely with OntoCAT metrics presented in [5].

2.2. OntoQA

The OntoQA tool [20] developed by LSDIS Lab at the University of Georgia measures the quality of ontology from the consumer perspective. The OntoQA tool measures the quality of the ontology in terms of Schema and Instance metrics. The schema metrics of OntoQA address the design of the ontology schema, some of which correspond with the OntoCAT intensional metrics calculated on the intensional ontology (the definition of the ontology). Instance metrics of OntoQA deal with the size and distribution of the instance data, some of which correspond with the OntoCAT extensional metrics calculated on the ontology occurrences.

OntoQA defines three metrics in its Schema metrics group. These are relationship richness, attribute richness and inheritance richness. Relationship richness measures the percentage of relationships that are not is-a or taxonomic relationships. Attribute richness measures the average number of attributes per class, dividing the cardinality of the attribute set by the cardinality of the class set. Inheritance richness measures the average number of subclasses per class.

Metrics for the Instance group are categorized into two subclasses: the whole instance ontology or class metrics. Class metrics are used to describe how a class is being populated in the instance ontology. Metrics for the whole instance ontology include class richness, average population, and cohesion. Class richness measures the distribution of instances across classes. Formally, it is defined as the ratio of the number of classes with instances divided by the number of classes defined in the ontology. Average population measures the average distribution of instances across all classes. Formally, it is defined as the number of instance divided by the number of class in the ontology. Cohesion measures the number of connected components in the instance graph built using the taxonomic relationships among instances.

Class metrics include importance, fullness, inheritance richness, connectivity, relationship richness and readability. Importance refers to the distribution of instance over classes and is measured on a per sub-tree root class. It specifies the percentage of instances that belong to classes in the sub-tree rooted at the selected class with respect to the total number of instances in the ontology. This definition is somewhat confusing because multiple instance sub-trees for a selected class could exist. It is assumed that this definition would include all instances of sub-trees with the selected class type. Fullness is primarily for use by ontology developers to measure how well the data population was done with respect to the expected number of instances of each class. It is similar to importance except that it is measured relative to the *expected* number of instances that belong to the sub-tree rooted at the selected class instead of the total number of instances in the ontology. Connectivity for a selected class is the number of instances of other classes connected by relationships to instances of the selected class. Relationship richness measures the number of the properties in the selected class that are actually being used in the instance ontology relative to the number of relationships defined for the selected class in the ontology definition. Readability measures the existence of human readable descriptions in the ontology. Human readable descriptions include comments, labels, or captions.

2.3 AKTiveRank

Several ontology search engines such as Swoogle [6] and OntoSearch [21] can be used by entering specific search terms to produce list of ontologies including search terms somewhere in the ontology. AKTiveRank [1] ranks the ontologies retrieved by an ontology search engine. Its initial implementation evaluated each retrieved ontology using four measures: class match, density, semantic similarity and centrality.

The class match measure evaluates the coverage of an ontology by providing a score based on the number of query terms contained in the ontology and evaluates using both exact match where the search term is identical to the class name and partial match where the search term is contained within the class name. The density measure evaluates the degree of details in the representation of the knowledge concerning the matching classes. The density value for an individual class is the weighted sum of the count of its number of super-classes, subclasses, direct and indirect relations (in and out), and siblings. The number of instances was initially included but dropped since it was felt that this parameter might bias evaluation toward populated ontologies [2]. This bias might penalize ontologies with higher quality definitions (schemas). The density measure for the query is the average for all matching classes. The semantic similarity measure (SSM) determines how close the classes that match the search terms are in an ontology. The semantic similarity is calculated between all pairs of matching classes and then the average is taken.

The centrality measure assumes that the more central a class is in hierarchy, the better analyzed and more carefully represented it is. A centrality measure for a class is calculated for each class that matches fully or partially a given query term based on its distance from the class midway from the root to the leaf on the path containing the matching class. Then the centrality measure for the query is the average for all matching classes. More recent research [2] identified the redundancy of the centrality measure because of its close correspondence with the density measure and replaced it with the betweenness measure. The higher the betweenness measure for a class, the more central that class is to the ontology. For betweenness, the number of shortest paths between any two classes that contains a class matching a queried concept is calculated. These numbers are summed over all queried concept. Their average determines the betweenness measure. The overall rank score for an ontology is the weighted aggregation of these resulting component measures is performed to produce the overall rank of the ontology.

The researchers creating AKTiveRank note that such a tool “needs to be designed in a way so as to provide more information of the right sort. Mere ranking, while it will no doubt be useful, may not be the whole story from a practical perspective” and further suggest that there is “a need to disentangle a number of different parameters which can affect the usability of a knowledge representation” since the perception of the knowledge engineers with respect to different ontology parameters “may vary depending on the particular application context” [1]. A limitation of this tool is that it only ranks intensional ontologies since all measurements are based on the definition of the ontology. There are some ontologies, especially terminological ontologies, whose intensional ontology is quite simple but whose extensional ontology is quite complex. An ontology consumer analysis tool should be able to process both components of an ontology to provide the user with as much information as possible.

3 OntoCAT Metrics

The ontology consumer analysis tool OntoCAT provides a comprehensive set of metrics to be used by the ontology consumer or knowledge engineer to assist in ontology evaluation for re-use. Quality measurements are not being provided but instead summarizations, size and structural metrics are provided. The metrics are separated into two categories: intensional metrics and extensional metrics. Intensional metrics are calculated based on the ontology definition itself, that is, its classes and subclasses and their properties. Extensional metrics measure the assignment of actual occurrences of ontological concepts, that is, the instances and how effectively the ontology is used to include the domain knowledge. Much research has been focused on extensional ontologies, in some part, because the consideration for reuse of ontologies has often been on terminological ontologies such as found in the biomedical fields.

The following metrics are relative to the metadata being assessed, C – class, P – property, A – attribute, and R – relationship. Metrics beginning with an "i" are for the intensional ontology, and those beginning with an "e" are for the extensional ontology. Some of the metrics do not return a numeric value but instead indicate identifying information. For example, iMaxClasses provides a list of classes that have the maximum number of properties. In the following, Cnt stands for count, Avg for average, and Rng for range. The main approach is to determine various metrics and to also examine them on both horizontal (per depth) and vertical (per path) slices of the ontology. Below only a sample of the metrics are presented due to page limitations. A complete description of all the metrics can be found in [16].

3.1 Size Metrics

Intensional. Typically an intensional ontology has one root concept, but multiple root concepts are possible. If no concept or class c_j is specified, the intensional size metric is calculated for the entire ontology, that is, over all the trees defined in the ontology. When a concept c_j is specified to be used as a root, the size metric is calculated on the tree specified by the selected concept c_j as its root. Although the measures using a root are referred to as size metrics, they do, however, use the "*is-a*" or subsumption hierarchy to determine the tree for which the size metrics are being determined.

iCnt(C) = the number of classes for the entire intensional ontology
iCnt(C)(c_j-root) = the number of classes for the sub-tree at the selected class c_j.
iCnt(P) = the number of properties defined for the entire intensional ontology
iCnt(P)(c_j-root) = the number of properties defined for the entire sub-tree at class c_j. A property may be inherited from its parents. Only new properties are counted for each class.
iCntTotal(P)(c_j) =the total (new + inherited) number of properties for class c_j.
iCnt(R) = the number of relationships defined for the entire intensional ontology. A relationship is a special kind of property that has a class as its range.
iCntTotal(R)(c_j) = the total (new + inherited) number of relationships defined for only the selected class

iMaxTotal(P to C)(c_j-root) = max number of (new + inherited) properties defined for a single class over all classes in the sub-tree at the selected class c_j

iMaxTotalClasses(P to C)(c_j-root) = class names for classes within sub-tree at the selected class c_j that have the max number of properties

Extensional.

eCnt(c_j) = the number of object occurrences for class cj

eCnt(C) = $\sum_j$ eCnt(c_j), the total number of object occurrences in the ontology

eCnt(C)(c_j-root) = $\sum_i$ eCnt(c_i), the total number of object occurrences in the sub-trees for the selected class c_j where c_i is in sub-tree c_j

eAvgCnt(C) = eCnt(C)/iCnt(C), the average number of occurrences for all classes

eMaxCnt(C) = $\max_i$[eCnt(c_i)] and identify eMaxCntClass, i.e., the class with the maximum number of occurrences in the ontology

eCnt(r_i) = the number of occurrences for relation r_i

eCnt(R) = $\sum_i$ eCnt(r_i) total number of occurrences for all relations in ontology

eAvgCnt(R) = eCnt(R)/eCnt(C), average number of relationships per occurrence

eMaxCnt(R) = $\max_i$[eCnt(r_i)] and identify eMaxCntRelation

3.2 Structural Metrics

Intensional structural metrics are similar to size metrics since they are over the entire intensional ontology, that is, over all the root trees defined in the ontology if no concept or class is specified. When a class is specified, the structural metrics are calculated for the entire sub-tree at that class c_j. The class hierarchy (sub-class/super-class) relationships are used for the structural organization.

iCnt(Roots) = number of roots in the ontology.

iCnt(leaves(c_j-root)) = number of leaf classes of the sub-tree at the selected class c_j

iCnt(leaves) = $\sum_j$ iCnt(leaves(c_j-root); the total number of leaf classes in the entire ontology where c_j-root is a root class.

iPer(leaves(c_j-root)) = iCnt(leaves(c_j-root)) /iCnt(C)(c_j-root) the fraction of classes that are leaves of the is-a hierarchy for the entire sub-tree at class c_j.

iAvg(leaves) = iCnt(leaves)/iCnt(C), the fraction of classes that are leaves for the entire ontology.

iMaxDepth(c_j-root) = $\max_j$ [depth($leaf_{ij}$)], the maximum depth of the sub-tree at the selected class c_j and return the class of the leaf at the maximum depth

Several intensional structural metrics are adapted from WordNet's (IC) measure [19]. The IC for class c_{ij} for c_j-root (the class may be in multiple trees, therefore, the subscript j specifies the root of the tree) is given as [5]:

iIC(c_{ij}) = 1- log(iCnt(C)(c_{ij}-root) + 1)/log(iCnt(C)(c_j-root))

The class c_j-root must be a root class of the ontology whereas, c_{ij}-root is any class within the inheritance hierarchy rooted at c_j-root. This measure can be used to identify the degree of information content on a per depth basis within the ontology. Using class information content as a new measure provides a novel way of examining the ontology for symmetry and balance across its horizontal slices. Some of the following measures proposed for each c_j-root of an intensional ontology are:

iIC(depth$_k$(c_j-root)) = $\sum_i$ IC(c_{ij}) for all c_i at depth k for c_j-root
iAvgIC(depth$_k$(c_j-root)) = iIC(depth$_k$(c_j-root))/iWidth(idepth$_k$(c_j-root))
iRngIC(depth$_k$(c_j-root)) = iMaxIC(depth$_k$(c_j-root)) - iMinIC(depth$_k$(c_j-root))
iAvgIC(c_j-root)= $\sum_k$ iAvIC(depth$_k$(c_j-root)) / iMaxDepth(c_j-root), the average IC over all depths in the tree at root c_j-root

Extensional. Structural metrics are calculated on the specified root concept c_j-root and the specified relationships used to create the structural organization of the extensional ontology. For example, in the WordNet extensional ontology, the specified relationships providing its structure are the hyponym and hypernym relationships. If no concept is specified or if the specified concept is the top most concept of the ontology, structural extensional metrics for the complete ontology are calculated. When c_j-root is specified for metrics of extensional ontologies, only root occurrences of this class with respect to the structural organization of the extensional ontology are considered. The metrics listed below that have the occ parameter are automatically produced for each root occurrence of the class selected by the user.

eCnt(roots) = number of root occurrences for all root classes
eCnt(leaves(c_j-root)) = number of leaves for all occurrences of class c_j-root$_i$.
eCnt(leaves(c_j-root (occ)) = number of leaves for specified occurrence of c_j-root$_i$.
eMaxCnt(leaves(c_j-root)) = max$_i$ [eCnt(leaves(c_j-root (occ$_i$)))], the maximum number of leaves in all rooted occurrences of class c_j-root, give its identity
eMinDepth(c_j-root(occ)) = min$_i$ [depth(leaf$_{ij}$(occ))], the minimum depth of the sub-tree at the selected root occurrence of root class c_j and return the leaf occurrence(s) at the minimum depth.
eWidth(depth$_k$(c_j-root)) = ($\sum_i$ eWidth(depth$_k$(c_j-root (occ$_i$))), the number of instances at depth k for all occurrences of the selected root class c_j

3.3 Summarization Metrics

The hub summary displays information on the hubs, i.e., object occurrences (for extensional) and classes (for intensional) having the maximum number of links in and out. For intensional, the count of links is the number of subclasses and superclasses defined. For extensional, the links are based on the relationships specified for creating its structure. A list of the top n hubs (user-specified), is reported with statistics for each hub. Note that the 'i' or 'e' preceding the metric is omitted since it is determined by whether it is for an intensional or extensional ontology.

depth(hub) = the depth of the tree where the hub concept is located
width(hub) = the number of other occurrences at that depth in the tree
Cnt(P(hub)) = number and list of properties defined for the hub in case of classes
Cnt(child(hub)) = the number and list of direct children of the hub
Cnt(parent(hub)) = the number and list of direct parents of the hub

A root summary may be calculated for both the intensional and extensional ontology and include class or occurrence counts for roots and leaves, the minimum, maximum and average depths of the intensional and extensional ontology, and the minimum, maximum, and average widths of the intensional and extensional ontology.

4 OntoCAT User Interface

OntoCAT is implemented as a Protégé plug-in so that it is incorporated into the ontology engineering environment itself. As an ontology is being developed, OntoCAT may be executed to determine how the structural types of measures change during the development cycle. Since OntoCat is part of the ontology engineering environment, evaluation can easily be performed without altering the ontology.

The implementation is generalized to handle the structural difference in ontologies and is parameterized to permit easily switching between an intensional and extensional ontology. The user selects the root class and relationship to be used to calculate the metrics. The implementation uses the OWL API because of its flexibility and easy access for ontologies. Metrics for ontologies in RDF/RDF(S) can also be calculated through conversion to OWL ontologies with Protégé's export function.

The main user interface consists of two split panes. The left "Selection" panel permits selection of the metrics. The "Result" panel displays the calculated metrics. In the following figures, a small version of the WordNet ontology is used. WordNet is a general terminological ontology of the English language which serves as a freely available electronic dictionary organized by nouns, verbs, adjectives and adverbs into synonym sets (synsets), each representing one underlying lexical concept [13].

4.1 Selection Panel

The user selects which metrics to calculate for the ontology. Figure 1 shows the "Selection" panel. The metrics are grouped into intensional size and structure and extensional size and structure. This organization allows the users to easily switch between the intensional and extensional ontology. The IC metrics are separated from the structural metrics for aesthetic reasons. Users also enter the depth values in the two text fields for calculating the IC metrics and the width metrics at the depth.

The next set of parameters to input after selection of metrics is the root concept on which to measure and the relationship used to build the extensional taxonomical structure. The user can select these parameters by clicking the "Metrics" or "Report" buttons. When these buttons are clicked a pop-up window is opened as shown in Figure 2 below. This window contains the list of classes and relationships defined in the ontology. For example, after selecting the metrics shown in Figure 1, the user clicks on the "Metric" button. The concept and relation selection window pops up. In Figure 2, the Lexical Concept class of the WordNet ontology is selected. By default metrics are calculated on the whole ontology so that users only need to select the class on which they want to calculate metrics. For the "Metric" button if no class is selected, then only the ontology level metrics are displayed since there is no space in the UI to display the metric results for all classes in the ontology. For the "Report" button, if no class is selected, the metrics are calculated on all classes in the ontology. The "Report" button generates a report of the selected metrics to a file in the tools home directory. This report is formatted for easy importing to an Excel spreadsheet to analyze and generate charts and tables as done in Section 5. If users do not select a class, the report is generated on each of the classes in the ontology.

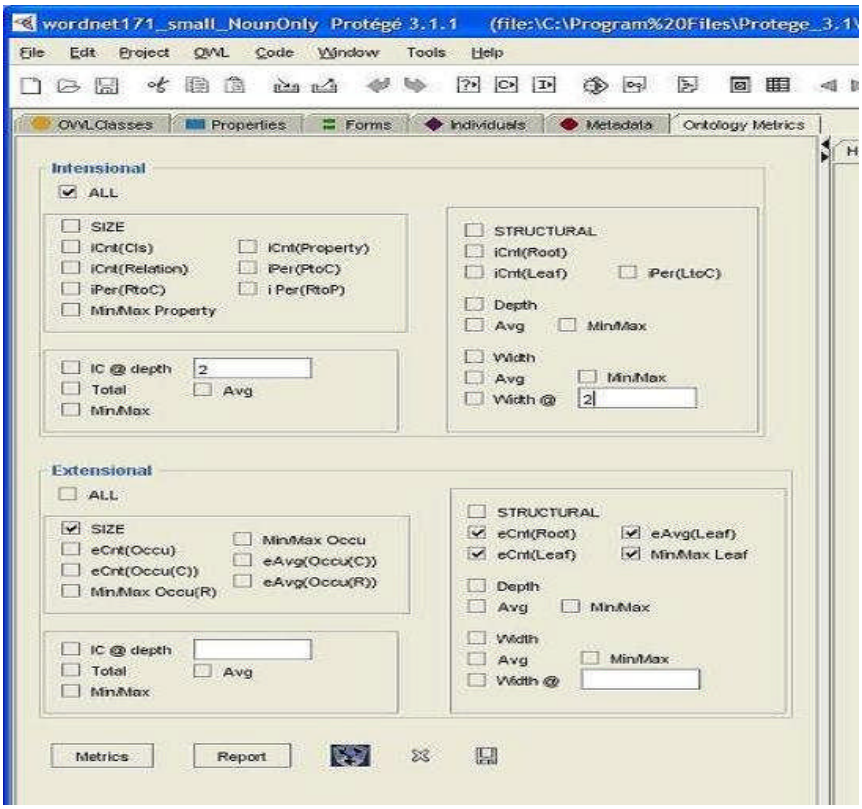


Fig. 1. Selection Panel with list of metrics

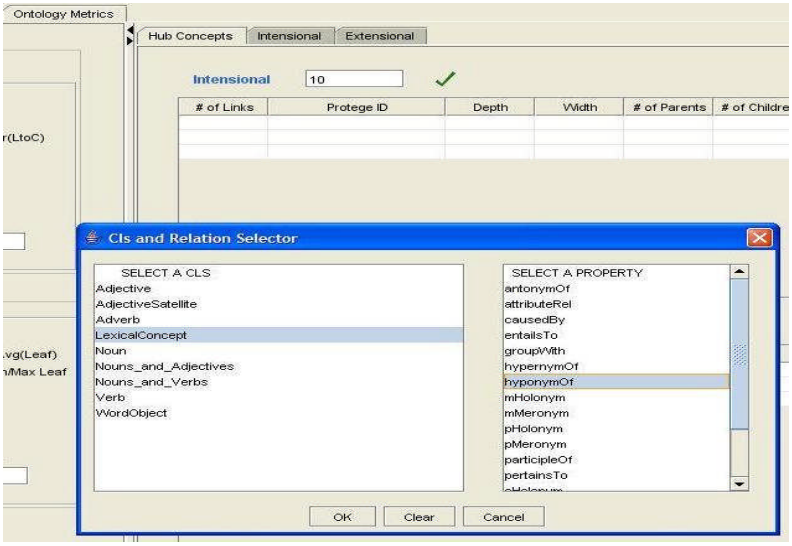


Fig. 2. Selection of Class and Relationship

The user selects the relationship for building the extensional taxonomic structure. For intensional, the structure uses the sub-class relationship. The extensional taxonomic structure differs from ontology to ontology. For example, WordNet uses the hyponymOf and hypernymOf relationships. If the extensional metrics button is selected, the parent relationship for structuring must be entered.

4.2 Result Panel

The "Result" panel displays the metrics from the "Selection" panel and has three tabs: Hubs, Intensional, and Extensional. Figure 3 shows the hub summary for a small WordNet ontology. The intensional table lists the hub classes, those with the maximum number of subclasses and super-classes. The extensional table lists the instance hubs, those with the maximum number of links in and out. The last table is specific to the class listed above the table. For example, the third table lists the extensional hubs for the lexicalConcept class selected in the "Selection" panel pop-up window. The summary provides the following: depth, width, number of properties and IC. By default the plug-in displays the top 10 hub concepts. Users can specify the number or percent of hubs to display by changing the value in the text fields (for example to 20 or 10%), located beside the table labels and clicking the ✓ button.

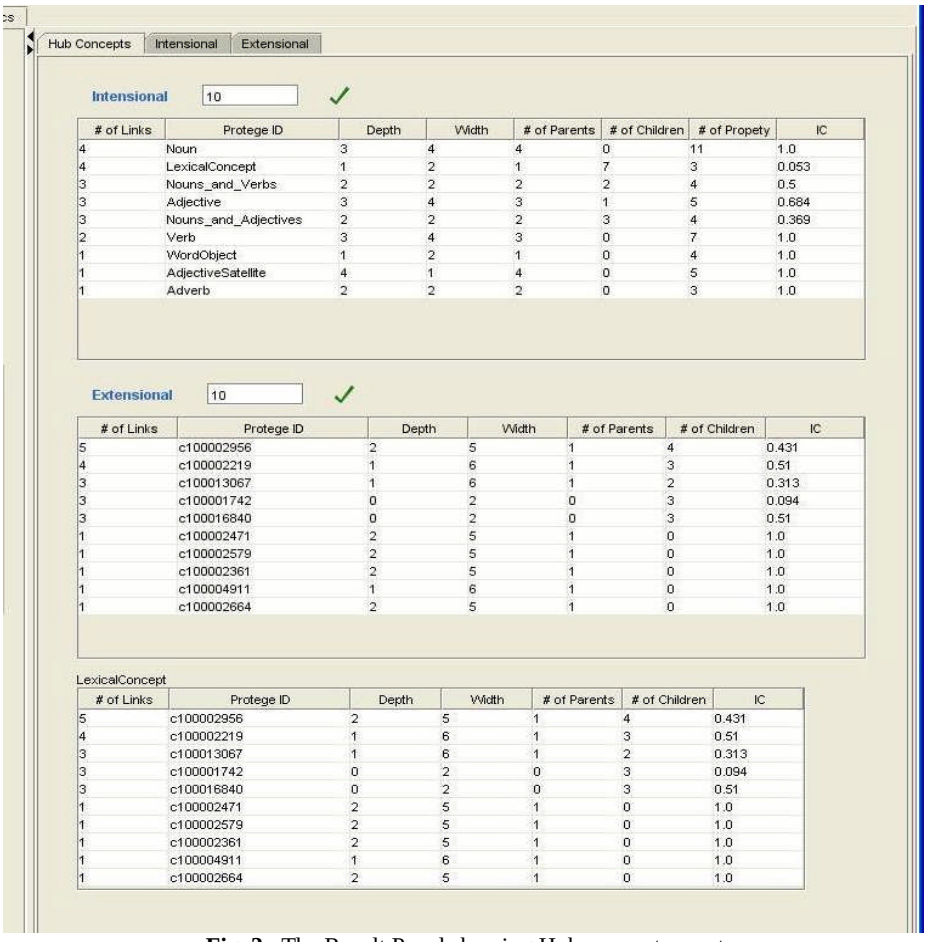
5 Analysis of UNPSCS and ecl@ass Ontologies

An important requirement of e-Commerce is effective communication between software agents. A common approach to provide machine-accessibility for agents is a standardized vocabulary of product and services terminology referred to as Product and Service Categorization Standards (PSCS) [10]. UNSPSC (United Nations Standard Products and Services Code) and eCl@ss are two example PSCS developed into intensional ontologies consisting of the schemas and definitions of the concepts in the product and service domain. UNSPSC is a hierarchical classification of all products and services for use throughout the global marketplace. eCl@ss, developed by German companies, offers a standard for information exchange between suppliers and their customers. It uses a 4-level hierarchical classification system that maps the market structure for industrial buyers and supports engineers at development, planning and maintenance. Martin Hepp has developed an OWL version (http://www.heppnetz.de/eclassowl).

A previous study of PSCS ontologies uses a framework of metrics "to assess the quality and maturity of products and services categorization standards" [10]. This framework is applied to the most current and multiple past releases of eCl@ss, UNSPSC, eOTD, and RNTD. In that study, the term "size of segments" corresponds to OntoCAT's iCnt(C)(c_j-root), the number of classes for a root class. The term "size" corresponds to OntoCAT's iCnt(C), the number of classes for the entire intensional ontology. The "property list size" corresponds to iCnt(P), the number of properties defined for the entire intensional ontology Using OntoCAT, an analysis for both

UNSPSC and eCl@ss was performed. Due to space limitations, only root summary reports are provided below in table format. Because eCl@ss has over 25000 roots, its root summary shows only a selected set of roots that have more interesting data.

Table 1 shows the root summary for the UNPSCS ontology. It is arranged in descending order of the total classes under that root class. Only the top 13 roots are shown due to space limitation. For all root classes there is a uniform maximum and minimum depth of 3. The root classes have all leaves at the same level and the maximum width occurs at the maximum depth, i.e., it is equivalent to the number of leaves for the root class. The minimum width varies but it always occurs at depth 1, i.e., the first level down from the root. The four root classes with the greatest number of classes and leaves are "Drugs Pharmaceutical_Products", "Chemicals including Bio Chemicals and Gas Materials", "Laboratory and Measuring and Observing and Testing Equipment", and "Structures and Building and Construction and Manufacturing Components and Supplies."



# of Links	Protege ID	Depth	Width	# of Parents	# of Children	# of Propety	IC
4	Noun	3	4	4	0	11	1.0
4	LexicalConcept	1	2	1	7	3	0.053
3	Nouns_and_Verbs	2	2	2	2	4	0.5
3	Adjective	3	4	3	1	5	0.684
3	Nouns_and_Adjectives	2	2	2	3	4	0.369
2	Verb	3	4	3	0	7	1.0
1	WordObject	1	2	1	0	4	1.0
1	AdjectiveSatellite	4	1	4	0	5	1.0
1	Adverb	2	2	2	0	3	1.0

# of Links	Protege ID	Depth	Width	# of Parents	# of Children	IC
5	c100002956	2	5	1	4	0.431
4	c100002219	1	6	1	3	0.51
3	c100013067	1	6	1	2	0.313
3	c100001742	0	2	0	3	0.094
3	c100016840	0	2	0	3	0.51
1	c100002471	2	5	1	0	1.0
1	c100002579	2	5	1	0	1.0
1	c100002361	2	5	1	0	1.0
1	c100004911	1	6	1	0	1.0
1	c100002664	2	5	1	0	1.0

# of Links	Protege ID	Depth	Width	# of Parents	# of Children	IC
5	c100002956	2	5	1	4	0.431
4	c100002219	1	6	1	3	0.51
3	c100013067	1	6	1	2	0.313
3	c100001742	0	2	0	3	0.094
3	c100016840	0	2	0	3	0.51
1	c100002471	2	5	1	0	1.0
1	c100002579	2	5	1	0	1.0
1	c100002361	2	5	1	0	1.0
1	c100004911	1	6	1	0	1.0
1	c100002664	2	5	1	0	1.0

Fig. 3. The Result Panel showing Hub concept report

Table 2 displays the root summary for the several of the *_tax roots of the ecl@ss ontology. Note that the maximum depth for all the root classes is 4 and the minimum depth is one. Unlike UNPSCS, the length of every path for each root class in the ontology is not identical since a variation exists in the average depth. The maximum width occurs not at the greatest depth but at depth 3 for all roots. Like UNPSCS, the minimum width varies but is always occurs at depth 1 for each root. Looking at the ratio of total number of leaves to total number of classes, UNPSCS has a much larger percentage of leaf classes for its roots as compared to ecl@ss.

Table 1. UNPSCS Root Summary

Concept Name	Total Classes	Total Leaf	Max Depth	Min Depth	Avg Depth	Max Width	Level @ max	Min Width	level @ min	Avg Width
Laboratory_and_Measuring_and_Observing_and_Testing_Equipment	1103	1008	3.00	3	3	1008	3	3	1	367.66
Structures_and_Building_and_Construction_and_Manufacturing_Components_and_Supplies	696	615	3.00	3	3	615	12	1	1	232
Chemicals_including_Bio_Chemicals_and_Gas_Materials	614	508	3.00	3	3	508	3	14	1	204.66
Drugs_and_Pharmaceutical_Products	611	514	3.00	3	3	514	3	15	1	203.66
Commercial_and_Military_and_Private_Vehicles_and_their_Accessories_and_Components	496	417	3.00	3	3	417	3	10	1	165.33
Communications_and_Computer_Equipment_and_Peripherals_and_Components_and_Supplies	416	369	3.00	3	3	369	3	3	1	138.66
Industrial_Manufacturing_and_Processing_Machinery_and_Accessories	395	342	3.00	3	3	342	3	13	1	131.66
Distribution_and_Conditioning_Systems_and_Equipment_and_Components	340	312	3.00	3	3	312	3	4	1	113.33
Power_Generation_and_Distribution_Machinery_and_Accessories	305	275	3.00	3	3	275	3	5	1	101.66
Farming_and_Fishing_and_Forestry_and_Wildlife_Contracting_Services	280	237	3.00	3	3	237	3	8	1	93.33
Industrial_Production_and_Manufacturing_Services	279	236	3.00	3	3	236	3	9	1	93
Politics_and_Civic_Affairs_Services	279	241	3.00	3	3	241	3	8	1	93
Domestic_Appliances_and_Supplies_and_Consumer_Electronic_Products	275	246	3.00	3	3	246	3	7	1	91.66

Table 2. ecl@ss Root Summary

Concept Name	Total Classes	Total Leaf	Max Depth	Min Depth	Avg Depth	Max Width	Level @ max	Min Width	level @ min	Avg Width
C_AAG961003-tax	10623	5038	4	1	3.94	5292	3	20	1	2655.75
C_AAB572002-tax	5317	2181	4	1	3.8	2624	3	35	1	1329.25
C_AAB072002-tax	3983	1669	4	1	3.82	1973	3	19	1	995.75
C_AAD302002-tax	3585	1317	4	1	3.71	1756	3	37	1	896.25
C_AAF876003-tax	2927	1315	4	1	3.88	1444	3	20	1	731.75
C_AAC473002-tax	2653	1186	4	1	3.88	1320	3	7	1	663.25
C_AAC350002-tax	2431	1024	4	1	3.82	1192	3	24	1	607.75
C_AAB315002-tax	2127	850	4	1	3.77	1041	3	23	1	531.75
C_AAA183002-tax	2065	832	4	1	3.79	1019	3	14	1	516.25
C_AAA862002-tax	1927	739	4	1	3.73	932	3	32	1	481.75
C_AAA647002-tax	1603	589	4	1	3.68	763	3	39	1	400.75
C_AAD111002-tax	1519	580	4	1	3.74	750	3	10	1	379.75
C_AAF397003-tax	1451	499	4	1	3.62	680	3	46	1	362.75
C_AAT090003-tax	1239	445	4	1	3.64	577	3	43	1	309.75
C_AAD025002-tax	1041	318	4	1	3.57	502	3	19	1	260.25
C_AAW154003-tax	1007	417	4	1	3.79	490	3	14	1	251.75
C_AKJ313002-tax	977	403	4	1	3.79	477	3	12	1	244.25
C_AAD640002-tax	879	329	4	1	3.7	420	3	20	1	219.75
C_AKK397002-tax	863	286	4	1	3.62	416	3	16	1	215.75
C_AAC286002-tax	701	253	4	1	3.68	339	3	12	1	175.25
C_AAN560003-tax	515	214	4	1	3.8	253	3	5	1	128.75
C_AKJ644002-tax	509	121	4	1	3.41	242	3	13	1	127.25
C_AAE587002-tax	493	189	4	1	3.73	240	3	7	1	123.25
C_AAD170002-tax	451	175	4	1	3.74	221	3	5	1	112.75
C_AAC168002-tax	405	131	4	1	3.58	191	3	12	1	101.25

6 Summary and Conclusions

OntoCAT provides a comprehensive set of metrics for use by the ontology consumer. It may be used to assist in ontology evaluation for re-use or regularly during ontology development and throughout the ontology's lifecycle to record a history of the changes in both the intensional and extensional ontology. It includes either directly or components of many of OntoQA metrics. It differs from AKTiveRank which uses query concepts to rank ontologies. OntoCAT could be used to further analyze the top ranked ontologies produced by AKTiveRank. Numerous ontologies from varying domains: WordNet, UMLS, UNSPSC, and eCl@ss have been analyzed by OntoCAT. Here the results for the two PSCS ontologies have been reported. The metrics identified and implemented as plug-in software for Protégé are the most comprehensive set of metrics currently available in a tool for both kinds of ontologies. The tool still needs more capabilities to summarize the metrics both in intuitive terms and visually for the user. Another useful feature would be producing analysis based on query terms to provide a context on which to calculate more detailed metrics reflecting topic coverage. The structural types of metrics proposed in [7] that do not already exist in OntoCAT are to be further investigated for inclusion in OntoCAT.

References

1. Alani, H. and Brewster, C. Ontology Ranking based on the Analysis of Concept Structures, *International Conference On Knowledge Capture*, Alberta, Canada (2005)
2. Alani, H. and Brewster, C. Metrics for Ranking Ontologies, Fourth International Evaluation of Ontologies for the Web Workshop (EON 2006), Edinburgh, UK, May (2006).
3. Brank, J., Grobelnik, M., and Mladenic, D. A survey of ontology evaluation techniques. In *Proceedings. of the 8th Int. multi-conference Information Society IS-2005,* 2005.
4. Corcho, O., Fernandez-Lopez, M., Gomez-Perez, A., Methodologies tools and languages for building ontologies, Data & Knowledge Engineering 46 41–64 (2003)
5. Cross, V. and Pal, A., Ontology Metrics, *2005 North American Fuzzy Information Processing Society*, Ann Arbor Michigan, July (2005)
6. Ding, L., T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A semantic web search and metadata engine. In *Proc. 13th ACM Conf. on Information and Knowledge Management*, Nov. (2004)
7. Gangemi, A., Catenacci, C., Massimiliano, C. and Lehmann, J., Ontology Evaluation and Validation: An integrated formal model for the quality diagnostic task, On-line: http://www.loa-cnr.it/Files/OntoEval4OntoDev_Final.pdf (2005).
8. Gomez-Perez, A. Evaluating Ontology Evaluation, in Why evaluate ontology technologies? Because it works!, *Intelligent Systems, IEEE*, Volume 19, Issue 4, Jul-Aug 2004.
9. Guarino, N. and Welty, C. Evaluating ontological decisions with OntoClean, *Communications of the ACM*, Volume 45, Number 2, February(2002)
10. Hepp, M., Leukel, J., and Schmitz, V. A Quantitative Analysis of eClass, UNSPSC, eOTD, and RNTD: Content, Coverage, and Maintenance, IEEE International Conference on eBusiness Engineering (ICEBE'05) pp. 572-581 2005.
11. Kalfoglou, Y., Evaluating ontologies during deployment in applications, position statement, .OntoWeb 2 meeting 07-12-2002,
12. Lozano-Tello, A.and Gómez-Pérez, A., ONTOMETRIC: A Method to Choose the Appropriate Ontology, *Journal of Database Management*, 15(2), 1-18, April-June 2004.
13. Miller, G., WordNet: a lexical database for English, *Comm. of ACM* 38 (11), 39–41 (1995)
14. Noy, N., Evaluation by Ontology Consumers, in Why Evaluate ontology technologies? Because it works!, *IEEE Intelligent Systems*, July/August (2004)
15. Orbst, L., Hughes, T., and Ray, S. Prospects and Possibilities for Ontology Evaluation: The View from NCOR, Fourth International Evaluation of Ontologies for the Web Workshop (EON 2006), Edinburgh, UK, May (2006).
16. Pal, A. An Ontology Analysis Tool For Consumers, Masters Thesis, Miami University, Oxford, OH May (2006).
17. Poulin J.S. Measuring Software Reuse: Principles, Practices, and Economic Models. Addison Wesley Longman, (1997)
18. Sabou, M., Lopez, V, Motta, E. and Uren, V. Ontology Selection: Evlauation on the Real Semantic Web, Fourth International Evaluation of Ontologies for the Web Workshop (EON 2006), Edinburgh, UK, May (2006).
19. Seco, N.,Veale, T. and Hayes, J. An Intrinsic Information Content Metric for Semantic Similarity in WordNet, *ECAI 2004*, 1089-1090 (2004)
20. Tartir, S. Arpinar, I.B., Moore, M., Sheth, A.P. and Aleman-Meza, B. OntoQA: Metric-Based Ontology Quality Analysis, *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, Houston, TX, USA, November 2005.
21. Zhang, Y., W. Vasconcelos, and D. Sleeman. Ontosearch: An ontology search engine. In *Proc. 24th SGAI Int. Conf. on Innovative Techniques and Applications of Artifial Intelligence*, Cambridge, UK (2004).

Improving the recruitment process through ontology-based querying

Malgorzata Mochol
Freie Universität Berlin
Institut für Informatik
AG Netzbasierte Informationssysteme
Takustr. 9, 14195 Berlin
Germany
mochol@inf.fu-berlin.de

Holger Wache
Vrije Universiteit Amsterdam
Artificial Intelligence Department
de Boelelaan 1081a, 1081HV Amsterdam
The Netherlands
holger@cs.vu.nl

Lyndon Nixon
Freie Universität Berlin
Institut für Informatik
AG Netzbasierte Informationssysteme
Takustr. 9, 14195 Berlin
Germany
nixon@inf.fu-berlin.de
http://nbi.inf.fu-berlin.de

Abstract: While using semantic data can enable improved retrieval of suitable jobs or applicants in a recruitment process, cases of inconsistent or overly specific queries which would return no results still have to be dealt with. In this paper the extension of a semantic job portal with a novel query relaxation technique is presented which is able to return results even in such cases. Subsymbolic methods estimate the (quantitative) similarity between job or applicant descriptions. Symbolic approaches allow a more intuitive way to formulate and handle preferences and domain knowledge. But due to their partial preference order they can not rank all results in practice like subsymbolic approaches. In this paper we propose a query relaxation method which combines both methods. This method demonstrates that by having data based on formal ontologies, one can improve the retrieval also in a user-friendly way.

1 Introduction

Human resources management, like many business transactions, is increasingly taking place on the Internet. In Germany 90% of human resource managers rated the Internet as an important recruitment channel [Ke05] and over half of all personnel recruitment is the result of online job postings [Mo03]. Although job portals are an increasingly important source for job applicants and recruitment managers, they still exhibit shortcomings in retrieval and precision as the stored job offers are in syntactic formats, i.e. searches are subject to the ambiguities of natural language and job descriptions and characteristics lack relations to similar or interdependant concepts. Particularly, queries which are over-

specified or inconsistent return no matches while relevant job offers could actually still be found if the consistency or specificity problem were to be resolved. Extending job offers in a portal with semantics can enable improved search and precision based on the use of ontologies and semantic matchings. A further extension of such a semantics-based job portal with novel query relaxation techniques additionally solves the problem of inconsistent and overspecified querying, which can be expected to occur in a real world setting.

In this paper we report on such results from the German national project *Wissensnetze* (Knowledge Nets[1]) and *European Network of Excellence Knowledge Web*[2]. Having identified the importance of online recruitment processes for businesses today, Section 1.1 will introduce requirements that arise when seeking to match appropriate applicants to vacancies or vice versa. *Wissensnetze* is working together with a German job portal provider to develop an innovative new approach to online recruitment based on the use of semantically annotated job and applicant descriptions. In Section 2 we introduce the web-based prototype of a semantic job portal that has been developed in this project. To meet the identified requirements, we introduce two semantic-based approaches which rely on the formal logical basis of ontology languages. Semantic matching techniques (Section 3) are applied to determine the level of similarity between a job and an applicant. However, we note that this technique alone can not meet real world requirements, which include the possibility of overly specific or inconsistent queries. Hence, in a co-operation supported by *Knowledge Web*, we are extending the prototype with state of the art query relaxation techniques (Section 4). As a result we are in a position to evaluate both the emerging semantic technique as well as the real world viability of the semantic job portal (Section 5). In general, we are able to demonstrate the value of ontology-based approaches to a typical business activity as well as the added value of the use of a formal logic based approach in that new semantic techniques are easily applicable to Semantic Web systems to solve open requirements.

1.1 Requirements

Currently, a large number of online job portals divide the online labour market into information islands and making it close to impossible for a job seeker to get an overview of all relevant open positions. In spite of a large number of portals employers still publish their openings on a rather small number of portals assuming that a job seeker will visit multiple portals while searching for open positions. Alternatively, companies can publish job postings on their own website [Mü00]. This way of publishing, however, makes it difficult for job portals to gather and integrate job postings into their database. Furthermore, the quality of search results depends not only on the search and index methods applied but also on the processability of the used web technologies and the quality of the automatic interpretation of the company-specific terms occurring in the job descriptions. The deficiencies of a website's machine processability result from the inability of current web technologies, such as HTML, to semantically annotate the content of a given website. Therefore, com-

[1] http://wissensnetze.ag-nbi.de
[2] http://knowledgeweb.semanticweb.org

puters can easily display the content of a HTML site, but they lack the ability to interpret the content properly.

In our opinion using Semantic Web technologies in the domain of online recruitment can overcome the problems of distribution, heterogeneity and machine non-processability of existing job information and substantially increase market transparency, lower transaction costs and speed up the procurement process for businesses. For this reason, in *Wissensnetze* we have developed job portal which is based on Semantic Web technologies as a basis for exploring the potential of the Semantic Web from a business and a technical viewpoint by examining the effects of the deployment of Semantic Web technologies for particular application scenarios and market sectors. Every scenario includes a technological component which makes use of the prospected availability of semantic technologies in a perspective of several years and a deployment component assuming the availability of the required information in machine-readable form. The combination of these two projections allows us, on the one hand, to build e-business scenarios for analysis and experimentations and, on the other hand, to make statements about the implications of the new technology on the participants of the scenario in the current early stage of development.

In the first version of the *semantic job portal* we concentrated on the modelling of the human resource ontology and development of a matching approach for comparisons of applicant profiles and job openings with focus on skills, occupations as well as industry sector descriptions [BHM+05]. The further specification of the complete job portal contains the comparison of the applicants and openings not only under consideration of skills and their levels but also professional experiences of a job seeker in relation to the requirements of the hiring company. We want to express not only the duration of a particular experience (3 years java programming) but also to deliver these job applications which maybe do not fit in 100% to the defined requirements but are still acceptable for the employer (3 years instead of 5 years industry experience). Furthermore, to verify the consistency of the job opening descriptions we also have to avoid the definition of nonsensical requirements like job postings which demand only very young (under 25) yet highly qualified people (with at least 10 years work experience). Following this, we need an additional method which starts checking the data with the strongest possible query that is supposed to return the "best" answers satisfying most of the given conditions and then weaken the query if the returned result set is either empty or contains unsatisfactory results.

Since we have been working very close with one German job search engine we were able to define several exemplary use cases which focuses on the definition of such requirements and issues. From the practical point of view the use-cases may represent the kind of queries which happen in the real world. However from the scientific point of view these use-cases are challenges to the techniques which we want to apply.

When implementing a (Semantic Web) job portal the requirements of the system depend on the meaningful use cases which are derived by the industrial project partner from its day to day business practices within the HR-domain. To clarify the still outstanding problems we will briefly present one of such use cases which (at first view) seems to be quite simple. However if we look closer and try to represent the data in an ontology or satisfy the requirements in the semantic portal we will meet some difficulties which at the same time show the complexity of such "simple" queries.

We are looking for a person which:

1. has an degree in computer science
2. wants to work in software consulting and development,
3. is an expert in C, Java, PHP, UML, .Net and WindowsNT,
4. has worked for at least 5 years in an industrial and 5 year in a research project,
5. should have experience as project or team manager,
6. should be not older then 25

This example serve as a guideline and a thread in the rest of the article.

2 Semantic Job Portal

In a Semantic Web-based recruitment scenario the data exchange between employers, applicants and job portals is based on a set of vocabularies which provide shared terms to describe occupations, industrial sectors and job skills [MPB06]. In this context, the first step towards the realization of the Semantic Web e-Recruitment scenario was the creation of a *human resources ontology (HR-ontology)*. The ontology was intended to be used in a job portal by allowing not only a uniform representation of job postings and job seeker profiles but first of all the semantic matching (a technique that combines annotations using controlled vocabularies with background knowledge about a certain application domain) in job seeking and procurement tasks. Another important requirement was development of the Semantic Web-based job portal concerning the user needs under consideration of the already common practice in the industry. Accordingly to this specification we focused on how vocabularies can be derived from standards already in use within the recruitment domain and how the data integration infrastructure can be coupled with existing non-RDF human-resource systems.

In the process of ontology building we first identified the sub-domains of the application setting (skills, types of professions, etc.) and several useful knowledge sources covering them (approx. 25)[BMW04]. As candidate ontologies we selected some of the most relevant classifications in the area, deployed by federal agencies or statistic organizations: German Profession Reference Number Classification (BKZ), Standard Occupational Classification (SOC), German Classification of Industrial Sector (WZ2003), North American Industry Classification System (NAISC), German version of the Human Resources XML (HR-BA-XML) and Skill Ontology developed by the KOWIEN Project[SBA03]. These sources are represented in different formats and languages with various levels of the formality (textual descriptions, XML-schemes, DAML) and cover different domains at different precision levels. Since these knowledge sources were defined in different languages (English/German) we first generated (depending on the language) lists of concept names. Except for the KOWIEN ontology, additional ontological primitives were not supported by the candidate sources. In order to reduce the computation effort required to compare

and merge similar concept names we identified the sources which had to be completely integrated to the target ontology. For the remaining sources we identified several thematic clusters for further similarity computations. For instance BKZ was directly integrated to the final ontology, while the KOWIEN skill ontology was subject of additional customization. To have an appropriate vocabulary for a core skill ontology we compiled a small conceptual vocabulary (15 concepts) from various job portals and job procurement Web sites and matched them against the comprehensive KOWIEN vocabulary. Next, the relationships extracted from KOWIEN and various job portals were evaluated by HR experts and inserted into the target skill sub-ontology. The resulting conceptual model was translated mostly manually to OWL (since except for KOWIEN the knowledge sources were not formalized using a Semantic Web representation language) [PBM05].

We have been still evaluating and refining the preliminary HR-ontology for the purpose of further development and to calculate the costs of reusing existing sources. About 15% of the total engineering time were spent on source gathering and about 30% were spent on customizing the selected source ontologies. Several ontologies have been fully integrated into the resulting ontology, while KOWIEN and the XML-based sources required additional customization effort. Although the entire integration took up over 45% of the total engineering time the reusing classification schemes like BKZ or WZ2003, which did not require any customization effort, definitely resulted in significant cost savings, while guaranteeing a comprehensive conceptualization of occupations and industrial sectors, respectively. The last phase of the building, refinement and evaluation process will require 10% of the overall time. The aggregation of knowledge from different domains and the evaluation of available, large-sized ontologies were tedious and time-consuming. Although, the benefits of using standard classification systems in this application setting outweigh the costs of the ontology reuse. The reuse process could be significantly optimized in terms of costs and quality of the outcomes if provided the necessary technical support.

Having modelled the HR-ontology and prepared the RDF-Repository to store applicant profiles and job description, we developed the matching engine which as the core component of the system plays the crucial role in the procurement process. The function of the matching engine is focused on in the following chapter.

3 Semantic matching

Semantic matching is a technique which combines annotations using controlled vocabularies with background knowledge about a certain application domain. In our prototypical implementation, the domain specific knowledge is represented by concept hierarchies like skills, skill level classification, occupational classification, and a taxonomy of industrial sectors. Having this background knowledge of the recruitment domain (i.e. formal definition of various concepts and specification of the relationships between these concepts) represented in a machine-understandable format allows us to compare job descriptions and applicant profiles based on their semantic similarity [PC95] instead of merely relying on the containment of keywords like most of the contemporary search engines do.

In our HR-scenario, our matching approach[3] utilizes metadata of job postings and candidate profiles and as the matching result, a ranked list of best candidates for a given job position (and vice versa) is generated.

Inside both a job posting as well as an applicant profile we group pieces of information into "*thematic clusters*", e.g. information about skills, information regarding industry sector and occupation category, and finally job position details like salary information, travel requirement, etc. Each thematic cluster from a job posting is to be compared with the corresponding cluster from an applicant profile (and the other way round). The total similarity between a candidate profile and a job description is then calculated as the average of the cluster similarities. The *cluster similarity* itself is computed based on the similarities of semantically corresponding concepts from a job description and an applicant profile. The *taxonomic similarity* between two concepts is determined by the distance between them which reflects their respective positions in the concept hierarchy. Following this, the distance `d` between two given concepts in a hierarchy e.g. `.Net` and `DCOM` (cf. Fig. 1) represents the path from one concept to the other over the closest common parent. The semantic differences between upper level concepts are bigger than those between concepts on lower hierarchy levels (in other words, two general concepts like `object oriented programming languages` and `imperative procedural programming languages` are less similar than two specialized like `C#` and `Java`) and the distance between siblings is greater than the distance between parent and child (`d(Java,C#) > d(Java,PureObjectOrientedLanguages)`).

Since we also provide means for specifying *competence levels* (e.g. expert or beginner) in applicants profiles as well as job postings we compare these levels in order to find the best match. Furthermore, our approach also gives employers the opportunity to specify the importance of different job requirements. The concept similarity is then justified by the indicated weight (i.e. the similarity between more important concepts) like the skills crucial for a given job position and will have greater influence on the similarity between a job position posting and an applicant profile.

Having the example from Section 1.1 we can apply the developed semantic matching engine to compare the requirements defined within a job opening with the job applicant descriptions (and another way round comparing the applicants profiles with the job descriptions). The results of the comparisons are presented in the form of a ranked list where each applicant profile can be separately viewed and considered in detail (cf. Fig. 1).

The approach described above allows comparisons of job openings with applicant profiles based on verifying occupation and industry descriptions, skills and their competence levels as well as some general information like salary and job type. Hence, the prototype can satisfy the first three points of the specification from the above mentioned job description (cf. Sec. 1.1) we are not able to deliver an answer to the other requirements especially concerning the minimal required experiences in different projects or experiences as team manager. To tackle this problem we have decided to extend our prototype by applying not only the semantic matching technique but also using the query relaxation methods to compare the job description with applicants.

[3] For further information about used matching framework SemMF see [OB05]

Comparison: Application - Job offer

Matching result (71.6% Total similarity)

Details of job offer (59.0% similarity, weighted with 0.2)

	Job offer	Applicant	Similarity
Travel desired:	no	no	100.0%
Job type:	part time	full time	0.0%
Salary:	35000 € / year	39000 €/year	77.1%

Details of occupation (88.0% similarity, weighted with 0.2)

	Job offer	Applicant	Similarity
Industrial sector:	Software consulting and development	Software consulting	100.0%
Occupation:	Computer sciences assistant	Computer sciences assistant - Software technology	76.0%

Required Competencies (70.3% similarity, weighted with 0.6)

Required competencies	Available competencies	Similarity
C (Expert)	C (Medium knowledge)	82.1%
DotNET (Expert)	DCOM (Beginner)	80.1%
WindowsNT (Expert)	CGI (Expert)	59.8%
UML (Expert)	XML (Beginner)	68.2%
Java (Expert)	C (Medium knowledge)	82.1%
PHP (Expert)	CGI (Expert)	59.8%

Figure 1: Matching result

4 Query relaxation

The previous approach basically uses a similarity measure which calculates the similarity between the job posting and candidate profile. Such a function $f(jp, cp) \mapsto [0..1]$ directly provide a ranking between the results because answers which are more similar can be higher ranked. In order to ensure that the job portal ranks certain answers higher than others similarity measures normally can be biased in that way that weights w_i are attached to some parts of the calculation, i.e. $f(p, r) = \sum_{i=1}^{n} w_i * f_i(p, r)$. With such weights users also can ensure that the system will respect his preferences during the ranking.

However, similarity functions also have their drawbacks. Like all subsymbolic approaches, similarity functions do not explain *how* the job posting and the request differ. They only return a value like 0.78 but they do not provide any information how the job posting and the candidate profile differ in detail. For example, they can not explain that the candidate has only three years experiences instead of requested five years.

Furthermore the similarity function is also not able to explain the differences betweens the answers. Another answer with a very close similarity value, say 0.79, may give the impression of a similar good candidate but it may differ on an absolutely different thematic cluster, e.g. the candidate has no experiences in leading a project. The difference with this answer is not obvious and is not explained. The similarity function suggests a ranking

but in fact the result is an unordered listing; a grouping of semantically similar answered would improve the acceptance of and the usability by the user.

On the other hand the user is directly able to specify how he wants to relax his request. The user may specify directly: "if nobody have 5 years industrial experiences then I will also accept 3 years experiences". Furthermore the system can also explain how this set of returned answers is related to the original query, e.g. here comes now the answers not with 5 but with 3 years experiences (cf. Section 1.1).

Such preferences can be specified in symbolic approaches. Advanced methods like [BBD+04] also allow conditional preferences where the preferences depends on some other decisions. However most of these approaches assume implicitly a flat data structures like e.g. a set of variables [BBD+04]. But here we are focussed with the need of the full expressive power of advanced object-centred representation of job descriptions and candidate profiles which may be difficult to encode in symbolic approaches.

In the following we describe an approach which use rewriting rules to capture preference and domain knowledge explicitly and show how this knowledge is used to relax the original query into a set of approximated queries. Rewriting rules are able to work on complex data structures. We propose an approach for query rewriting based on conditional rewriting rules to solve the problem of incorporating domain knowledge and user preferences for similar matching job request and openings. This rewriting relaxes the over-constrained query based on rules in an order defined by some conditions. This has an advantage that we start with the strongest possible query that is supposed to return the "best" answers satisfying most of the conditions. If the returned result set is either empty or contains unsatisfactory results, the query is modified either by replacing or deleting further parts of the query, or in other words relaxed. The relaxation should be a continuous step by step, (semi-)automatic process, to provide a user with possibility to interrupt further relaxations.

Query rewriting with rewriting rules helps to incorporate domain knowledge and user preferences in the semantic matching in an appropriate way. It comes back with a set of rewritten queries. However the results of each rewritten query may be a set of answers which is not ordered. How to order or rank them? For this part we can fall back to the similarity function with their implicitly encoded knowledge and rank the answers for each rewritten query. To summarize query rewriting provides a high level relaxation including grouping the results according the domain knowledge and the user preferences and the similarity function provides some kind of fine tuning when the results in one group is ranked.

Before we investigate concrete relaxation strategies in the context of our example domain, we first give a general definition of the framework for re-writing an RDF query very briefly.

4.1 Formal definition

The RDF data model foresees sets of statements which are in the form of triples [Ha04]. In [DSW06] Dolog et.al. proposed a rule-based query rewriting framework for RDF queries independent of a particular query language which we summarize here. The framework is based on the notion of triple patterns (RDF statements that may contain variables) as the

basic element of an RDF query and represents RDF queries in terms of three sets:

- triple patterns that must be matched by the result (mandatory patterns)
- triple patterns that may be matched by the results (optional triple patterns).
- Conditions in terms of constraints on the possible assignment of variables in the query patterns.

More precisely Dolog et.al. define a (generic) RDF query as a triple of these three sets.

Definition 1 *RDF Query*
Let $\mathcal{T}$ be a set of terms, $\mathcal{V}$ a set of variables, $\mathcal{RN}$ a set of relation names, and $\mathcal{PN}$ a set of predicate names. The set of possible triple patterns $\mathcal{TR}$ is defined as $\mathcal{TR} \subseteq (\mathcal{T} \cup \mathcal{V}) \times (\mathcal{RN} \cup \mathcal{V}) \times (\mathcal{T} \cup \mathcal{V})$. A query Q is defined as the tuple $\langle M_Q, O_Q P_Q \rangle$ with $M_Q, O_Q \in \mathcal{TR}$ and $P_Q \subseteq \mathcal{P}$ where M_Q is the set of mandatory pattern (patterns that have to be matched by the result, O_Q is a set of optional pattern (patterns that contribute to the result but do not necessarily have to match the result) and $\mathcal{P}$ is the set of predicates with name $\mathcal{PN}$, defined over $\mathcal{T}$, and $\mathcal{V}$.

A result set of such a RDF query is set of substitutions. Formally a substitution τ is a list of pairs (X_i, T_i) where each pair tells which variable X_i has to be replaced by $T_i \in \mathcal{T} \cup \mathcal{V}$. A ground substitution replaces variables X_i by a term and not by another variable, i.e. $T_i \in \mathcal{T}$ for all i. The (ground) substitution τ replaces variables in M_Q and O_Q with appropriate terms. If $\tau(M_Q)$ is equal to some ground triples then the substitution is valid. All valid ground substitutions for M_Q plus existing ground substitutions for O_Q constitute answers to the query. Additionally the predicates P_Q restrict these substitutions because only those bindings are valid answers where the predicates, i.e. $\tau(P_Q)$, are also satisfied. The predicates additionally constraint the selection of appropriate triples.

Re-writings of such queries are described by transformation rules $Q \xrightarrow{R} Q^R$ where Q the original and Q^R the rewritten query generated by using R. Rewriting rules consists of three parts:

- A matching pattern represented by a RDF query in the sense of the description above
- A replacement pattern also represented by an RDF query in the sense of the description above
- A set of conditions in terms of special predicates that restrict the applicability of the rule by restricting possible assignments of variables in the matching and the replacement pattern.

Based on the abstract definition of an RDF query, we can now define the notion of a rewriting rule. We define rewriting in terms of rewriting rules that take parts of a query, in particular triple patterns and conditions, as input (PA) and replace them by different elements (RE).

Definition 2 *Rewriting Rule*
A rewriting rule R is a 3-tuple $\langle PA, RE, CN\rangle$ where PA and RE are RDF queries according to Definition 1 and CN is a set of predicates.

For conditions the same constructs as for queries are used where the possible results are also constrained by predicates. Patterns and replacements formally have the same structure like queries. They also consist of a set of triples and predicates. But patterns normally do not address complete queries but only a subpart of a query. Normally the subpart addresses some triples as well as some predicates in the query. In order to write more generic rewriting rules the pattern must be instantiated which is done by an substitution.

Definition 3 *Pattern Matching*
A pattern PA of a rewriting rule R is applicable to a query $Q = \langle M_Q, O_Q, P_Q\rangle$ if there are subsets $M'_Q \subseteq M_Q$, $O'_Q \subseteq O_Q$ and $P'_Q \subseteq P_Q$ and a substitution θ with $\langle M'_Q, O'_Q, P'_Q\rangle = \theta(PA)$.

In contrast to term rewriting systems [BN98] the definition of a query as two sets of triples and predicates differentiate the pattern matching. The identification of the right subpart of the query for the pattern match is simplified because of the use of sets. Only a subset of both sets has to be determined which must be syntactically equal to the instantiated pattern. Please note that due to set semantics, the triples and predicates in the pattern may be distributed over the query.

A re-writing is now performed in the following way: If the matching pattern matches a given query Q in the sense that the mandatory and optional patterns as well as the conditions of the matching pattern are subsets of the corresponding parts of Q then these subsets are removed from Q and replaced by the corresponding parts of the replacement pattern. The application of R is only allowed if the predicates in the conditions of R are satisfied for some variable values in the matching and the replacement pattern.

Definition 4 *Query Rewriting*
If a rewriting rule $R = \langle PA, RE, CN\rangle$

- *matches a query $Q = \langle M_Q, O_Q, P_Q\rangle$ with subsets $M'_Q \subseteq M_Q$, $O'_Q \subseteq O_Q$ and $P'_Q \subseteq P_Q$ substitution θ and*
- *$\theta(CN)$ is satisfied*

then the rewritten query $Q^R = \langle M^R_Q, O^R_Q, P^R_Q\rangle$ can be constructed with $M^R_Q = (M_Q \setminus M'_Q) \cup \theta(M_{RE})$, $O^R_Q = (O_Q \setminus O'_Q) \cup \theta(O_{RE})$ and $P^R_Q = (P_Q \setminus P'_Q) \cup \theta(P_{RE})$ with $RE = \langle M_{RE}, O_{RE}, P_{RE}\rangle$.

The former definition clarifies formally how to generate a rewritten query Q^R from Q with the help of R, i.e. $Q \xrightarrow{R} Q^R$. We denote with $Q^{\mathcal{R}}$ all queries which can generated from Q with all rules $R \in \mathcal{R}$. On each query from $Q^{\mathcal{R}}$ the rewriting rules can be applied again. We denote with $Q^{\mathcal{R}*}$ all queries which can be generated by application of the rules in $\mathcal{R}$ successively.

4.2 Application in the job portal

Each job request and opening is annotated with an RDF description which is a set of triples. A query over these job openings is formulated as triple patterns and a set of conditions that restrict the possible variables bindings in the patterns. Each triple pattern represents a set of triples. The corresponding abstract definition of a query focuses on the essential features of queries over RDF.

To clarify the approach we take the example 1 from the Section 1: someone who has experience in C, Java, PHP, UML, .Net and WindowsNT. Looking for such a person requires from the system to translate this free text description into an instance retrieval problem. The query must be translated into a concept expression. The retrieval process will return all job seekers which belong to that concept expression, i.e. satisfying all the requirement in the concept expression. The following OWL[4] expression shows the concept expression for some person who has experience in some of (the intersectionOf property) the OWL classes C, Java, PHP or UML [5] .

```
<owl:Class rdf:ID="Query">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Person"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom>
        <owl:Class>
          <owl:intersectionOf rdf:parseType="Collection">
            <owl:Class rdf:about="C"/>
            <owl:Class rdf:about="Java"/>
            <owl:Class rdf:about="PHP"/>
            <owl:Class rdf:about="UML"/>
          </owl:intersectionOf>
        </owl:Class>
      </owl:someValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasExperience"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

In the following we give some examples for the rewriting rules which use the aforementioned example as a basis.

[4] OWL is an extension of RDF allowing for more expressive features than RDF like number restrictions etc.

[5] Originally we modelled these as nominals (enumerations like Week =Monday, Tuesday, ...). Nominals are instances and classes at the same time. However current DL systems have problems with nominals therefore we use classes in the current approach.

A very simple rewriting rule takes into account the required skill e.g. Java. It relax the some requirements in the experiences, i.e. instead of `JAVA` the `PureObjectOrientedLanguages` or even the `ObjectOrientedLanguages` could be possible weakenings of the original query:[6]

```
pattern(<owl:Class rdf:about="Java"/>)
==>
replace(<owl:Class rdf:about="PureObjectOrientedLanguages"/>)
&& true.
```

This means that whenever anywhere the term representing the Java language in a query appears then it can be replaced by a more general term representing pure object oriented languages, of which Java is one.

Making use of the predicates we can generalize previous rewriting rules and generate generic rules that are guided by information from the ontology. The predicate `subsumed` for example is satisfied when X is more specific than Y. With the following rewriting rule we are able to consider the knowledge in the ontology.

```
pattern(<owl:Class rdf:about="X"/>)
==>
replace(<owl:Class rdf:about="Y"/>)
&& subsumed(X,Y).
```

In the same way some number restrictions can be applied. In our example the requirement that a person has experiences in a five year industrial project is encoded with the help of the (artificial) class `FiveYearsOrMore`. This class represents all Numbers representing years which are larger or equal to five. This class can be replaced by the class `TwoYearsOrMore` which obviously is more general (weaker) then the former. Furthermore we can restrict the replacement in that way that we only allow this for the restriction on property `hasDuration`. The corresponding rewriting rule look like:

```
pattern(<owl:Restriction>
           <owl:onProperty rdf:resource="#hasDuration"/>
           <owl:someValuesFrom>
                <owl:Class rdf:ID="FiveYearsOrMore"/>
           </owl:someValuesFrom>
        </owl:Restriction>)
==>
replace(<owl:Restriction>
            <owl:onProperty rdf:resource="#hasDuration"/>
            <owl:someValuesFrom>
                <owl:Class rdf:ID="TwoYearsOrMore"/>
            </owl:someValuesFrom>
        </owl:Restriction>)
&& true.
```

[6] For the sake of readability the examples are simplified.

The main problem of the re-writing approach to query relaxation is the definition of an appropriate control structure to determine in which order the individual rewriting rules are applied to general new queries. In other words how to explore $Q^{\mathcal{R}*}$. Different strategies can be applied to deal with the situation where multiple re-writings of a given query are possible. Example is a Divide and Conquer strategy: The best results of each possible combinations of re-writings is returned. In the current version of the system we have implemented a simple version with similarities to skylining [KK02, LL87] which is well-known in database query relaxation.

In particular, we interpret the problem of finding relaxed queries as a classical search problem with small adaptations. The search space is defined by the set $Q^{\mathcal{R}*}$ of all possible queries. Each application of a rewriting rule R on a query Q is a possible action denoted as $Q \xrightarrow{R} Q^R$. A query represents a goal state in the search space if it does have answers. In the current implementation we use breadth-first search for exploring this search space. Different from classical search, however, the method does not stop when a goal state is reached; each goal state has to be determined because each goal state represent one sequence of successful rewriting when answers are provided. However a goal state need not to be explored further (i.e. no further re-writings must be applied to a goal state) but each search branch has to be closed by a goal state (or by a predefined depth). Breadth-first search ensures that each goal state represents the best solution to the relaxation problem with respect to a certain combination of re-writings. The goal states form a "skyline" for the rewriting problem and each of them is returned to the user together with the query answers.

The second difference to classical search is that we do not allow the same rule to be applied more than once with the same parameters in each branch of the search tree. The only kind of rules that can in principle be applied twice are rules that add something to the query (Rules that delete or replace parts of the query disable themselves by removing parts of the query they need to match against). Applying the same rule that extend the query twice leads to an unwanted duplication of conditions in the query that do not change the query result, but only increase the complexity of query answering.

5 Conclusions and Future Work

We have shown the use of semantic techniques in a prototype job portal in which both job offers and applicants are described according to ontologies. While preparing the ontologies from non-ontological sources (classification schemes) was time consuming and tedious, by doing so we are enabled to use ontology-based querying approaches to match relevant job applicants to vacancies and vice versa, as well as rank them in terms of similarity. Furthermore, we have shown that semantic matching alone does not allow for levels of similarity to be differentiated and for inconsistent or overly specific queries to be resolved. Hence we introduce another technique called query relaxation, in which queries can be rewritten to allow similarity to be ranked in different directions (e.g. in terms of subjects the applicant has experience in, or the number of years total experience he or she

has) or to widen the scope of the query in order to find matches (e.g. by searching on a superclass of the class searched for, or by making cardinalities less restrictive).

At present, the semantic job portal demonstrates improved precision and recall in the semantic matching, finding relevant job offers or applicants which would not be selected by a syntactic (text-based) query algorithm. However we have noted that alone this does not resolve more complex queries which can be expected in the real world. In the *European Network of Excellence Knowledge Web* the co-operation of leading Semantic Web researchers with selected industry partners with a real world business problem to solve is being supported. One of the first steps taken in the network was to collect industrial use cases where semantic technologies could form a potential solution, as well as derive from those use cases industry requirements for Semantic Web research [LPNS05]. The recruitment scenario was one of the industrial use cases provided and was identified by Semantic Web researchers as an ideal real world use case to test their results in query relaxation techniques. Within the Network the job portal is now being extended to support the rule rewriting approach described in this paper.

Our intention is to test the extended prototype against the original prototype (which supports only the semantic matching) using a set of benchmark queries. The HR ontology has been extended for this purpose with the property of experience and instances using this new property added to the semantic data used by the job portal. The rule rewriting tool has been implemented and an interface to the tool which is more general than the DIG interface used by most Semantic Web reasoners (in order to not limit ourselves to Description Logics) has been specified. We have also defined the first concrete technical details of rule rewriting. We plan to carry out the first tests at the beginning of 2007. This will provide us with a valuable real world test case to analyse the value of query relaxation techniques as an extension to semantic matching in ontology-based systems, in order to solve real world problems of inconsistent or overly specific queries. Performance is also be measured as another advantage of query relaxation is expected to be allowing more robust and efficient querying upon large knowledge bases which can scale to real world enterprise size.

Further research issues include determining a general guideline for the specification of rewriting rules and a generic framework for working with those rules. In combination, we believe this work not only demonstrates the use and value of Semantic Web techniques in a real world industrial test case, it indicates that any assessment of the cost of shifting to an ontology-based approach must also take into account the value to be gained from the availability of semantic techniques that is made possible when system data is based on formal logic through an ontology. In this paper we have introduced two such techniques: semantic matching and query relaxation, which are shown to be of value to the online recruitment process.

Acknowledgement The work described in this paper is supported by the EU Network of Excellence KnowledgeWeb (FP6-507482) and Knowledge Nets Project which is a part of the InterVal-Berlin Research Centre for the Internet Economy funded by the German Ministry of Research BMBF.

References

[BBD+04] Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H., und Poole, D.: Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of AI Research*. 21:135–191. 2004.

[BHM+05] Bizer, C., Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R., und Eckstein, R.: The Impact of Semantic Web Technologies on Job Recruitment Processes. In: *International Conference Wirtschaftsinformatik (WI'05)*. 2005.

[BMW04] Bizer, C., Mochol, M., und Westphal, D. Recruitment, report. April 2004.

[BN98] Baader, F. und Nipkow, T.: *Term rewriting and all that*. Cambridge University Press. New York, NY, USA. 1998.

[DSW06] Dolog, P., Stuckenschmidt, H., und Wache, H.: Robust query processing for personalized information access on the semantic web. In: *7th International Conference on Flexible Query Answering Systems (FQAS 2006)*. Number 4027 in LNCS/LNAI. Milan, Italy. June 2006. Springer.

[Ha04] Hayes, P.: RDF Semantics. Recommendation. W3C. 2004.

[Ke05] Keim, T. e. a. Recruiting Trends 2005. Working Paper No. 2005-22. efinance Institut. Johann-Wolfgang-Goethe-Universität Frankfurt am Main. 2005.

[KK02] Kießling, W. und Köstler, G.: Preference sql - design, implementation, experiences. In: *VLDB 2002, Proceedings of 28th International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China*. S. 990–1001. Morgan Kaufmann. 2002.

[LL87] Lacroix, M. und Lavency, P.: Preferences; putting more knowledge into queries. In: Stocker, P. M., Kent, W., und Hammersley, P. (Hrsg.), *VLDB'87, Proceedings of 13th International Conference on Very Large Data Bases, September 1-4, 1987, Brighton, England*. S. 217–225. Morgan Kaufmann. 1987.

[LPNS05] Leger, A., Paulus, F., Nixon, L., und Shvaiko, P.: Towards a successful transfer of knowledge-based technology to European Industry. In: *Proceedings of the 1st Workshop on Formal Ontologies Meet Industry (FOMI 2005)*. 2005.

[Mü00] Mülder, W.: Personalinformationssysteme - Entwicklungsstand, Funktionalität und Trends. *Wirtschaftsinformatik. Special Issue IT Personal*. 42:98–106. 2000.

[Mo03] Monster: Monster Deutschland and TMP Worldwide: Recruiting Trends 2004. In: *2. Fachsymposium für Personalverantwortliche*. Institut für Wirtschaftsinformatik der Johann Wolfgang Goethe-Universität Frankfurt am Main. 2003.

[MPB06] Mochol, M. und Paslaru Bontas, E.: Practical Guidelines for Building Semantic eRecruitment Applications. In: *International Conference on Knowledge Management, Special Track: Advanced Semantic Technologies (AST' 06)*. 2006.

[OB05] Oldakowski, R. und Bizer, C.: SemMF: A Framework for Calculating Semantic Similarity of Objects Represented as RDF Graphs. In: *Poster at the 4th International Semantic Web Conference (ISWC 2005)*. 2005.

[PBM05] Paslaru Bontas, E. und Mochol, M.: Towards a reuse-oriented methodology for ontology engineering. In: *Proc. of 7th International Conference on Terminology and Knowledge Engineering (TKE 2005)*. 2005.

[PC95] Poole, J. und Campbell, J.: A Novel Algorithm for Matching Conceptual and Related Graphs. *Conceptual Structures: Applications, Implementation and Theory*. 954:293–307. 1995.

[SBA03] Sowa, F., Bremen, A., und Apke, S. Entwicklung der Kompetenz-Ontologie für die Deutsche Montan Technologie GmbH. `http://www.kowien.uni-essen.de/workshop/DMT\_01102003.pdf`. 2003.