

An Ontological Approach to Representing Historical Knowledge

by

Vanesa Mirzaee Abar

B.Sc., Shahid Beheshti University, Tehran, Iran, 1996

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
Master of Applied Science

in

THE FACULTY OF GRADUATE STUDIES
(Department of Electrical and Computer Engineering)

We accept this thesis as conforming
to the required standard.

THE UNIVERSITY OF BRITISH COLUMBIA

January 2004

© Vanesa Mirzaee Abar, 2004

Library Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Vanessa Mirzaee Abar
Name of Author (please print)

30/01/2004
Date (dd/mm/yyyy)

Title of Thesis: An ontological Approach to Representing
Historical Knowledge

Degree: Masters of applied science Year: 2004

Department of Electrical and Computer engineering
The University of British Columbia
Vancouver, BC Canada

Abstract

Electronic versions of historical documents are traditionally limited in both the representations they allow as well as the level of computational manipulation they support. Most electronic documents have two minimal advantages over their hard-copy printed counterparts: availability and keyword search. This means that electronic versions of printed matter do not allow the search for concepts, or the relations between concepts, found within a document.

In this work, we present a novel approach to this issue by using ontologies to represent the semantics of the knowledge contained in historical documents. We utilize an evolving methodology to design and build an ontology to represent the information in the book, "*The History of the Iranian Constitution*." After a review of available ontology development environments, we selected Protégé-2000 to formalize and instantiate our ontology. The ontology was later evaluated by utilizing a set of competency questions and motivating scenarios. Our implementation was successful in answering these questions as well as in providing support for the selected scenarios. Our implementation and the evaluation results are presented along with our proposed future work.

Table of Contents

Abstract.....	ii
Table of Contents.....	iii
List of Tables	v
List of Figures.....	vi
1 Introduction.....	1
1.1 Problem Statement and Motivation	1
1.2 Thesis organization	4
2 Background and Related Work.....	6
2.1 Related Work	6
2.1.1 Model Edition Partnership (MEP) project	8
2.1.2 Historical Event Markup and Linking (HEML) project	8
2.1.3 Analysis of these approaches and our selected approach	10
2.2 Knowledge Representation and Ontologies.....	12
2.2.1 Why ontologies?	13
2.2.2 Defining the term “ontology”	16
2.2.3 What do ontologies look like?	19
2.2.4 Ontology classification	22
2.2.5 Methods for building an ontology.....	28
2.2.5.1 Uschold and King Methodology (Enterprise Methodology)	30
2.2.5.2 Gruninger and Fox Methodology (TOVE Methodology).....	37
2.2.5.3 Uschold and Gruninger Methodology (Unified Methodology)	43
2.2.5.4 Gomez and Fernandez Methodology (METHONTOLOGY).....	45
3 Methodology.....	54
3.1 Selected Approach	54
3.1.1 Identify the purpose, scope, and intended users	56
3.1.2 Ontology Building	58
3.1.2.1 Domain analysis and knowledge acquisition.....	58
3.1.2.2 Investigate the possibility of using an existing ontology.....	59
3.1.2.3 Building an informal ontology model.....	60
3.1.2.4 Building a formal ontology model.....	61

3.1.2.4.1	Selecting a development tool.....	62
3.1.2.4.2	Protégé-2000.....	68
3.1.3	Ontology Evaluation.....	80
4	Build a history ontology.....	81
4.1	Identify the purpose, scope, and intended users.....	81
4.2	Ontology building.....	85
4.2.1	Domain analysis and knowledge acquisition.....	85
4.2.2	Building an informal ontology.....	88
4.2.3	Building a formal ontology model.....	105
4.2.3.1	Mapping our domain concepts to a class hierarchy.....	105
4.2.3.2	Assigning properties to classes (attributes & relations).....	107
4.2.3.3	Defining Constraints.....	122
4.2.4	Ontology Evaluation.....	123
5	Conclusions and Future Work.....	139
5.1	Summary of thesis and results.....	139
5.2	Conclusions.....	140
5.3	Future Work.....	140
	Bibliography.....	142

List of Tables

Table 2-1 Enterprise Ontology Terminology.....	36
Table 3-1 a general description and design features of the tools within the OntoWeb framework (based on the work in [WebOnto 2002]).....	63
Table 3-2 Usability features supported (WebOnto framework)	63
Table 3-3 Inference services provided by the tools (WebOnto framework)	64
Table 3-4 Interoperability, knowledge representation, and methodological support comparison (adapted from [WebOnto 2002]).....	65
Table 3-5 Evaluation framework of the WonderTools Project.....	66
Table 3-6 Summary of the results obtained in the WonderTools Project.....	67
Table 3-7 Slot widgets in Protégé-2000.....	70
Table 3-8 Storage back-end plug-ins in Protégé-2000	70
Table 3-9 Tab plug-ins in Protégé-2000	71
Table 4-1 First draft of the competency questions.....	83
Table 4-2 Motivating Scenarios.....	84
Table 4-3 Competency questions related to the notion “PLACE”	85
Table 4-4 Competency questions related to the notion “EVENT”	86
Table 4-5 Competency questions related to the notion “PERSON”	86
Table 4-6 Competency questions related to the notion “DOCUMENT”	87
Table 4-7 Partial view of the glossary of terms	88
Table 4-8 Concept dictionary related to the notion “PLACE”	90
Table 4-9 Concept dictionary related to the notion “PEOPLE”	91
Table 4-10 Concept dictionary related to the notion “EVENT”	91
Table 4-11 Concept dictionary related to the notion “DOCUMENT”	92
Table 4-12 Concept dictionary related to the notion “TIME”	93
Table 4-13 Mapping between the terminology we used and Protégé-2000 terminology	105

List of Figures

Figure 2-1 A sample document using HEML representation language.....	10
Figure 2-2 Merging different ontologies.....	14
Figure 2-3 Integrating several ontologies into one more general ontology.....	15
Figure 2-4 Partial view of a simple newspaper ontology (concept/sub-concept relations)	21
Figure 2-5 Partial view of the newspaper ontology (relations amongst concepts).....	21
Figure 2-6 Constraint imposed on concept property within the newspaper ontology (axiom).....	22
Figure 2-7 CYC ontology	23
Figure 2-8 Sowa's ontology.....	23
Figure 2-9 UMLS.....	25
Figure 2-10 TOVE	25
Figure 2-11 Example of application ontology	26
Figure 2-12 Part of the <i>frame ontology</i> in Ontolingua.....	27
Figure 2-13 Uschold and King Methodology (Enterprise Methodology)	31
Figure 2-14 Gruninger and Fox Methodology (TOVE Methodology).....	38
Figure 2-15 TOVE Ontologies.....	41
Figure 2-16 TOVE organization concept taxonomy.....	42
Figure 2-17 Part of hierarchy of roles in organization ontology (TOVE)	42
Figure 2-18 Gruninger and Uschold Methodology (Unified approach)	44
Figure 2-19 METHONTOLOGY ontology life cycle	48
Figure 2-20 Ontology requirements specification document for the Chemical ontology	50
Figure 2-21 Part of the Chemical ontology.....	51
Figure 2-22 Water-fall, iterative, and evolving approach.....	52
Figure 3-1 Our Ontology Development process.....	55
Figure 3-2 Results for ease of use comparison of the selected tools (WonderTools project)	66
Figure 3-3 Screenshot of the Protégé 2000 development environment.....	72
Figure 3-4 Slots in Protégé-2000	73

Figure 3-5 Facets in Protégé-2000	74
Figure 3-6 Defining axioms in Protégé-2000	76
Figure 3-7 A knowledge acquisition form in Protégé-2000	77
Figure 3-8 Meta Class “:STANDARD CLASS” in Protégé-2000	78
Figure 3-9 An example of user-defined meta-class in Protégé-2000	79
Figure 4-1 Main concepts within our ontology.....	94
Figure 4-2 Overview of main concepts and relations in our history ontology	95
Figure 4-3 Time model in our domain.....	96
Figure 4-4 A CALENDAR DATE within our history ontology.....	97
Figure 4-5 Interrelations amongst concepts related to “PLACE”	98
Figure 4-6 Relations between the concept “PLACE” and other concepts within our ontology	99
Figure 4-7 Concept hierarchy for “AGENT,” “PERSON,” “GROUP OF PEOPLE,” and “COUNTRY”	100
Figure 4-8 Relations between “PERSON” and “GROUP OF PEOPLE” and other concepts in our domain	101
Figure 4-9 Time-based hierarchical relationship between a PERSON and POSITION	102
Figure 4-10 Conceptual model for position hierarchy	102
Figure 4-11 Interrelations amongst events.....	103
Figure 4-12 Association of an “EVENT” with other concepts in our domain	104
Figure 4-13 Concept “DOCUMENT” and its relation with other concepts in the domain	104
Figure 4-14 Our ontology class hierarchy in Protégé-2000.....	106
Figure 4-15 Inheritance of properties and relations by subclasses	107
Figure 4-16 Reification of the relation MEMBERSHIP as a concept	109
Figure 4-17 An instance of the reified relation “MEMBERSHIP”	110
Figure 4-18 Reification of the relation RESIDENCY as a concept.....	110
Figure 4-19 An instance of the reified relation RESIDENCY	111
Figure 4-20 Reification of the relations BIRTH-PLACE and DEATH-PLACE as concepts.....	112
Figure 4-21 An instance of the reified relation DEATH-PLACE	113

Figure 4-22 Reification of the relation TITLE-OF-HONOR as a concept	113
Figure 4-23 An instance of the reified relation TITLE-OF-HONOR	114
Figure 4-24 Reification of the relation INVOLVMENT as a concept	114
Figure 4-25 An instance of the reified relation INVOLVEMENT	115
Figure 4-26 Modeling the temporal place hierarchical organization	117
Figure 4-27 Class implementation of main place concepts in Protégé	118
Figure 4-28 Reified relations used to represent the temporal dynamic organization of places in Protégé	118
Figure 4-29 An instance of the place hierarchy in Protégé	119
Figure 4-30 Reified relations used to capture the dynamicity of the POSITION hierarchy	120
Figure 4-31 Implementation of reified relations for position in Protégé-2000	121
Figure 4-32 An instance of the reified relation POSITION-HIERARCHY and HOLDS- POSITION in Protégé-2000	122
Figure 4-33 Using Protégé-2000 built-in query engine to answer competency questions	124
Figure 4-34 Using the PAL query engine to answer competency questions	125
Figure 4-35 “Knowledge Tree” visualization of the place hierarchy for “Iran”	126
Figure 4-36 Using TGViz visualization tab to browse the hierarchy of places related to country “Iran”	127
Figure 4-37 Results for Competency Question #1	129
Figure 4-38 Results for Competency Question #2	130
Figure 4-39 Results for Competency Question #3	131
Figure 4-40 Results for Competency Question #4	132
Figure 4-41 Results for Competency Question #5	133
Figure 4-42 Results for Competency Question #6	134
Figure 4-43 Results for Competency Question #7	135
Figure 4-44 Results for Competency Question #8	136
Figure 4-45 Results for Competency Question #9	137
Figure 4-46 Results for Competency Question #10	138

Acknowledgements

First of all, I would like to offer my sincere appreciation to my supervisors Dr. Babak Hamidzadeh and Dr. Lee Iverson. I am deeply grateful for their valuable guidance and insightful ideas throughout the entire project. I would also like to thank Dr. Philippe Kruchten for his valuable feedback and opinions as a member of my committee.

I shall make a special thank to Doris Metcalf and Darija Tomasic in the ECE department for their help and support.

Many thanks to my fellow grad students in the UCL Lab. Most importantly I would to thank my family and friends for their everlasting support.

Last but not least, I would like to thank my dear Mario for his love, friendship and his patient support. I have no doubt that without his help I could not go this far.

Vanessa Mirzaee Abar

University of British Columbia

January 2004

To my mother

1 Introduction

1.1 Problem Statement and Motivation

Over the last two decades, we have witnessed the dawn of digital documentation. Electronic versions of traditionally printed matter offer different individuals with diverse interests the opportunity to access a wealth of information in a simple manner. When the knowledge found in these printed documents is captured in electronic forms, additional tasks can be performed on it than is possible with traditional hard-copy versions. The main advantage that the electronic medium provides is the possibility of searching and manipulating the information in innovative ways. With an electronic document, a search is not limited to the index provided by the author or publisher at the back of the book or end of the article. However, current electronic versions of text documents have certain characteristics that impose limitations when we try to represent, access, search and manipulate this information.

Computational representation of knowledge: electronic information exists mostly in natural language text format [Alani, Kim et al. 2003]. This type of information is not well suited for representations that allow capturing anything more than words without meaning. In other words, it cannot be easily computationally manipulated.

Repetitiveness: Information found in electronic forms is usually repetitive. A lot of material related to a particular subject may be found, but often different sources disagree on the structure used to represent it. A good example of this repetitiveness of information is seen in historical biographies. We may find dozens of different books or websites referring to Princess Diana, all of these portraying nearly the same information, but in different forms.

Reuse and Sharing: Many times, the existence of repetitive information is due to the particular format used to represent the available information. Currently, available data is structured in a manner that does not facilitate its reuse, which leads to this repetitiveness. This might be avoided if we had an efficient model to refer to existing information elsewhere [Swartz 2002].

Underutilized affordances: Traditionally, we assume that a document or article will be read in a sequential manner, as has been done since the advent of writing. Up to now, it has been assumed that the main advantage of electronic formats over printed matter is the convenience of being able to find the material without having to physically attain it from a library or other repository. Once we have this information in a digital format, it is unclear as to how the user might interact with this information besides being able to print it and/or read it. New technology has the potential to provide us with additional functionality over traditional printed matter [Hockey 1999].

When the information is stored electronically, it can be organized in many different ways, presented in different formats (graphs, charts, etc.) and used for a variety of purposes, only one of which is creating a printed publication. The additional possibilities that this medium provides for capturing and representing knowledge give the user an opportunity to interact with this knowledge in innovative ways.

Searching for semantic: The Internet provides us with a large amount of easily accessible information. However, this information can be rendered useless if we do not possess the appropriate tools or methods to utilize and understand it.

With existing search engines, it can sometimes be difficult to find a document with the desired information one seeks. Even if we do find this document, it can still be hard to find a particular subject or specific information within the document. In the event that an appropriate document is found and that the specific area of interest within the document is pinpointed, it is still not easy to capture the semantic within this document. Currently, available paradigms representing electronic versions of documents lack an understanding of the semantic within these documents [Palmer 2001].

Queries within a document are usually limited to key-word searches. Relations between concepts within a document cannot be found by using a keyword search; we are only able to find the instances of the concepts contained in the document. For example, two instances, person X and person Y can be easily queried by keyword search; however, unless users read at least some parts of the document, they can not determine whether these two people are related to each other, how this relationship is defined, and during what time period this relationship holds.

Given the time and effort invested in creating electronic information and how sharing of this information has been widely adopted throughout the world, it makes sense to pause and think about these issues and try to find paradigms to better represent and allow access to the available knowledge.

The work presented in this thesis tries to address the issues we have introduced so far in the specific domain of *historical information*. A wealth of historical information has been transmitted into electronic form. This representation of historical data suffers from all the aforementioned limiting characteristics.

In particular, narrative history texts contain certain elements that characterize the genre; they are usually organized along a time line and share similar key concepts such as people, places, events, and so forth. These texts share another significant similarity: they are all about events that take place in the past. Additionally, for any given period of time, these texts are not densely populated.

Typically, these electronic collections are available in text or Hyper Text Markup Language (HTML) data forms exclusively. This is unfortunate since text and HTML data cannot be computationally manipulated in an effective manner.

Considering the characteristics of historical knowledge currently available in an electronic format and the shortcomings of traditional methods used to represent this knowledge, we propose a paradigm in which the semantics and meaning behind the words in a historical text can be captured and represented.

The main goal of this project is to explore the possibility of capturing the meaning (the semantic) behind the words in a natural language history text. We propose a new conceptual paradigm for representing and formulating documents of a historical nature.

Our purpose is to develop a conceptual model that not only represents the content of a historical document, but also illustrates its meaning. In order to capture the semantic, temporality, and dynamicity of this knowledge, we have chosen to use an ontological approach. This approach is used to represent and formulate knowledge in a manner that allows it to be computationally manipulated, shared and reused amongst different applications. Ontologies are conventionally used for semantic-based information

representation and retrieval. Additionally, they can capture the semantic of the concepts to be represented or searched for.

An ontology is a conceptual model, a resource containing knowledge about “concepts” that exist in the world and how they relate to each other. A concept is a symbol for representing meaning with well defined attributes and relations with other concepts [Mahesh 1996].

An important characteristic of the proposed ontology model is its ability to represent a temporally dynamic hierarchy of concepts. This is of particular importance with historical data, since the concepts and the relations between them change and evolve through time. Two concepts related to one another at a certain point of time may not be related to it at another time stamp. Additionally, this model not only captures the relationships between the concepts but also demonstrates the interrelated hierarchal structure within them.

The temporal features of this ontology will enhance our semantic model, since they will allow us to capture the time dimension of the existing relations. This temporal representation gives the relations more realism approaching that which they have in the real world.

1.2 Thesis organization

The remainder of this thesis is organized as follows:

- In Chapter 2, **Background and related work**, we introduce the state of the art in research related to representation of historical information in electronic formats. This research has been mostly carried out by communities interested in the field of the humanities. This is followed by an introduction to our selected approach “ontology engineering approach.” This includes an overview of terms and definitions of key concepts in the field, motivation to use ontologies, ontology classification along with examples of different types of ontologies, and an in depth discussion and analysis of several ontology building methodologies.
- In Chapter 3, we describe our chosen **methodology**. A detailed explanation of the methods and techniques we adopt and use to build our conceptual model (ontology) is provided. This is followed by a presentation of a survey of existing ontology

development tools. As a result of this investigation, we choose Protégé-2000 as the application for building our formal ontology. An in depth description of Protégé, its features and characteristics is provided as well.

- In Chapter 4, we explain the procedure undertaken for **building a history ontology** based on the guidelines described in Chapter 3. The procedure include a description of the domain that the ontology is intended to capture, identifying the purpose, scope, and intended users of the ontology, defining the ontology building process, utilization of the tool, evaluating the resulting ontology and analyzing the experimental results. The results obtained for each step of our ontology development are presented after each stage.
- In Chapter 5, we present our **conclusions and proposed future work**. This includes a summary of the thesis, its main contributions and results obtained.

2 Background and Related Work

In this chapter we first introduce the state of the art of research relating to representing and manipulating historical information in electronic formats. Two examples of current research projects in this area are presented (MEP and HEML). This is followed by a discussion of the virtues and shortcomings of existing approaches for representing historical information in a manner that both captures the semantic behind a document and allows its reuse and sharing. This discussion is followed by the introduction of our selected approach to represent historical knowledge: “ontology engineering.” We proceed to explain our selected approach in detail by presenting an overview of the terms and definitions of key concepts in the field, our motivation to use ontologies, a detailed ontology classification with examples of different types of ontologies, and an in depth discussion and analysis of several ontology building methodologies.

2.1 Related Work

A wealth of historical information has been encompassed in electronic forms and is available through several different resources such as digital libraries. Most of the research relating to these historical documents, or for that matter, documents in any other humanity area, deals with how to create digital copies of these documents and store them in repositories in a manner that facilitates later access to them. This digital media usually integrates meta-data or a notation which provides information about its content [Burchardt 2001]. The major functionality that these electronic documents using meta-data or other types of notation provide is the facility to retrieve the best-matched document to any search request. Many researchers from different fields are involved in projects that try to address these and related problems; this is an active area of research [Hockey 2000].

An example of one such research projects is the Art and Humanities Data Service (AHDS) History project (HDS), which aims to provide a guideline for creating, describing, using and preserving historical digital resources [AHDS; HDS]. The HDS is funded by the Joint Information Systems Committee (JISC) of the UK Higher Education Funding Councils to collect, catalogue, manage, preserve and encourage the re-use of historical digital resources.

This committee provides guidelines on the following:

- Collecting and preserving historical digital resources
- Providing access to collections of historical digital resources held by different organizations
- Developing online data and metadata delivery systems to enhance access to these collections
- Promoting standards in the creation, description, use and preservation of historical digital resources

Assuming that all this historical knowledge preserved in an electronic format is available for retrieval, two questions arise: what happens after retrieving this information? What will the user do or want to do with these documents?

It is reasonable to assume that a user might wish to obtain the semantic behind the words of these documents. Obtaining this kind of information traditionally requires the user to read the document (or at least some part of it). It would be ideal for electronic historical documents provided methods and techniques of posing these types of historical questions, such as those dealing with relationships between characters named in the historical document, the connections between events, the location of these events, the hierarchy of governmental positions at any given time, the changes that these hierarchies undergo throughout time, etc.

The common approach utilized to representing this type of knowledge (inside a historical document) uses mark-up languages such as Standard Generalized Markup Language (SGML) and Extensible Markup Language (XML), which tag the information within the document. To the best of our knowledge, there are no current projects in the research community that focus on generating a representation addressing similar interests similar to our. However, we do find two projects that utilize ideas somewhat similar to ours, although taking different approaches. These projects are presented in the following sections.

2.1.1 Model Edition Partnership (MEP) project

The MEP project involves editors from seven different editorial projects in the history domain. Two of these projects are designed to aid in creating image editions, and the other five help prepare letterpress publications. These documentary editions provide the basic source material for the study of American history, by adding the historical context which makes the material meaningful to readers. Documentary editors prepare the material for publication by transcribing the documents, organizing the sources into a coherent sequence which tells the story (the history) behind them, and annotating these documents with information to help the reader understand them [Hockey 1999].

The goals for this partnership are to develop frameworks for electronic historical editions, to develop computational approaches to create new editions and to provide a series of models and guidelines to be shared amongst editors while preparing the material for publication [Chesnutt 1995].

In this work, a series of models were developed to transform historical documents to electronic versions. This project employs SGML to support access to the information that exists within the documents. This is done by developing a set of SGML Document Type Definitions (DTDs), which define a markup system for publishing historical documents [MEP 2003].

2.1.2 Historical Event Markup and Linking (HEML) project

Another project related to our work is the HEML project. This project proposes a markup scheme to be used for electronic representation of historical events on the web. In this work, XML tags are defined to describe historical events [Robertson 2003].

HEML defines a lightweight XML language that associates web resources with historical events described in terms of time, location, and participants. Figure 2-1 presents a sample of a document that has been transformed into HEML language.

The goal in developing HEML is to use this scheme as a framework to tag documents that record historical events, their date and location. This facilitates assembling a computerized collection of information and associating it with the document. It then

becomes possible to search for a description of events on a certain day, or in a certain region amongst collections that share this same scheme [Robertson 2001].

According to the HEML scheme an event is composed of the following:

1. A label to name the event
2. One or more keywords that group conceptually similar events
3. A location in which the event takes place
4. A 'chronology' of the event, describing the time in, or during which, the event takes place
5. A list of people or groups of people who participate in the event
6. A list of evidence for the event, either in physical form (such as printed books or 16mm film) or as a web resource

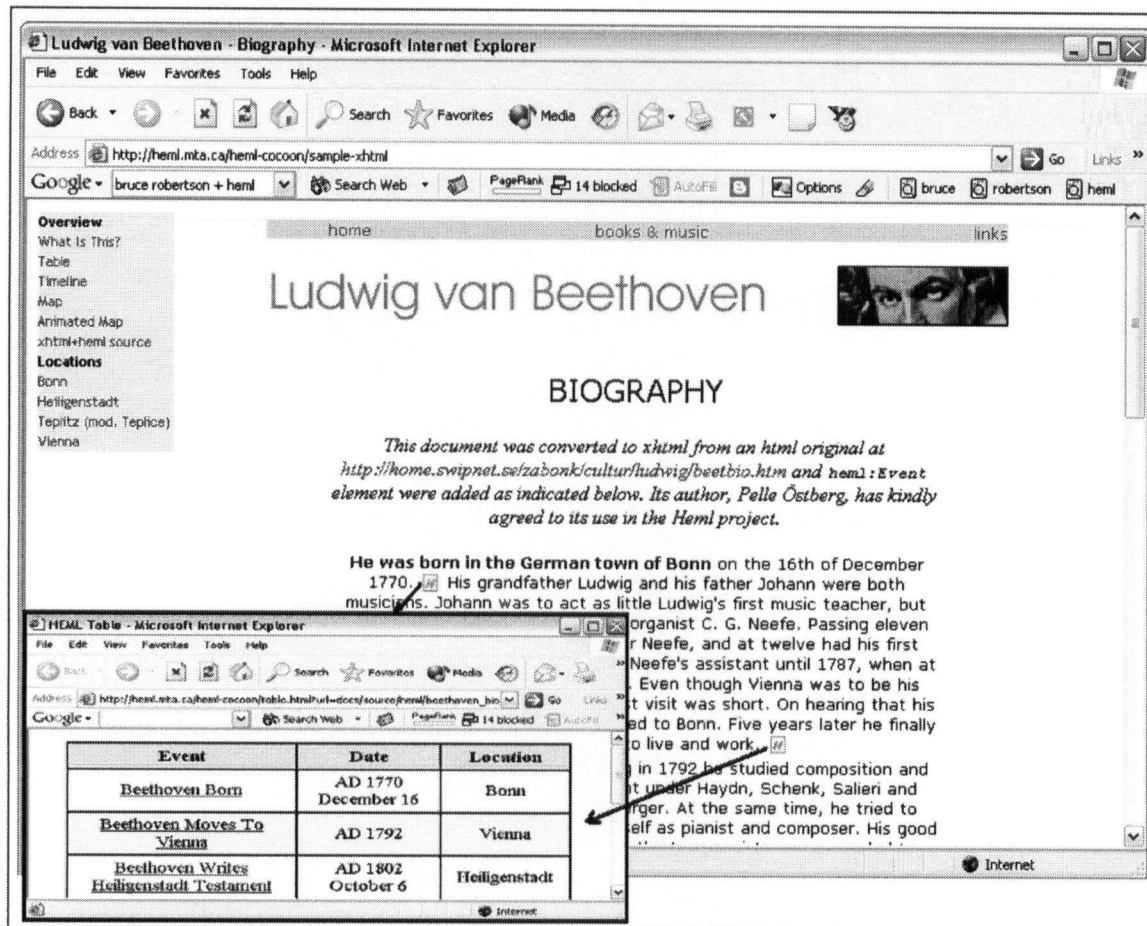


Figure 2-1 A sample document using HEML representation language

This figure illustrates a document in which HEML markup language has been used. By choosing the HEML links within the document a new pop-up window which demonstrates the date and location that is associated with the event is shown (adapted from <http://heml.mta.ca/heml-cocoon/sample-xhtml>)

Although this representation does allow a certain amount of the knowledge contained in a history document to be captured and represented, it lacks and understanding of the semantic within the document or the rich relations between the concepts.

2.1.3 Analysis of these approaches and our selected approach

Our goal of developing a paradigm to represent semantic knowledge within historical documents is unattainable with the approaches proposed in these and other similar projects.

The current preferred approach for creating information infrastructures such as those presented in the last two projects is to use markup languages. Amongst these markup

languages, XML is the most popular choice. However, XML can only represent the syntax within a document, not the semantic. This means that the tags in an XML document are semantic free. That is, these tags do not have any predefined meaning; they do not represent the semantic, meaning, or behavior of concepts in the document and only present content and structure [Jarrar 2000].

These projects use XML as an exchange format. However, an XML document by itself is not completely useful for our purposes. Associating semantics to these tags requires additional mechanisms to describe the behavior and the meaning for the concepts within a document. This association cannot be made unless there is a consensus amongst people within a community interested in a specific domain on what these semantics are [Spyns, Meersman et al. 2002]. In addition to this, In order to capture the dynamicity of the information in a historical document, we require additional functionality that to which is attainable by using markup languages such as XML.

An ontology is the basic building block used to define richer relationships between different concepts. It allows members of a community of interest to establish a joint terminology, which enables greater flexibility and helps achieve reusability and sharing of knowledge.

Ontologies are often used to represent a specification of domain information by providing a consensual agreement on the semantic of the knowledge that the domain is aimed to express [Spyns, Meersman et al. 2002]. Another motivation behind using ontologies is that they allow for sharing and reuse of knowledge in a computational representation. Based on these characteristics inherent to ontologies, we choose an ontological approach to represent the history domain we intend to capture.

From the general point of view, the issues discussed in this thesis can be situated within the Knowledge Representation (KR) area of Artificial Intelligence (AI), and more precisely, within what is called ontology-based knowledge representation. Therefore, the rest of this chapter gives an overview of the research done in the AI community in the fields of knowledge representation with a specific emphasis on ontology design methodologies and techniques.

2.2 Knowledge Representation and Ontologies

Over the past three decades, researchers have spent an enormous amount of effort to collect and represent knowledge about the real world in forms, which can be easily manipulated computationally. The Artificial Intelligence (AI) Community is actively involved in studying the issues related to this field.

Knowledge Representation is a branch of AI. This field refers to the study of how knowledge can be appropriately represented in computational models and what kinds of reasoning can be done with that knowledge. John F. Sowa [Sowa 2000a] defines Knowledge Representation (KR) as analyzing knowledge in some domain and transforming its informal specification to a computable model. Generally speaking, in order to capture the formal specification of knowledge, one must follow these steps [Sowa 2000a]:

- Identify the kind of things that require representation
- Provide an informal specification of these things
- Map the informal specification to a computable form

The current move within the AI community toward capturing the semantic within the knowledge and creating knowledge models that can be shared and reused, has driven knowledge representation into a new era. The main approach for capturing and representing knowledge in this new framework is characterized by the use of ontologies.

The term “ontology,” which is now become extremely popular in the field of Artificial Intelligence, has its origin in philosophy. The historical meaning of the word ontology in philosophy is “the metaphysical study of the nature of being and existence” [WordNet]. This refers to a branch of philosophy called Epistemology, which deals with the nature and organization of the real world.

The common understanding of this term in the AI communities is to the identification of the representable things that exist in a specific domain and the relationships amongst all of these existing things [Sowa 2000]. In other words, an ontology is a conceptual model, a resource containing knowledge about “concepts” that exist in the world and how they

relate to each other. A concept is a symbol for representing meaning with well defined attributes and relations with other concepts [Mahesh 1996].

In the following sections we discuss and describe the terms and concepts related to ontologies as well as provide the reason for using them as representational models for our history domain.

2.2.1 Why ontologies?

There are several reasons to develop and use ontologies as a part of Knowledge Representation systems. The following is a list of reasons taken from ontology literature [Guarino 1997; Noy and McGuinness 2001].

- To clarify and share the structure of knowledge

Many applications share common vocabulary and concepts. One of the main motivations for building ontologies is to enable sharing a common understanding of the structure of knowledge amongst people and applications. Different users or applications can use the same ontology to represent common terms and concepts in a particular domain of knowledge. These shared ontologies can later be used to facilitate knowledge extraction and manipulation from different systems, and reduce the high cost of knowledge acquisition.

- To allow reusing knowledge

The ability to integrate and reuse an existing ontology, without needing to rebuild it, provides a great benefit.

One of the main motivations for ontology research comes from its capability to facilitate reuse of domain knowledge. There are two types of ontology reuse methods: merge and integration [Pinto, Perez et al. 1999].

- “Merge” refers to building an ontology in a domain reusing two or more different ontologies from the same domain. The ensuing ontology is a combination of the involved ontologies in a way that means one cannot identify the sources by having the resultant ontology (Figure 2-2). The purpose of merge is to create a

more general ontology by combining several other ontologies within the same domain.

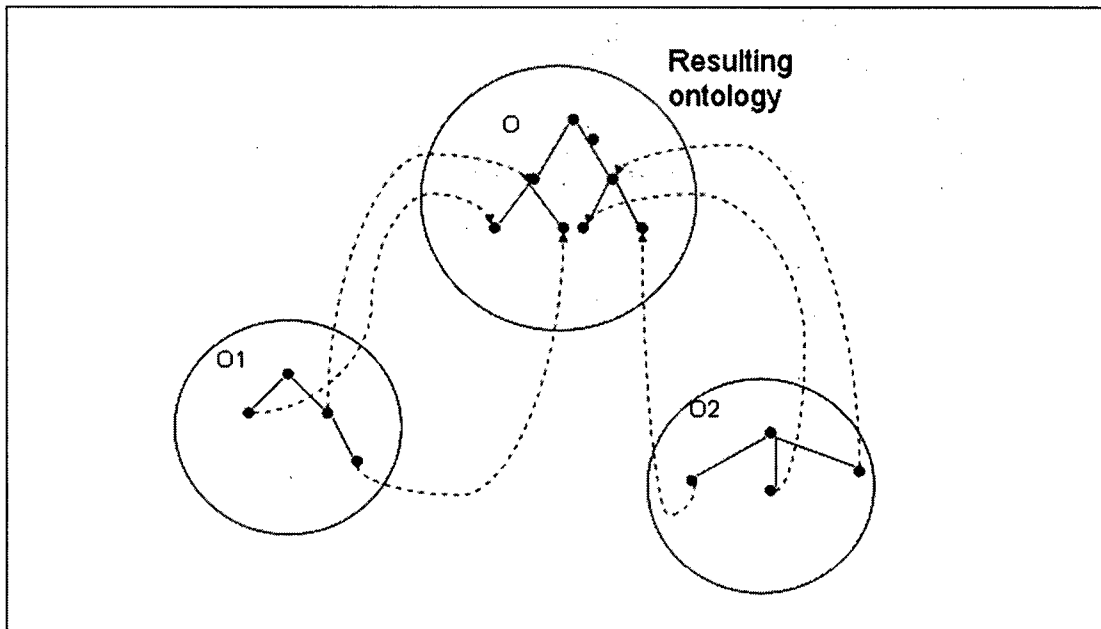


Figure 2-2 Merging different ontologies

This figure illustrates what the resulting ontology of a merge process would look like. It can be seen that the resulting ontology (O) is a combination of the two involved ontologies (O1, O2). However from the resulting ontology one can not identify which parts belongs to which original source.

- Integration refers to building an ontology in a particular domain reusing ontologies from different domains. In an integration process, the involved ontologies remain largely unchanged, or at most undergo minor changes. It is possible to identify the sources by having the resulting ontology (Figure 2-3). An example of this can be seen in the reuse of an existing time ontology to create one that includes a time notion. Many models in different domains use the notion of time. A generic ontology for time can be included in developers' own models. In other words, if an ontology has already been developed, it can be employed by other users in their models to capture certain aspects in their domains. Several existing ontologies can also be combined to build a larger ontology.

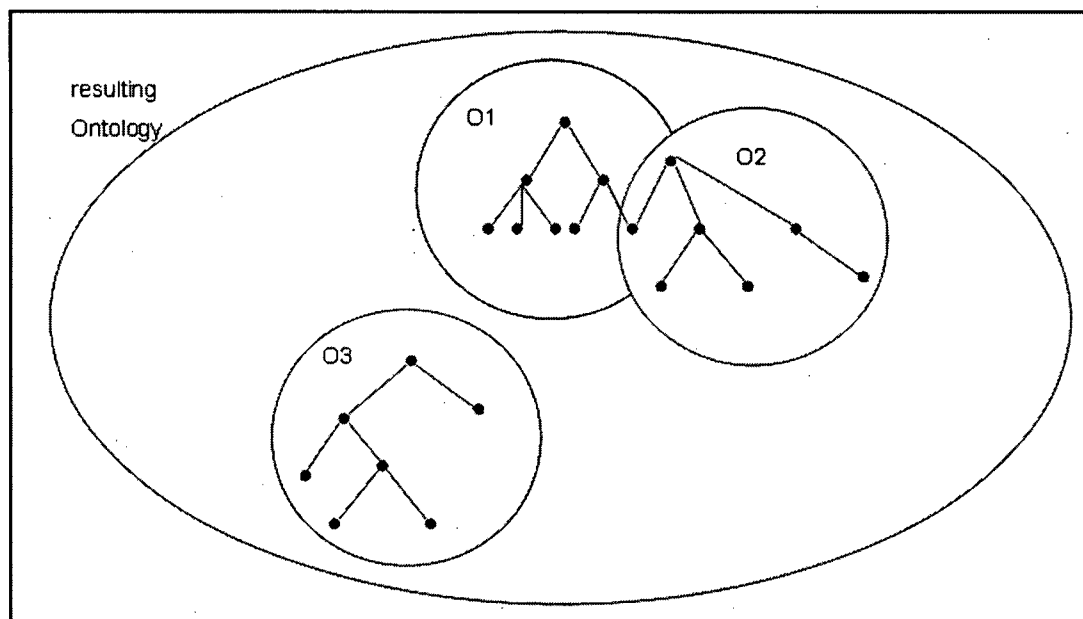


Figure 2-3 Integrating several ontologies into one more general ontology

This figure shows how the source ontologies (O_1 , O_2 , O_3) in an integration process can be easily recognized in the resulting ontology

Although reuse is referred to as one the biggest advantages of using ontologies in the literature, it is not clear how a merge or integration must be done. Ontology interoperability has been recognized as a challenging yet unachieved task. No current ontology building methodology really addresses this issue or deals with it explicitly. There is no consensus for the methods used in merging and integration. These are still unclear and more of an art than a methodology. These issues are still part of ongoing research in the area [Pinto, Perez et al. 1999; Beck and Pinto 2003].

Guarino [Guarino 1997] states that reusability can be acquired if the domain knowledge has higher generality and granularity than the problem it intends to solve. Given that the knowledge is dependent on the problem it intends to solve, the main concern is how to relate the problem to the domain concepts. Different applications can make use of the same basic knowledge related to a domain, however, they need to adapt this knowledge base to solve their particular problems. We believe that, one cannot expect to find off-the-shelf ontologies and be able to use them to solve any given problem. However, we can

expect to find ontologies that can be easily modified to fit the needs of the desired application.

- To make the assumptions used to create the domain explicit

Making the domain assumption explicit helps with understanding, locating, and later altering these assumptions in case the knowledge about the domain changes. The description and meaning of the knowledge that has been captured in the ontology is explained and clarified in these assumptions.

- To allow discerning between domain knowledge and the operational knowledge

As described in the last section, ontologies can be classified into either domain or task ontologies. If the specification of the domain concepts is available, different types of tasks can be performed on the knowledge. On the other hand, the same task ontology can be applied to different domain ontologies using the same structure.

- To study domain knowledge

Having a formal specification of the domain knowledge helps with the study, reuse, and extension of an existing ontology.

2.2.2 Defining the term “ontology”

As mentioned before, the term ontology has its origins in philosophy. While within the philosophy community there is an agreement upon the definition of ontology as “a particular theory in metaphysics concerned with the nature and relations of being and the kind of existents”, [Webster] there is some dispute amongst those in the AI community upon this term’s meaning. Many definitions of ontology have been offered in AI literature; some of these contradict one another.

This section gives an overview of the definitions of the term ontology within the AI community and follows with a definition that best suits the purpose of this research.

To distinguish between the meanings of the term “ontology” in philosophy and AI, Guarino [Guarino and Giaretta 1995] delineates the following two terminologies:

- “Ontology” (with upper-case “o”), which refers to the philosophical meaning of the term, is “a branch of metaphysics which deals with the nature and organization of reality.”
- “ontology” (with lower-case “o”) denotes the AI meaning of the term.

People in different contexts and with different interests have proposed diverse definitions for the term. At a glance, an ontology represents some kind of world view with a set of concepts and relations amongst them, all of these defined with respect to the domain of interest. Some scholars redefine the term in an effort to capture an absolute view of the world. For instance, John F. Sowa [Sowa 2000a] defines an ontology as

“The study of existence, of all kind of things (abstract and concrete) that make up the world”

Uschold et al [Uschold and Gruninger 1996] describe an ontology as

“A vocabulary of terms and some specification of their meaning”

Given the vague nature of these definitions, it is unlikely that they will prove to be useful for those of us contemplating the creation of real usable ontologies. Having said this, there exists a set of more explicit definitions, which better capture the essence we are looking for.

The most cited definition of the term is proposed by Gruber. In [Gruber 1995] Gruber defines the term as

“An explicit formal specification of a conceptualization”.

To clarify the definition, the following terminology has been proposed [Gruber 1995; Studer, Benjamins et al. 1998].

- “*Conceptualization*” is an abstract, simplified model of concepts in the world, usually limited to a particular domain of interest.
- “*Explicit*” indicates that the type of domain concepts and the constraints imposed on their use are explicitly defined.

- “*Formal*” means that the ontology specification must be machine readable

Taking into account that the main reason to create ontologies is knowledge sharing, Borst [Borst 1997] modifies Gruber’s definition to

“An ontology is a formal specification of a shared conceptualization”

Studer et al. [Studer, Benjamins et al. 1998] combine the two definitions above,

“An ontology is an explicit formal specification of a shared conceptualization”

However, in [Guarino and Giaretta 1995] Guarino argues that an ontology is only a “partial account of conceptualization,” not an explicit specification of conceptualization as Gruber claims. He identifies a conceptualization as a description for the intended meaning of the terms employed to represent the relevant relations. It has been argued that these meanings remain the same even if they have been used in different contexts or arrangements. In other words, a conceptualization is a set of informal constraints imposed on the structure of the knowledge.

As we can infer from what is stated above, the definition of ontology is closely connected to the interpretation of “conceptualization.” Different interpretations of the term “conceptualization” produce confusion when interpreting the term “ontology.”

Researchers in different communities interpret the terms in a manner that better serves their purposes. This in turn gives rise to confusion and discrepancies that further complicate the clarification of these terms.

In some definitions, conceptualization and ontology are kept clearly distinct. However, some others see the terms as indistinct. In either case, we can say that every ontology model for knowledge representation is either explicitly or implicitly committed to some conceptualization.

The definition proposed by Guarino best serves the purpose of this thesis. We agree with Guarino that an ontology is, possibly, an incomplete agreement about a conceptualization and not a specification of the conceptualization. This means that an ontology is an agreement between people in a community sharing interest in a common domain. We reify the definition as

"An ontology is an incomplete formal description of concepts and their relations in a domain of interest"

By "incomplete" we mean that the definition of concepts should be left open for further interpretation and manipulation by the users of the ontology according to their specific area and domain of interest and their intended application. "Formal" means that the ontology specification can be easily translated into a machine readable code, however, this is not mandatory when defining ontologies at an abstract level.

We also believe that this definition must be left open to further interpretation regarding the specific area of interest and research and the application in which the ontology is used.

2.2.3 What do ontologies look like?

An ontology is usually represented by concepts within a particular domain in a hierarchical form. This hierarchy can also be referred to as a taxonomy of the domain of discourse. Likewise, the concepts can be referred to as vocabulary or terms. The taxonomy is the central part of most ontologies.

Almost any ontology includes something besides the taxonomy of concepts. An ontology consists of concepts, relations amongst these concepts, and axioms which define constraints on these relations. These axioms are used to make the interpretation of the content (concepts and relations) within an ontology explicit (same across different systems). The relations and axioms associated with the concepts in an ontology form the "ontology content" which tries to capture the semantics of the domain. In contrast, a taxonomy represents the syntax of a domain.

In short, for a taxonomy to be recognized as an ontology, it must have at least the following properties: [McGuinness 2002]

- Finite yet extensible set of vocabulary
- Unambiguous interpretation of concepts and their relationships
- Strict hierarchical sub-concept relations between concepts

The taxonomy of an ontology can be a simple concept hierarchy or a complex hierarchy with dimensions at each level. An example of a simple taxonomy is a tree-like hierarchy representing concepts. The branches of this tree are “is-a” relations, and different branches growing from a concept to a sub-concept are disjoint. In a complex taxonomy, the general layout of the concepts is different. In this case, there are several top-level categories at the same level. These layers are subcategorized along parallel dimensions and described by combinations of values along these dimensions.

As an example, a partial view of a simple newspaper ontology can be seen in Figure 2-4 through Figure 2-6. Figure 2-4 illustrates a partial view of the “is-a” (concept/sub-concept) relation within the newspaper ontology. From this figure we can say that a **Salesperson** is an **Employee**, and that an **Employee** is a **Person**.

Figure 2-5 demonstrates the relations (other than the “is-a”) that hold amongst the concepts. In this example, we can see that an **Article** *has* an **Author**. This **Author** can be a **Columnist**, a **Reporter**, an **Editor**, or a **News-Service person**. Subsequently, an **Editor** *is responsible for* **Employees** and so on.

Figure 2-6 shows a constraint imposed on employee salary. This constraint states that no **Employee** can get a salary greater than the **Editor** who *is responsible for* him/her.

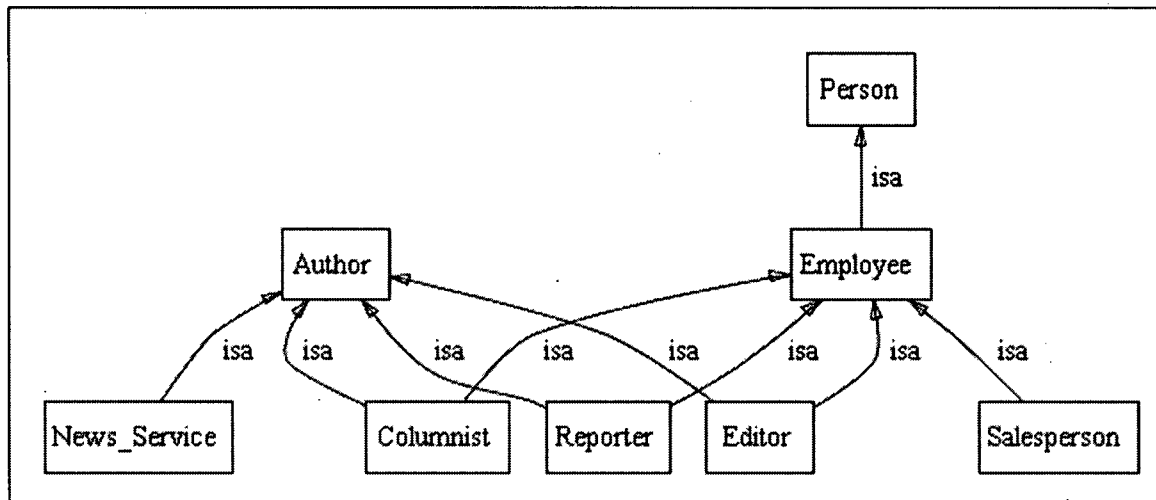


Figure 2-4 Partial view of a simple newspaper ontology (concept/sub-concept relations)

This figure illustrates the is-a (being type of) relation within the concepts in the ontology. We can see that a columnist is an author and an employee at the same time. Additionally, an employee is a type the more general concept person

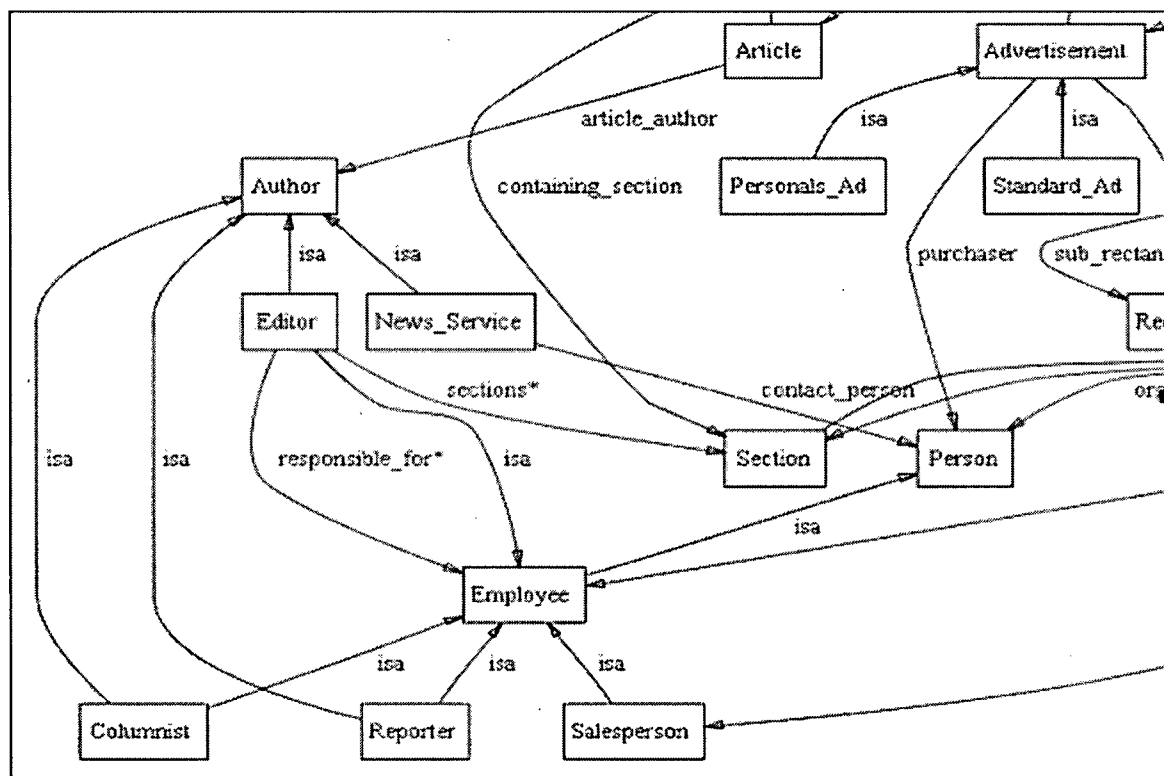


Figure 2-5 Partial view of the newspaper ontology (relations amongst concepts)

This figure illustrates the relations that hold amongst the concepts within the newspaper ontology. In this figure an editor, who belongs to a more general notion author, is at the same time responsible for employees who can be either columnists, reporters or salespeople.

Name editor-employees-salary-constraint	Description The salary of an editor should be greater than the salary of any employee which the editor is responsible for.
Statement <div> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> </div> <pre> (forall ?editor (forall ?employee (=> (and (responsible_for ?editor ?employee) (own-slot-not-null salary ?editor) (own-slot-not-null salary ?employee)) (> (salary ?editor) (salary ?employee)))))) </pre>	Range <pre> (defrange ?editor :FRAME Editor) (defrange ?employee :FRAME Employee responsible_for) </pre>

Figure 2-6 Constraint imposed on concept property within the newspaper ontology (axiom)

This figure shows a constraint imposed on the employee's salary. This constraint states that no employee can get a salary higher than the editor who is responsible for him/her.

2.2.4 Ontology classification

There are different approaches for categorizing ontologies. The most common one classifies each ontology into one of the following four categories according to the specification and generality of the domain that it aims to represent [Guarino 1997; Noy and Hafner 1997; Beck and Pinto 2003].

- General or Top-Level ontologies

Ontologies within this category are very general and aim to represent the various types of things that exist in the world without considering any particular domain or problem. The knowledge represented in this type of ontology includes terms related to general (top-level) concepts in the world such as things, events, processes, and spatio-temporal components. Because of their generality, these ontologies can be used across different domains. Some examples are represented in [CYC] and [Sowa 2000a]. The following

figures (Figure 2-7 and Figure 2-8) show upper level hierarchies in CYC and Sowa's top-level ontology.

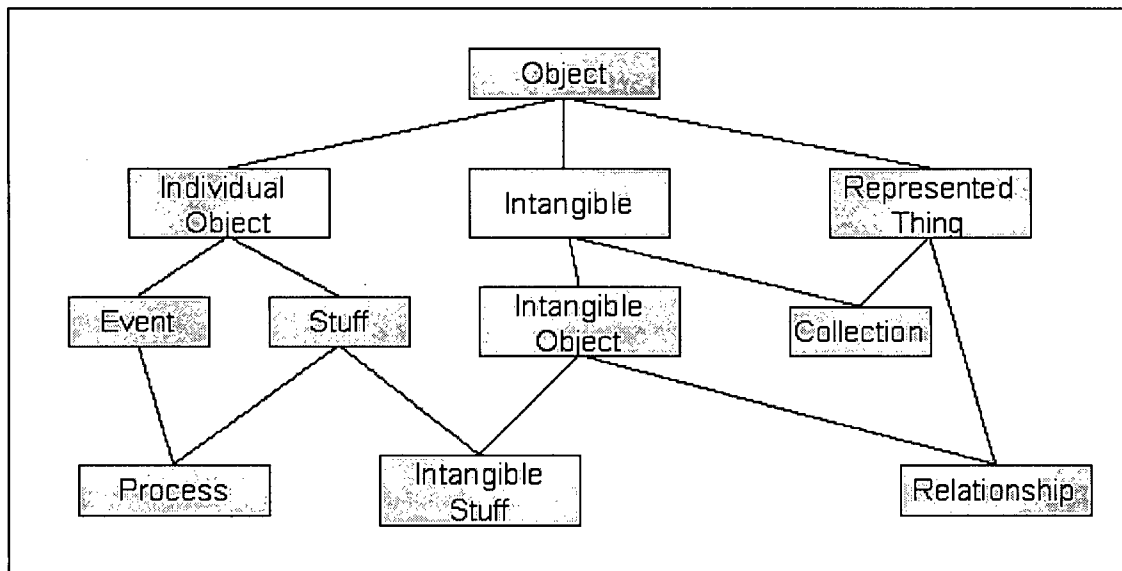


Figure 2-7 CYC ontology
Part of CYC's Top-Level categories

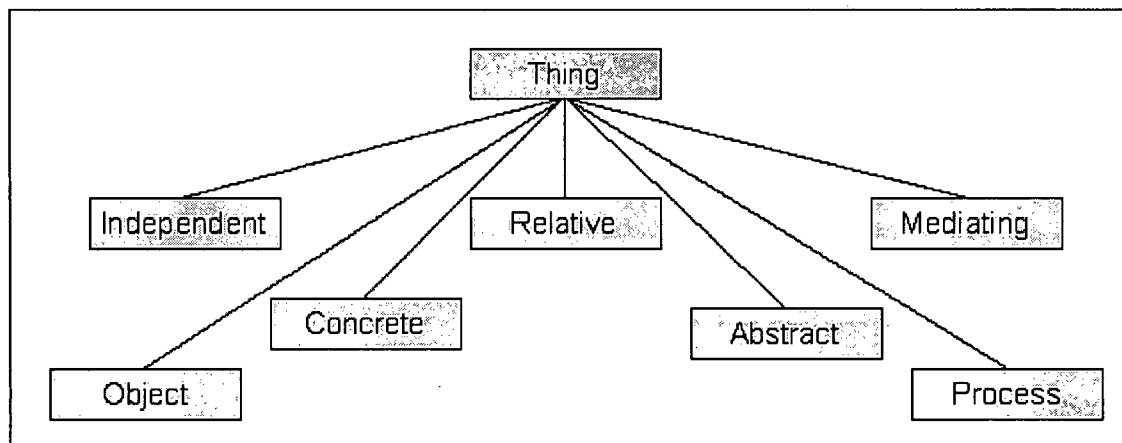


Figure 2-8 Sowa's ontology

Chart explaining the hierarchy of top-level categories in Sowa's ontology. This approach considers three distinct top-level categories: Concrete vs. abstract; Independent vs. Relative vs. Mediation; and Object vs. Process.

John. F. Sowa categorizes concepts in the real world based on philosophical foundations. His approach has three distinct top-level categories as can be seen in Figure 2-8.

- Concrete versus Abstract
- Form (Firstness) versus Role(Secondness) versus Mediation(Thirdness)

These categories are not mutually exclusive. For instance, a Man is a Form since it can be described without considering anything exterior to a Person. In relation with other concepts the Man can take the role of a father, son, or teacher. Parenting is a Mediation since it relates several concepts to one another.

- Object versus Process

Objects maintain their identity over a period of time, whereas Processes change their state within a certain timeframe.

- Domain-specific ontologies

These ontologies represent knowledge specific to a particular domain of discourse.

The Unified Medical Language System (UMLS) is an instance of a domain ontology (Figure 2-9). This system has been developed to assist retrieval and integration of biomedical information from several different sources. The main purpose of this project is to facilitate the use of different sources by people whose terminologies might be dissimilar. UMLS has both taxonomic and non-taxonomic hierarchies. The former are used to represent medical concepts and the latter represent the relations amongst these concepts [UMLS].

Another example of this type of ontology is the Toronto Virtual Enterprise (TOVE) project presented in [Fox 1992; Fox and Gruninger 1994; Gruninger and Fox 1995] (Figure 2-10). The project's goal is to create a generic, reusable enterprise model which is a computational representation of the structure, activities, processes, information, resources, people, behavior, goals, and constraints of a business, government, or other enterprise.

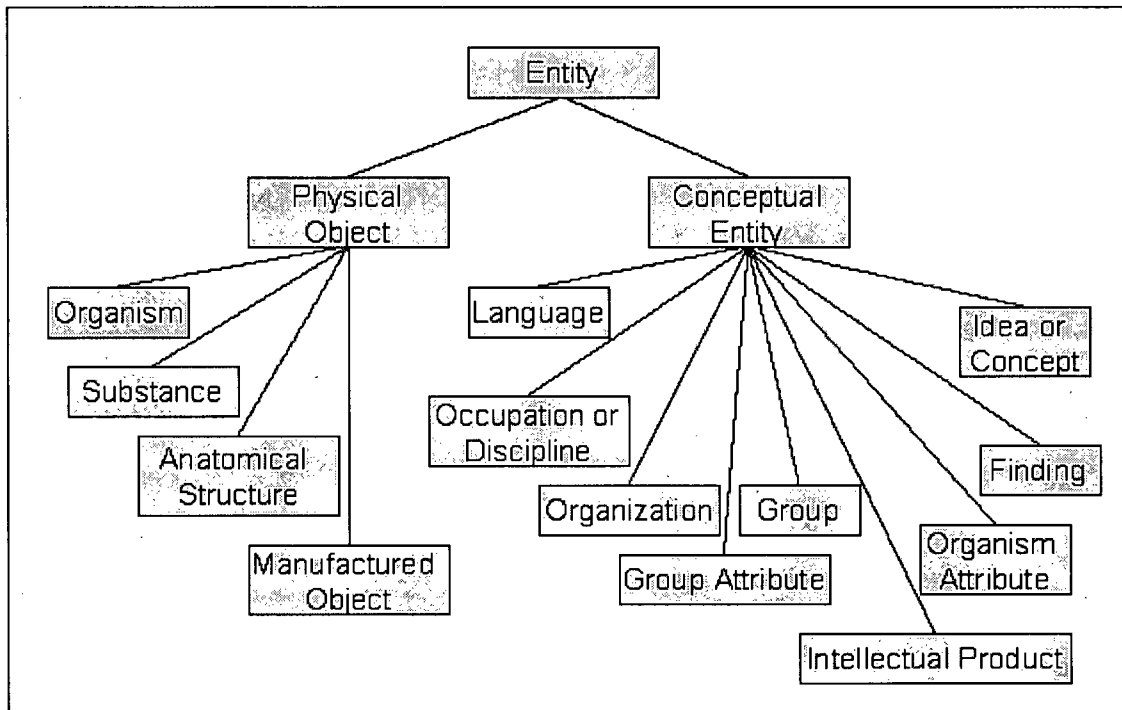


Figure 2-9 UMLS

Top-Level hierarchy of entities (adapted from [Noy and Hafner 1997])

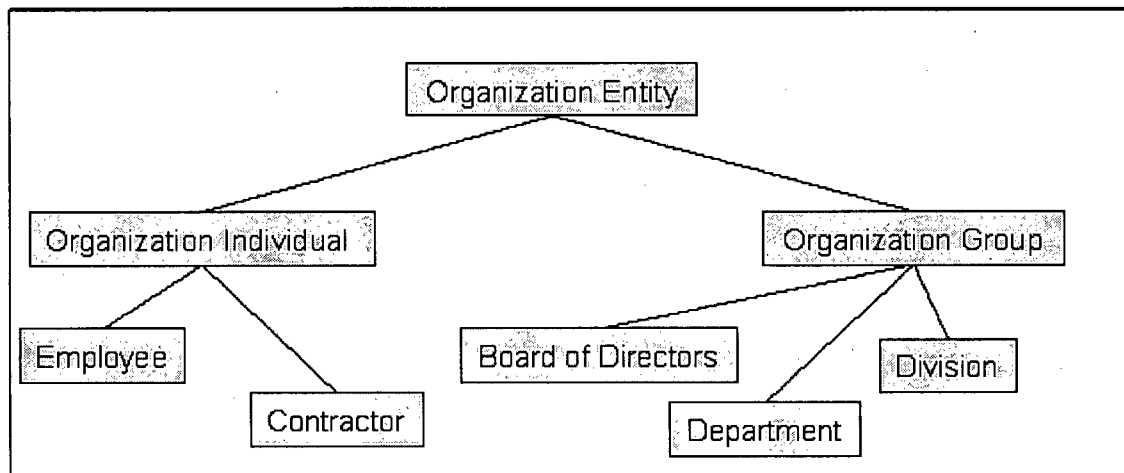


Figure 2-10 TOVE

An Organization-Entity Hierarchy in the Organization ontology (adapted from [Noy and Hafner 1997])

Since both top-level and domain ontologies try to represent knowledge about the real world, they share several key similarities which makes distinguishing between them

difficult. However, one should keep in mind that even when these two types of ontologies represent the same domain, their particular view of this domain can be different.

- Application ontologies

These kinds of ontologies contain both the domain and their related tasks. They not only express the concepts in a domain but also relate them to the tasks that can be performed on them. These ontologies are related to problem solving methods. They are used to describe the concepts employed to solve the problem associated with a particular task.

As an example of this type of ontology, we present part of what is described in [vanHeijst, Schreiber et al. 1997]. This example uses CASNET [Weiss, Kulikowski et al. 1978] which allows the expression of causal links describing the processes associated with diseases and the development of diagnosis applications (Figure 2-11). Domain knowledge is represented in unlabeled boxes. Likewise, labeled boxes represent knowledge associated with problem solving methods.

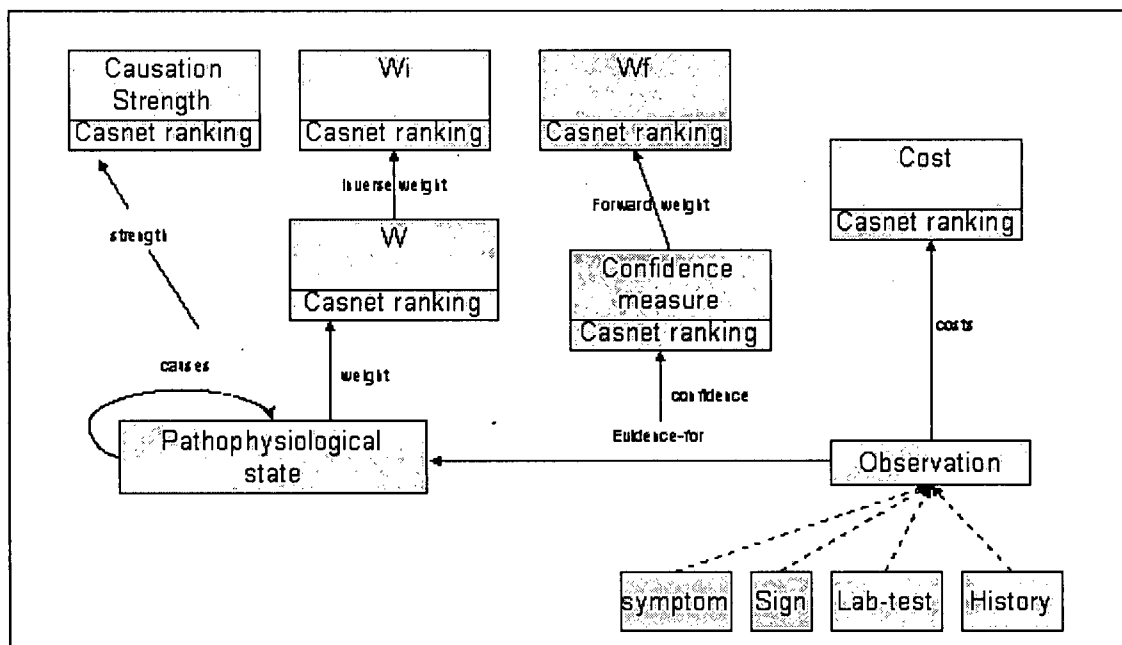


Figure 2-11 Example of application ontology

Taken from [Beck and Pinto 2003]

This figure presents a partial view of an application ontology in the medical diagnoses domain. This ontology not only allows expression of the domain knowledge (labeled boxes) but also the knowledge related to problem solving methods (unlabeled boxes).

- Representation ontologies or meta-ontologies

These ontologies are specifically formalized in a manner that facilitates the sharing and reusing of knowledge. Developers and users alike employ them to share ontologies that utilize the same representation primitives to formalize knowledge. In meta-ontologies, the underlying conceptualization expresses the representation primitives.

One example of this kind of ontology is the frame ontology presented by [Gruber 1993]. The frame ontology defines the terms that capture the conventions used in object-centered knowledge representation systems (Figure 2-12). It is created using the Knowledge Interchange Format (KIF), an ontology building language.

<p><u>Class RELATION</u> <u>Subclass-Of:</u> <u>Set</u> <u>Axioms:</u> $(\Leftarrow (\text{Relation } ?\text{Relation}) (\text{And } (\text{Set } ?\text{Relation}) (\text{Forall } (?\text{Tuple}) (\Rightarrow (\text{Member } ?\text{Tuple } ?\text{Relation}) (\text{List } ?\text{Tuple}))))))$ <u>Defined in theory:</u> <u>Frame-ontology</u> <u>Source code:</u> <u>frame-ontology.lisp</u> <u>Also defined in:</u> <u>Kif-relations</u></p>
<p><u>Class CLASS</u> <u>Subclass-Of:</u> <u>Relation</u> <u>Slots Of instances:</u> <u>Arity:</u> 1 <u>Axioms:</u> $(\Leftarrow (\text{Class } ?\text{Class}) (\text{And } (\text{Relation } ?\text{Class}) (\text{= } (\text{Arity } ?\text{Class}) 1))))$ <u>Defined in theory:</u> <u>Frame-ontology</u> <u>Source code:</u> <u>frame-ontology.lisp</u></p>

Figure 2-12 Part of the *frame ontology* in Ontolingua

(<http://www.ksl.stanford.edu/software/ontolingua/>)

This figure represents two of the representation primitives that exist in the “*Frame Ontology*”: “**RELATION**” which describes the association between concepts within an ontology, and “**CLASS**” which represents concepts in the domain that the ontology is intended to capture.

When trying to better understand the above classification, one must take into consideration that people from different communities with diverse interests view the world from different angles. An important aspect in one context can be trivial or even insignificant in another. This leads us to believe that expecting to reach a universal consensus for a general ontology representing any part of the world with its concepts and relations in their entirety, is an unrealistic goal. Even if this goal is achieved, it is still unfeasible to try to solve any particular problem with the obtained ontologies.

Likewise, domain-specific ontologies are not only empowered by but also limited by the particular applications that are intended to use them [Beck and Pinto 2003]. Even for the simplest case of a domain-specific ontology, there can be many different interpretations of the levels of importance and detail in the specification of concepts and relations.

Notwithstanding, we believe that general ontologies should still be developed to serve as a resource and aid for developing both domain and application ontologies. The key issue here is how to adapt a general ontology to build a domain-specific or application ontology.

More importantly, integrating two or more ontologies, whether they are general or domain specific, remains an important issue. For general ontologies, the main concern is in the design, which should facilitate appending domain-specific ontologies to them. Similarly, for a domain specific ontology, the issue is whether it can easily reuse previously defined knowledge from a general ontology. Yet another issue is whether a domain specific ontology can be easily integrated into a more general one.

2.2.5 Methods for building an ontology

Building a well-developed, usable, and sharable ontology represents a significant challenge. There is great diversity in the way ontologies are designed as well as in the way they try to represent the world. Before ontology sharing and reuse become a practical reality, some standard should emerge to define what an ontology should consist of and how it should be represented. In short, we need to know how to build a usable ontology.

A range of methods and techniques have been reported in the literature regarding ontology building methodologies. However, there is an ongoing argument amongst those in the ontology community about the best method to build them [Noy and Hafner 1997; Lopez and Perez 2002; Beck and Pinto 2003].

Most of the ontology building methodologies are inspired by the work done in the field of knowledge engineering to create methodologies for developing knowledge based systems (KBS). For example, the Enterprise Methodology (described in section 2.2.5.1), like most KBS development methodologies, distinguishes between the informal and formal

phases of ontology development. METHONTOLOGY (described in section 2.2.5.4) adapts the work done in the area of knowledge based evaluation for the ontology evaluation phase. A Knowledge Base (KB) is a knowledge model of a KBS. This KB contains both abstract and specific knowledge regarding a particular area defined in a computational format. There are similarities between knowledge bases and ontologies. Both KB and ontologies define, gather and represent knowledge about concepts and their relations in a computational format for a particular domain of interest. The additional feature provided by ontologies is the ability to reuse and share this knowledge. Ontologies prevent KBS developers from having to build KB's from scratch. Instead, they do so by using reusable components. However, knowledge bases and ontologies differ in the following characteristics [Perez 1994a; Perez 1994b].

First, the knowledge captured in ontologies is more general than that captured by KB's. Therefore, knowledge in ontologies is more appropriate for reuse and sharing across applications [Beck and Pinto 2003].

Second, ontologies differ from KBs in that they do not usually contain reasoning methods within themselves. Ontologies do not include methods relating them with the use of any kind of knowledge.

Third, the language used to formalize KB's has an influence on the quality and quantity of the knowledge gathered in said KB's. However, the selected language to be used within an ontology has no effect on its knowledge [Perez 1994a; Perez 1994b].

Even though these two approaches, Knowledge Bases and Ontologies, differ in some aspects, they are still closely related and have a strong influence on each other.

In the reminder of this section, we describe the most representative ontology building methods that address the problem of building ontologies from scratch. These are presented in chronological order. We analyze their approach, strong points, and shortcomings according to the following criteria:

- Level of Detail and Clarity: how well is the methodology described?

The approach is analyzed to determine if it provides a precise and comprehensive description of the techniques and methods suggested.

- Techniques for Identifying Ontology Terminology

Look at the techniques and strategies proposed for capturing the terms related to the ontology and how efficient these methods and techniques are.

- Generality

Determine whether the proposed approach is application dependent or not. There are two general points of view when it comes to building an ontology. The first considers building an ontology in general, the second concentrates on case studies in the development of a single ontology related to a particular problem or domain of discourse. None of the methodologies we analyze in this study are strictly case studies. However, each of them has a different degree of inherent generality.

- Ontology Evaluation

Analyze whether any techniques are provided to evaluate the completeness of the resulting ontology according to the specified requirements (if these were determined).

- Usability

Investigate whether the technique applies to building a real ontology. If this is the case, we study whether this ontology is successful according to the intended user community. We also determine if the user community is closely involved in the development process.

All of these methods and techniques are still determined to some extent by the particular circumstances in which they are applied. We must note that, in any given circumstance there might be no available guideline for deciding on what techniques and methods to apply [Uschold 1996].

2.2.5.1 Uschold and King Methodology (Enterprise Methodology)

The first methodology discussed here is originally proposed by Uschold and King [Uschold and King 1995] in 1995. They present a guideline derived from their own experience in building a particular ontology in an enterprise modeling process [Uschold, King et al. 1998]. Their goal is to introduce a comprehensive methodology (containing a set of techniques, methods, and guidelines) for building ontologies.

In this work, Uschold and King outline the stages they believe are required to build an ontology. Their approach is described in detail in this section.

Their idea of a method for developing an ontology consists of the following four stages:

- Identify the purpose and intended users
- Building the ontology
 - Ontology capture
 - Ontology coding
 - Integrating existing ontologies
- Evaluation
- Documentation

Figure 2-13 illustrates their approach.

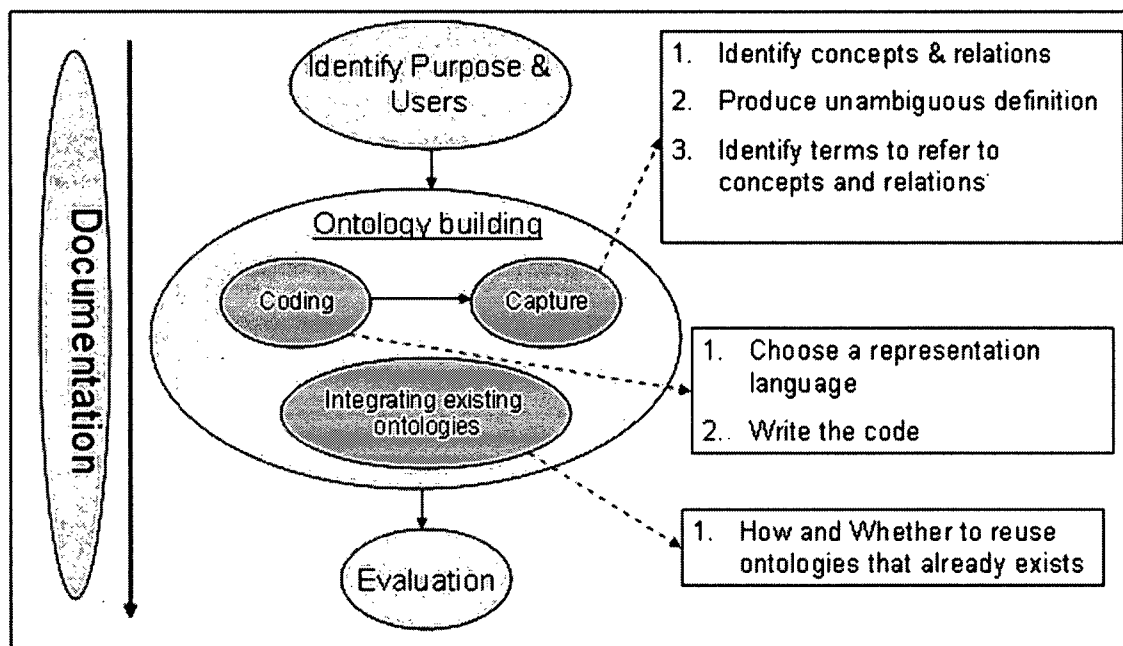


Figure 2-13 Uschold and King Methodology (Enterprise Methodology)

This methodology is composed of four different stages: 1. Identifying purpose and users; 2. ontology building; 3. evaluation; and 4. documentation. As it can be noticed, the documentation takes place all along the development process.

The following is a detailed description of each stage in the process:

- Identify the purpose and intended users

It is important to know why an ontology is being developed and what purposes it must serve when it is complete. In other words, it is essential that ontology developers clarify their own purpose for building the ontology. In order to efficiently develop an ontology, the ontology creators must identify both the intended purpose and end users.

According to Uschold, we can identify four major purposes for developing an ontology [Uschold 1996]:

- Facilitating communication between people
- Allowing interoperability amongst systems
- Providing reusability
- Knowledge acquisition
- Building the ontology

Uschold and King identify the following three main sub-steps in this section:

- Ontology capture

An important part of the effort involved in the development of ontologies is directed at identifying those categories and concepts in the real world that are of interest, defining them, and defining the appropriate terms to refer to them. This procedure is referred to as *ontology capture* in this methodology. The authors refer to *ontology capture* as an informal definition of the ontology. This stage consists of the following:

- a. Identification of the key concepts and relations amongst them in the domain of interest. (The main focus is on the concepts not the words used to represent them.)
- b. Clearly define such concepts and relations in a natural language text.
- c. Employ reference terms for these concepts and relations.
- d. Reach an agreement on all of the above.

To determine the relevant concepts, a brain-storming session is used. However, according to them, there is a need for consulting with domain experts to avoid ambiguities and differences in opinion. This is followed by categorization of terms that are closely related into groups. The result of this process is a set of categories of terms.

An important issue in identifying and categorizing the concepts is deciding on the level of generality and granularity of the selected categories. In order to categorize the concepts, they adopt the middle-out approach proposed by Lakoff [Lakoff 1987]. The idea behind this approach is that the categories are not simply organized in a hierarchy from the most general to the most specific, but rather organized cognitively in a way that categories are situated in the middle of the general-to-specific hierarchy. Going up from this level is the generalization and going down is the specialization.

In general, there are three approaches available for categorizing concepts in a domain of discourse: top-down, bottom-up, and middle-out. In the top-down approach, categorization starts from the most general concepts. The bottom-up approach first identifies the most specific concepts and then groups them into categories. In middle-out, categorization starts from neither the most general nor the most specific but from the most important concepts and evolves in both directions from there.

The disadvantage of using the top-down approach is that since we start with a few general concepts we might suffer from ambiguity. On the other hand, in the bottom-up approach we may provide too much detail which might not be used in a final version of the ontology [Uschold and Gruninger 1996].

As stated before, Uschold and King employ the middle-out approach to construct the hierarchy of concepts in their case study. First, the most important concepts are identified and from there the rest of the hierarchy is captured through generalization (the top-down method) and specification (bottom-up). The next step in this phase is to identify the cross-references within and amongst the concepts in the groups.

After constructing the categories, the next step in *ontology capture* is to provide an unambiguous definition for all the terms and decide if they are important enough to be included in the ontology. Uschold and King provide a general guideline on how to produce definitions and how to handle the ambiguity in definitions as follows: a. Determine a precise and clear natural language text definition of all the terms. b. Dictionaries and other technical glossaries and most importantly, the domain experts are used to ensure the consistency of the newly added term with the terms already in use. c. Specify the relationship of the term being defined with other commonly used

terms that are similar to it. d. Provide additional information (and possibly examples) to help understand the meaning and usage of the term. The final step is to perform a critical review of the definitions.

- Ontology coding

According to the authors, coding is the explicit formal representation of the conceptualization acquired in the previous step (*ontology capture*). This includes choosing a representation language, and producing the formal ontology.

- Integrating existing ontologies

The authors do not present a clear definition of the subject but rather only note that the integration is a difficult problem to overcome which needs additional attention from the ontology community. The main issue is how to achieve an agreement among different users to be able to share and reuse an already defined ontology.

- Evaluation

The issue of evaluation is not clearly addressed by the authors. The approach taken is to adopt what has been done in the field of knowledge-based systems for evaluation and adapt it to ontologies. Competency questions [Gruninger and Fox 1994], the queries that the final ontology should be able to answer, are used to evaluate the ontology. However, unlike other approaches, they do not use competency questions to identify the concepts at the early stage of developing the ontology. They claim that competency questions are too specific to be used as a guide at this early stage.

- Documentation

Uschold and King suggest that important assumptions should be carefully documented. However, they do not provide a precise guideline on how this is to be done.

Analysis

We refer to the previously proposed analysis criteria to present our opinion of Uschold and King's methodology:

- Level of Detail and Clarity: how well is the methodology described?

The authors outline the stages that should be taken when building an ontology but fail to provide enough detail for some of these steps. In particular the "*Ontology Coding*," "*Integration*" and "*Evaluation*" sections are only presented in a superfluous manner.

- Approach to Identifying Ontology Terminology

The main focus of this methodology is the *ontology capture* stage which includes identifying the ontology terminology. They focus particularly on categorizing and handling ambiguous terms. This work addresses this particular analysis criterion quite well, given its focus. The authors adapt common Knowledge Acquisitions techniques available at the time for defining this stage and also utilize the middle-out approach to categorize the ontological concepts.

- Generality

The proposed method is application-independent and very general. The procedure outlined in this work is completely independent of the intended application for the ontology. It can be used for developing a variety of different applications in different domains.

- Ontology Evaluation

The authors use competency questions to evaluate the completeness of the ontology but no detail is provided as to how this is done. The issue is addressed but not thoroughly.

- Usability

The authors apply their methodology to build an ontology called Enterprise Ontology [Enterprise]. The proposed ontology is a collection of terms and definitions relevant to business enterprises. This project is developed as a key element for supporting communications within the Enterprise Project and as a way to capture the knowledge related to enterprise domain. The Enterprise Project is a UK government promoted

project in enterprise modeling. The main goal of this project is to attain a wide view of an enterprise that can assist in decision making.

According to the authors, the ontology is very general and contains most of the common terms relating to an enterprise. However, the application intended to use this ontology must modify or extend it to respond to any particular situation. The authors claim that the main part of the modifications deals with the extension of existing concepts toward more specialized terms to meet the requirement of the application purpose.

The Enterprise Ontology consists of several conceptual components such as *Activities* and *Processes*, *Organization*, *Strategy*, and *Marketing* (Table 2-1) [Stader 1996; Uschold, King et al. 1998].

Activity	Activity Specification, Execute, Executed Activity Specification, T-Begin, T-End, Pre-Conditions, Effect, Doer, Sub-Activity, Authority, Activity Owner, Event, Plan, Sub-Plan, Planning, Process Specification, Capability, Skill, Resource, Resource Allocation, Resource Substitute.
Organization	Person, Machine, Corporation, Partnership, Partner, Legal Entity, Organizational Unit, Manage, Delegate, Management Link, Legal Ownership, Non-Legal Ownership, Ownership, Owner, Asset, Stakeholder, Employment Contract, Share, Share Holder.
Strategy	Purpose, Hold Purpose, Intended Purpose, Strategic Purpose, Objective, vision, Mission, Goal, Help Achieve, Strategy, Strategic Planning, Strategic Action, Decision, Assumption, Critical Assumption, Non-Critical Assumption, Influence Factor, Critical Influence Factor, Non-Critical Influence Factor, Critical Success Factor, Risk.
Marketing	Sale, Potential Sale, For Sale, Sale Offer, Vendor, Actual Customer, Potential Customer, Customer, Reseller, Product, Asking Price, Sale Price, Market, Segmentation Variable, Market Segment, Market Research, Brand Image, Feature, Need, Market Need, Promotion, Competitor.
Time	Time Line, Time Interval, Time Point.

Table 2-1 Enterprise Ontology Terminology

This table illustrates the main conceptual components within the “Enterprise Ontology”. These include Activity, Organization, Strategy, Marketing, and Time along with their related concepts.

The *Activity* section is intended to capture any term related to performing an action within an enterprise. Two central concepts in the *Organization* section are *Organizational-Unit* and *Legal-Entity*. The former needs to be recognized only within the organization,

however; the latter is also recognized outside the organization by legal authorities and has responsibilities and rights. All other terms in this section are defined around these two concepts. The *Strategy* section aims to capture all the terms related to the enterprise strategic plans, such as *Purpose* and *Plan*. Terms related to *Marketing* are categorized in the *Marketing* section. This section includes terms such as *Sale*, *Sale-Price*, *Vendor*, and so on.

The authors state that the Enterprise project, and specifically, the showcased ontology is used in real cases by organizations promoting and involved in project development. For example, Lloyd's Register uses the results obtained from this project to allow for more effective modeling and re-engineering of business processes for strategic planning. IBM UK utilizes the results for modeling its own internal organization. However, we are unable to locate any detailed account of these implementations or any user feedback. The authors fail to report any evidence of taking into account usability issues in their development process.

2.2.5.2 Gruninger and Fox Methodology (TOVE Methodology)

The second ontology building methodology to be reviewed is presented by Gruninger et al. in [Gruninger and Fox 1994; Gruninger and Fox 1995; Fox and Gruninger 1997; Fox and Gruninger 1998].

Gruninger and Fox define a method based on their experiment in developing an enterprise model called TOVE (TOronto Virtual Enterprise). They based their work on the ontology definition proposed by Gruber [Gruber 1993] "an ontology is a formal description of entities and their properties, relationships, constraints, and behaviors." In short, their approach can be described in the following stages [Figure 2-14]: defining motivations and requirements, identifying ontology terminology (informally and formally), defining constraints on the derived terminology, and finally, evaluation (examination of ontology completeness).

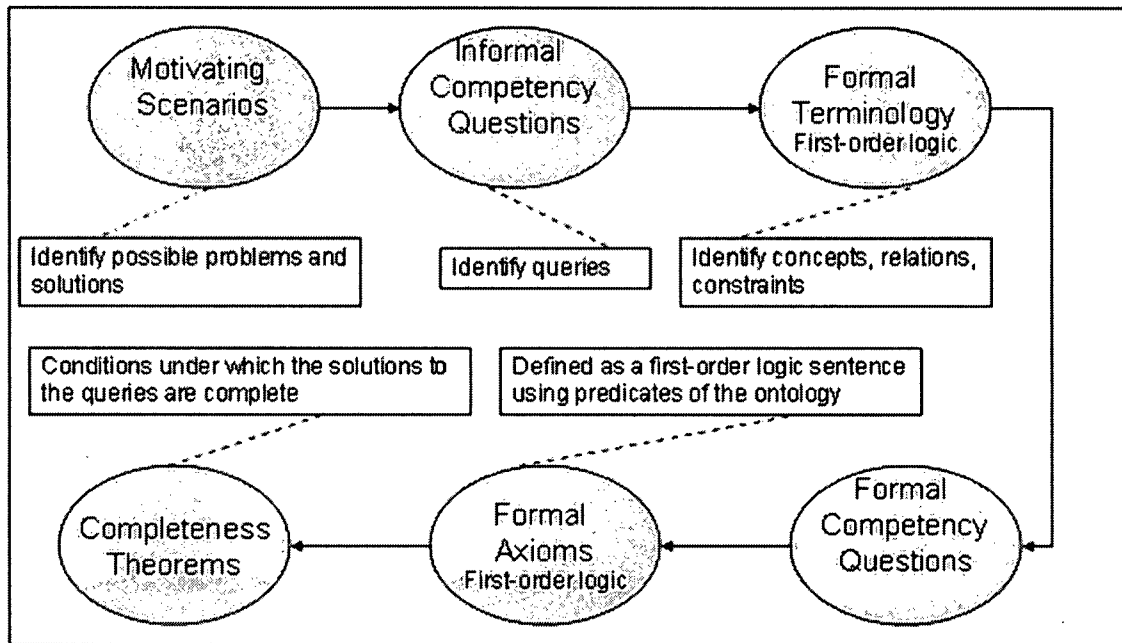


Figure 2-14 Gruninger and Fox Methodology (TOVE Methodology)

This figure illustrates the ontology development process in the “TOVE Methodology” proposed by Gruninger and Fox.

The authors supplement their approach for guiding the design of ontologies with a framework to evaluate the adequacy of these ontologies.

The steps proposed for this approach are explained here:

- Identify Motivating Scenarios

According to Gruninger et al., we need to develop a new ontology or extend an existing one when the existing ontologies do not address the problems in the applications.

Gruninger and Fox define such problems as *motivating scenarios*. According to TOVE methodology, it is fundamental that any ontology development or extension proposal has a motivating scenario and a set of possible solutions for the problem stated in this scenario. An informal, natural, language text terminology of the concepts and relations in the ontology can be derived with the help of these scenarios.

- Create Informal Competency Questions (Ontology Requirements)

Based on the motivating scenarios acquired in the prior stage, a set of questions arises.

These *competency questions* are the queries that the ontology must provide answers for

when completed. These questions justify the choice of concepts and relations in the ontology. At this stage, these questions are expressed in natural language text and are rather informal. Therefore, they are called *informal competency questions*. According to Gruninger and Fox, the competency questions should be organized and defined in hierarchical levels. This means that responses to lower level queries are required to also answer higher level (more general) queries.

- Provide a Formal Ontology Specification

Once the informal competency questions are in hand, the ontology terminology can be obtained from these questions. This terminology includes the concepts, their properties, and the relations amongst them. Gruninger and Fox opt for first-order logic to represent the terminology and competency questions in a formal language. Concepts, attributes, relations amongst them, and the constraints on them must be formally defined as well. The output of this phase is a specification for the definition of the terms and the constraints on them (definition). These are referred to as *axioms*.

- Examine the Completeness of the Ontology

According to Gruninger et al., by using the completeness theorem and a set of formal competency questions, we can evaluate the adequacy of the defined axioms for the ontologies by determining whether each and every competency question can be stated by a set of axioms defined for the terminology [Gruninger and Fox 1994].

Analysis

- Level of Detail and clarity: how well is the methodology described?

The primary focus of this work is on the formalization and evaluation phases. The approach does not explain other phases in depth, for example, it does not have any specific scoping phase. The authors do not address the issue of integration, even though the created ontology they describe is comprised of several separate ontologies.

- Approach to identifying ontology terminology

The authors propose to use motivation scenarios and competency questions for extracting the terminology, and adopt the middle-out approach for categorizing these terminologies.

However, neither of these methods is described in full detail. Furthermore, motivating scenarios are only one of several ways in which the tasks can be described. In order to obtain a more complete methodology, it becomes necessary to include other types of representation.

- Generality

Since the method uses competency questions and motivation scenarios, it is loosely application dependent. This is because these competency questions and motivation scenarios are dependant on the intended application.

- Ontology evaluation

Completeness theory is used to evaluate whether the ontology is able to answer all the competency questions. This theorem is based on whether the developers can formulate the competency questions, based on the defined axioms within the ontology.

At the time (1995), the authors were the first to propose a formal evaluation of an ontology. The process consists of representing competency questions formally, and then by proving completeness theorems with respect to those questions based on the logical representation of concepts. This project makes an important step in an underdeveloped area of ontology research: formal representation and evaluation.

- Usability

According to the authors, the goal of the TOVE project [TOVE] is to introduce a generic, sharable and reusable computational representation of an enterprise model applicable across a variety of enterprises.

To represent the knowledge within the enterprise an integration of the following ontologies is carried out (Figure 2-15) [Fox and Gruninger 1998].

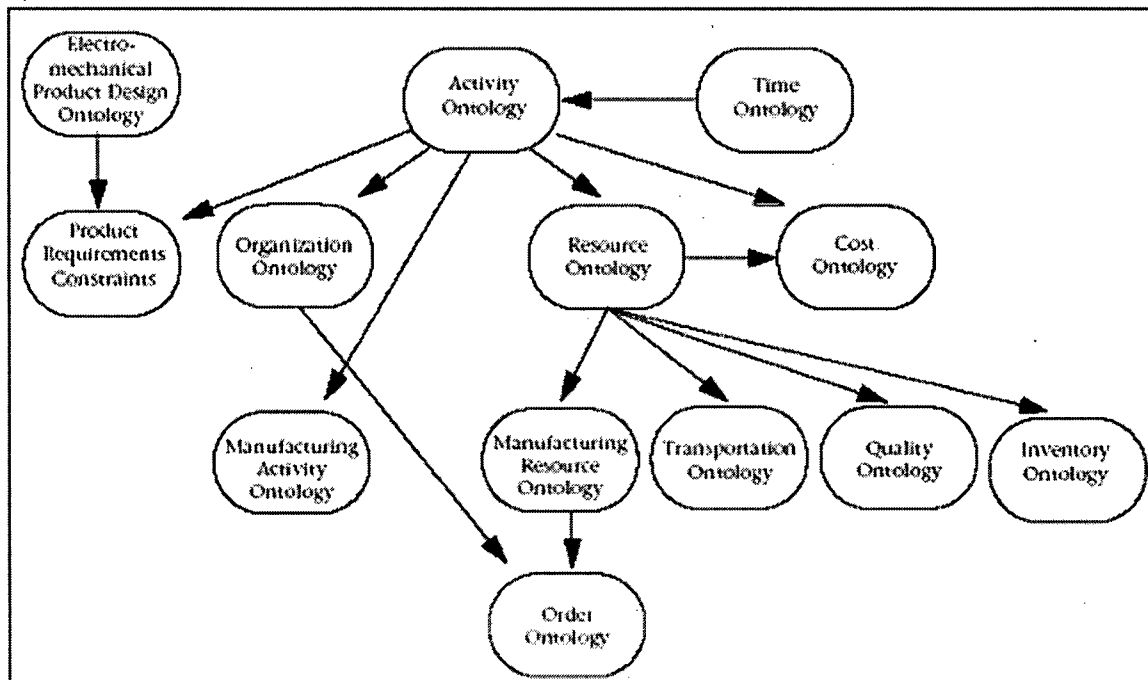


Figure 2-15 TOVE Ontologies
(adapted from [Fox and Gruninger 1998])

- Activity ontology [Guarino and Pinto 1995]

The activity ontology is developed to capture the actions performed within an enterprise, such as planning and scheduling.

- Resource ontology [Fadel, Fox et al. 1994]

The resource ontology is developed for use in manufacturing enterprises for modeling resources. Its aim is to enable applications to reason about the nature of resources and their availability.

- Organization ontology [Fox, Barbuceanu et al. 1996]

The focus of this ontology is on organization structure, roles, authority and empowerment. Figure 2-16 and Figure 2-17 illustrate the basic concepts of the organization ontology and part of the role hierarchy within the organization.

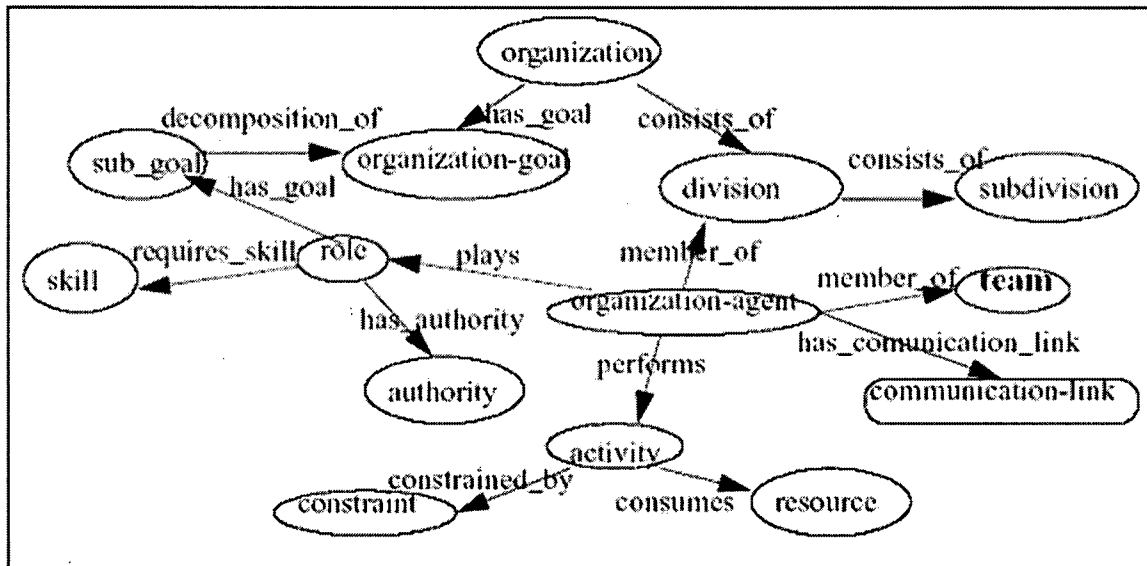


Figure 2-16 TOVE organization concept taxonomy
(adapted from [Fox, Barbuceanu et al. 1996])

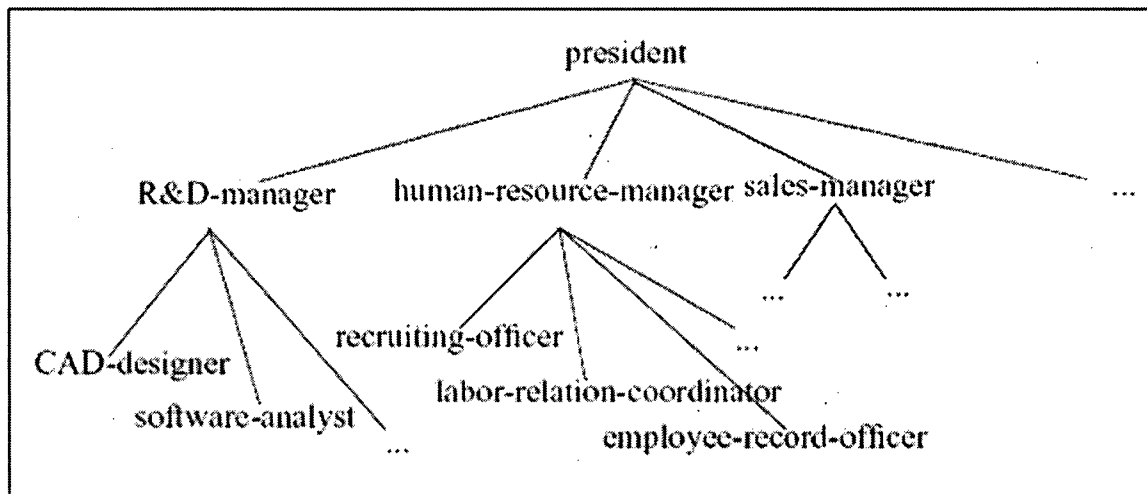


Figure 2-17 Part of hierarchy of roles in organization ontology (TOVE)
(adapted from [Fox, Barbuceanu et al. 1996])

- Quality Management Ontology [Kim, Fox et al. 1999]

This ontology is developed to formally organize the body of knowledge regarding quality related concepts. The quality management ontology is used in conjunction with the ISO 9003 Micro-Theory in a lamp-making company to test the ISO 9003 compliance of the company. According to the authors, this ontology can be used later to develop

applications to reason about products, processes, or system-wide quality factors, so as to facilitate better quality management decisions.

The authors of this methodology claim to have implemented a series of practical ontologies using this methodology. However, they fail to report any evidence of accounting for usability issues in their development process or their final ontologies. This leads us to conclude that the authors do not adequately address usability issues.

2.2.5.3 Uschold and Gruninger Methodology (Unified Methodology)

In [Uschold 1996] and [Uschold and Gruninger 1996] Gruninger and Uschold merge the independently developed TOVE and Enterprise methodologies (discussed in the previous sections) and propose a new approach they call *Unified Methodology*. Both approaches emphasize separating the informal and formal stages in ontology development. One takes a formal view; the other, an informal view. These two approaches are different in the ontology capture and evaluation phases but are intended to be used in the same domain (enterprise domain).

In this combined approach, the Enterprise Methodology [Uschold and King 1995] is used to capture the informal ontology with specific emphasis on issues such as scoping, handling ambiguity, reaching agreement and producing definitions. Likewise, the TOVE approach [Gruninger and Fox 1995] is applied to transform the previously obtained informal ontology into a formal ontology.

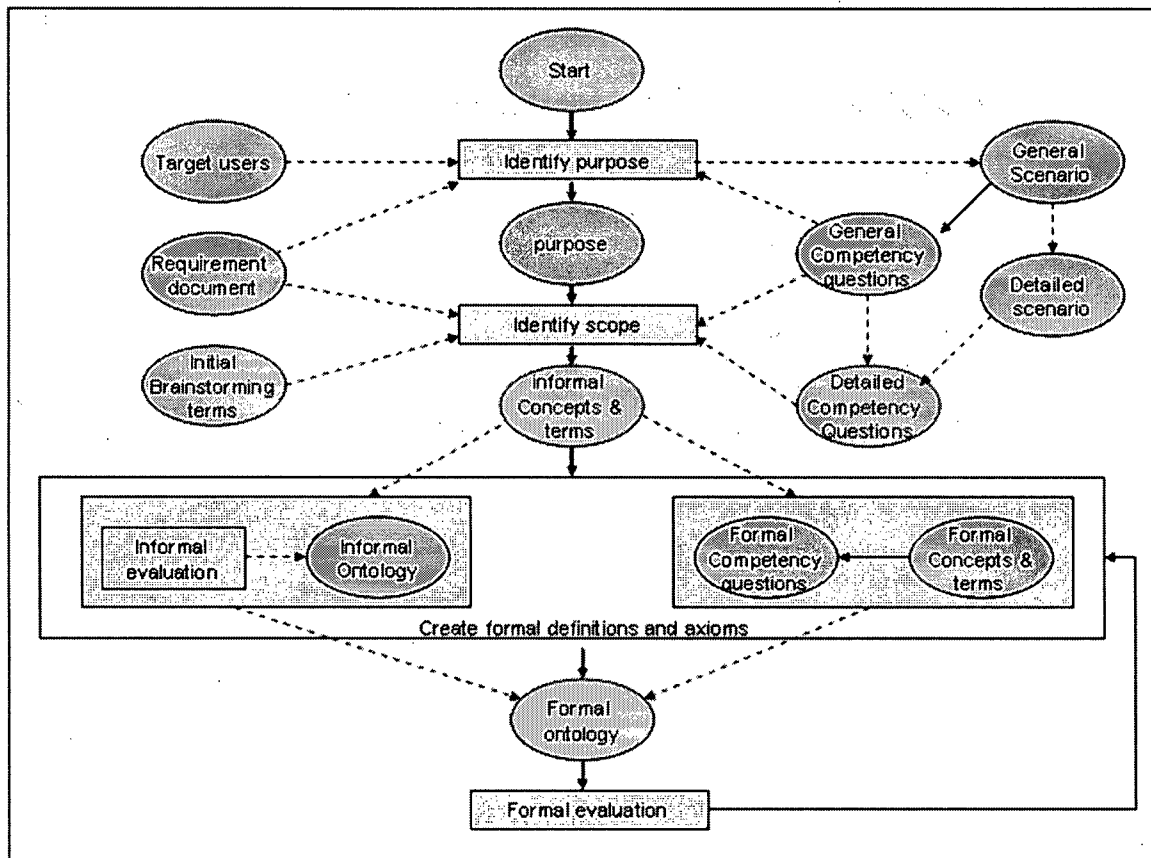


Figure 2-18 Gruninger and Uschold Methodology (Unified approach)

This figure shows the different approaches that can be taken with respect to the degree of ontology formality.

Uschold and Gruninger take into account that the applicable methods for building an ontology vary with respect to the degree of ontology formality, its purpose, users, and the domain that it intends to capture. In short, they vary depending on the particular ontology being built.

While building an ontology, one needs to decide on the level of formality desired. This is mainly determined by identifying the purpose and intended users of the ontology. The formality ranges from highly-informal (e.g. glossaries) through to structured but informal (e.g. the text version of Enterprise Ontology), semi-formal (e.g. formal version of Enterprise Ontology) to rigorously formal (e.g. TOVE). In general cases, the degree of formality required is proportional to the degree of automation in the tasks that the ontology is intended to support. The chosen method for building an ontology therefore depends on the degree of formality, which in turn depends on the intended purpose of the

ontology. For instance, for non-technical users, mainly domain experts, when the primary purpose is to build a shared vocabulary to facilitate communication between human users, an informal terminology glossary may be adequate. However, in order to allow inter-operability or reuse and sharing of knowledge, the ontology needs to be represented more formally. Some cases require that both formal and informal ontologies be built; this is in order to address the needs of both technical and non-technical users. In other cases it may still be useful to build a complete informal ontology, even when its intended users are technical. This ontology might be later used to document and define a formal encoding. Figure 2-18 illustrates the different approaches that can be taken in regard to the level of formality. The authors' goal is to provide a general and flexible guideline which can be applicable in different circumstances.

Analysis

The main concern with this work is the inclusion of both formal and informal approaches, and the provision of a comprehensive and unified guideline according to the level of formality that the ontology developer seeks.

Even though this merging covers some of the shortcomings of previous methods, it still suffers from the same drawbacks we have mentioned before. Although this method now has more comprehensive capture, formalization, and evaluation phases, it still lacks a clear approach for integration and does not support general usability criteria.

2.2.5.4 Gomez and Fernandez Methodology (METHONTOLOGY)

The last methodology to be analyzed is presented by Gomez-Perez et al. in [Perez, Lopez et al. 1996; Lopez, Perez et al. 1997; Blazquez, Lopez et al. 1998; Lopez, Perez et al. 1999; Lopez and Perez 2002]. According to Gomez-Perez et al., their goal for this work is to transfer ontology development from an art to an engineering discipline. The authors aim to define a standardization of the ontology life cycle (development) with respect to the requirements of the Software Development Process (IEEE 1074-1995 standard). This work has its origin in knowledge engineering methodologies for developing knowledge-based systems. The proposed methodology, "METHONTOLOGY framework" includes the clarification of the ontology development process, identifying a set of activities that

must be performed to build an ontology, an ontology life cycle that presents the order in which such activities must proceed, and presents techniques which can be used at each stage of the ontology life cycle. According to Gomez-Perez et al., all the activities that are carried out in an ontology development process may be classified into one of the following three categories:

- Project management activities
 - Planning: identifying the main tasks to be carried out, their order, completion time, and the required resources (people, hardware, and software)
 - Control: assuring that the intended tasks fulfill the requirements they are designed for
 - Quality assurance: guaranteeing that the quality of each and every result (ontology, software, and documentation) of these processes is up to the required standards
- Development oriented activities
 - Specification; identify the purpose and scope of the ontology. In the METHONTOLOGY framework, this includes: identifying the purpose of the ontology and its intended usage, scenario of use, users, level of formality of the proposed ontology and its scope (which contains its terminology, terminology's characteristics and granularity).
 - Conceptualization; organize the domain knowledge into a meaningful model at the knowledge level.
 - This work provides a guideline (set of tasks that must be performed) to organize the domain knowledge in a conceptual model (conceptualization). This conceptual model will describe the terminology of the domain of discourse, including concepts, their relations, structure, and inter-linked relations. The conceptualization guideline introduced in METHONTOLOGY provides a set of methods and techniques which not only assist the users in organizing the knowledge in a conceptual model but also helps users decide what to identify,

describe, formalize, and how to do it. Their approach for conceptualization is described in more detail in [Perez, Lopez et al. 1996].

- Formalization; representing the conceptual model as a formal model. This can be done with the use of description logic (logic-based) or frame-oriented (frame-based) representation systems.
- Implementation of the ontology in a formal language: this means implementing formal models in a computational language.
- Maintenance of the implemented ontology
- Support activities
 - Knowledge Acquisition; obtain knowledge about a domain of discourse. They consider knowledge acquisition as an independent process in the development of an ontology. According to the authors, most of the acquisition is performed during the requirement specification stage and continues as the development procedure moves forward but to a lesser degree. The techniques that they propose for this stage include: a. Formal and Informal interviews with domain experts. b. Applying text analysis with the help of experts, books and handbooks. c. Using knowledge sources such as experts, books, handbooks, figures, tables and other ontologies. d. Integration; reuse existing ontologies when possible. This will involve finding the ontologies and testing whether the definitions to be borrowed are coherent with the terminology proposed in the conceptualization phase.
 - Evaluation; they define evaluation as “making a technical judgment of the ontologies, their associated software environment, and documentations with respect to a frame of reference.” According to Gomez-Perez et al. evaluation should take place during each stage and between the stages throughout the ontology life cycle.
 - Documentation; to be able to use and reuse/share an ontology one must provide a detailed, clear, documentation for each and every step of the ontology development process.

- Configuration management; keep track of every version of documentation, software and ontology code to be able to track changes.

The authors refer to the order of steps that should be taken to build an ontology as the “ontology life cycle.” This *life cycle* closely resembles the classical software engineering life cycle (see Figure 2-19).

METHONTOLOGY defines the following stages for the ontology life cycle: specification, conceptualization, integration, implementation, evaluation, and maintenance. Figure 2-19 illustrates that documentation, evaluation, knowledge acquisition, and integration are carried out through the entire life cycle of the ontology development. Earlier versions consider an integration stage between the formalization and implementation stages, but later versions consider this to be a support activity. For each and every phase of the ontology life cycle, certain documentation is produced.

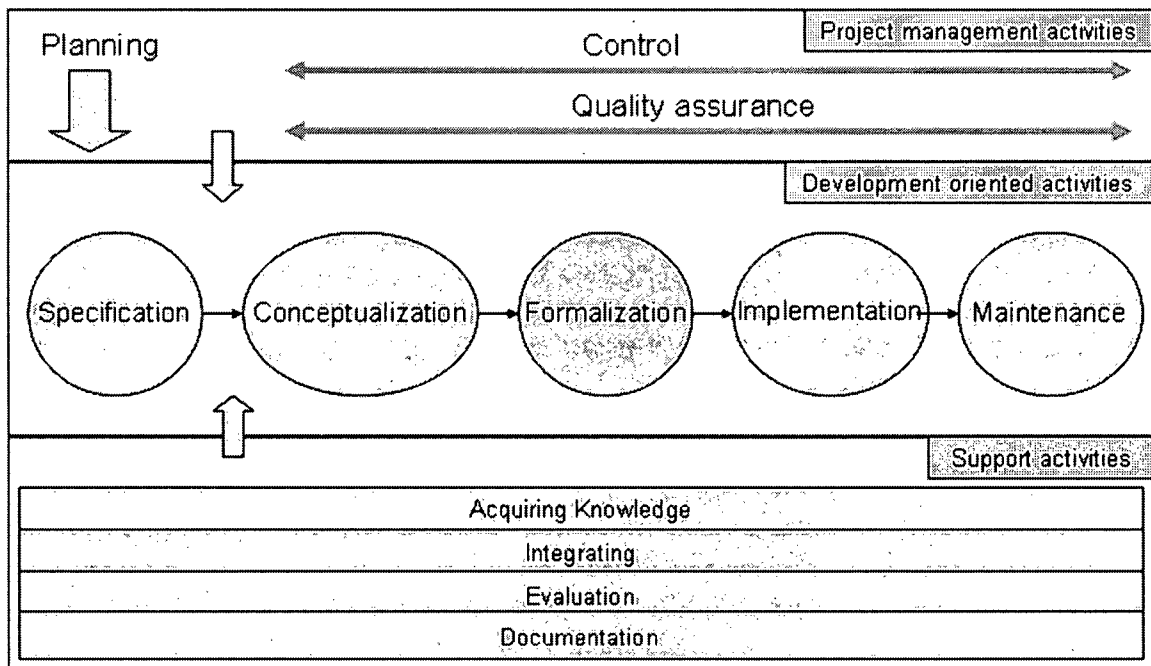


Figure 2-19 METHONTOLOGY ontology life cycle

This figure illustrates the ontology development process according to the METHONTOLOGY framework. In this work, activities performed throughout the ontology development fall into one of three categories: project management, development oriented, and support activities.

According to the authors, since the proposed model gives the user the flexibility to modify, insert, and delete concepts and relations to the ontology at any time, the ontology

life cycle is an evolving process. In an evolving prototyping life cycle, the developer can go back from any stage of the development process into any of the previous stages. As long as the ontology does not satisfy the specified requirements, this process can be repeated.

Analysis

- Level of Detail and Clarity: how well is the methodology described?

Authors describe the steps that should be taken in depth and in a clear manner. The different techniques used are described in depth as well. However, the authors concentrate more on the conceptualization phases than the formalization and implementation stages. Therefore, their approach towards conceptualization is presented in a clearer more in depth manner.

- Approach to identifying ontology terminology

In this work, the authors adapt knowledge acquisition methods proposed in the knowledge engineering methodologies for developing Knowledge Based Systems.

The proposed approach for categorizing terminologies is the middle-out approach.

Within the conceptualization phase, very specific low-level steps are taken which can target domains where the intention is to capture terminologies, rather than functionality or behavior. In this case, the methods that they describe are well suited to these kind of domains.

- Generality

The general approach proposed is application independent. However, when it comes to the detail of certain conceptualization sub-steps, the techniques provided are somehow specific to the domain ontology they develop, and not applicable to other domains in a straightforward manner.

- Ontology evaluation

The authors adapted the “evolving prototype life cycle” for developing ontologies.

Within the evolving prototype the evaluation takes place along the entire life cycle of the ontology. The evaluation is performed throughout each stage and between the stages all

along the ontology life cycle. They provide a set of guidelines to evaluate the completeness, consistency and redundancy of the created ontology [Perez 1994b]. This work is based on what has been done before in the area of Knowledge Based Systems evaluation. However, this process is not described in detail and rather a simple sketch of the general idea is provided.

- Usability

According to the authors, several ontologies in different domains built using the proposed method. One of these is the Chemical ontology, which consists of knowledge in the domain of chemical elements and crystalline structures [Lopez, Perez et al. 1999]. Figure 2-20 shows part of the planning and scope documentation for the Chemical ontology and Figure 2-21 illustrates the Chemical ontology concept classification tree.

Domain: Chemical	
Date: May 15, 1996	
Developed by: Asunción Gómez-Pérez and Mariano Fernández López	
Purpose:	Ontology about chemical substances to be used when information about chemical elements is required in teaching, manufacturing, analysis, and so on.
Level of formality: Semiformal.	
Scope:	List of 103 elements: <i>lithium, sodium, chlorine, mercury,</i> List of concepts: <i>element, halogen, noble gas, semimetal, metal, third-transition metal,</i> Information about at least the following properties: <i>atomic number, atomic weight, electronegativity, melting point,</i>
Sources of knowledge:	
(a)	Three interviews with the expert.
(b)	The following books: [Handbook, 84-85] <i>Handbook of Chemistry and Physics</i> , 65th ed., CRC Press Inc., 1984-1985.

Figure 2-20 Ontology requirements specification document for the Chemical ontology
(adapted from [Lopez, Perez et al. 1999])

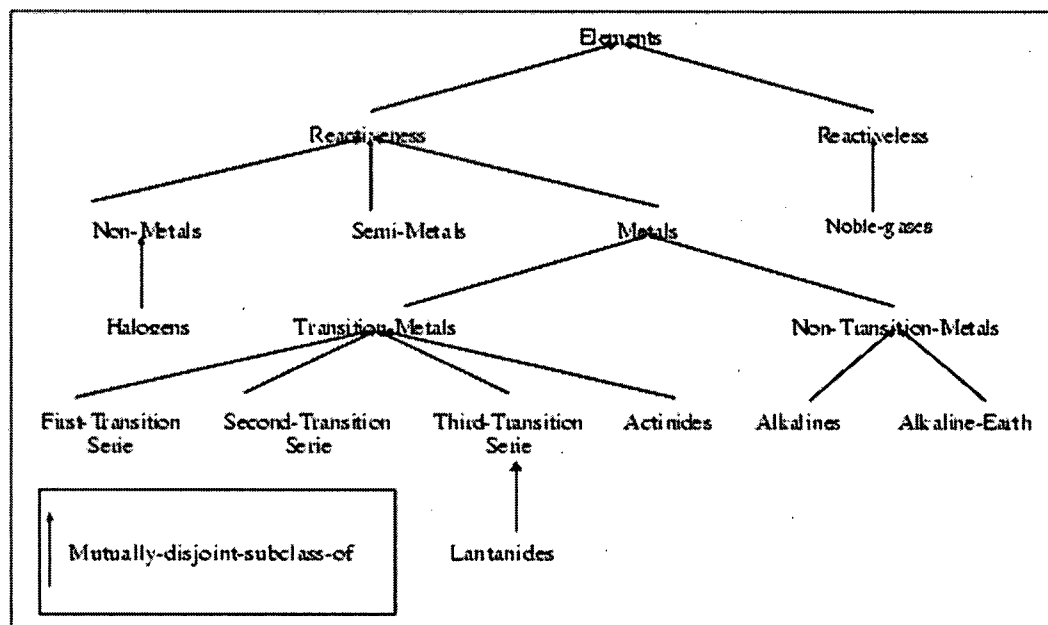


Figure 2-21 Part of the Chemical ontology
(adapted from [Lopez, Perez et al. 1999])

According to Fernandez and Gomez, the Chemical ontology is being used in conjunction with several applications. One of these applications is an information-retrieval system that allows Spanish speaking users to consult and access, in their own language, the information contained in this Chemical ontology. The system uses this ontology in conjunction with a linguistic ontology to generate Spanish text descriptions as responses to the queries in the chemistry domain. Another application, which allows students to learn and test their skill in the domain of chemistry, uses Chemical ontology as the source of its knowledge. However, the authors do not mention how much the end-users are involved in the development process or whether their results are satisfactory to them. This leads us to conclude that the authors do not fully address usability issues.

Analysis Summary

Even though the methodologies presented here represent the state of the art as far as ontology building methodologies go, none of them has been standardized and established long enough to have a significant user community [Beck and Pinto 2003]. Therefore, a standard, universally accepted approach is yet to be defined.

Although there are some differences between the methodologies described above, most of these approaches include the following stages: specification, conceptualization, implementation, evaluation and integration. However, every methodology presented places emphasis on a different area or stage.

There are three models to present an ontology life cycle: water fall, iterative, and evolving. Figure 2-22 illustrates how the stages go along each of these life cycle models.

The evolving prototype gives the developer the flexibility to modify, insert, and delete definitions in the ontology at any time. The developer can go back from any stage of the development process into any of the previous stages. As long as the ontology does not satisfy the specified requirements, this process can be repeated [Beck and Pinto 2003].

An iterative life cycle consists of several complete iterations or prototypes of an ontology to be developed. The water-fall approach tries to attain the final ontology in a single attempt with a sequential progression through the life cycle.

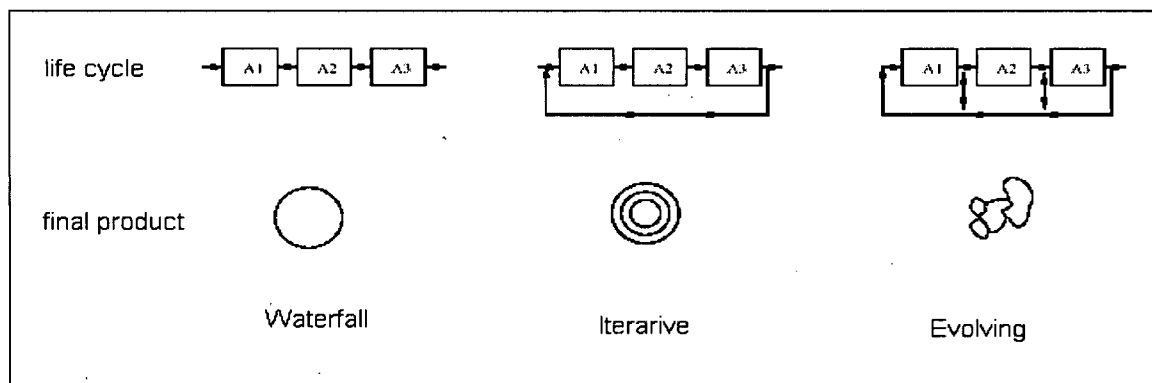


Figure 2-22 Water-fall, iterative, and evolving approach
(adapted from [Beck and Pinto 2003])

The methodologies described in this section use either water-fall life cycle models (TOVE and ENTERPRISE) or the evolving prototype life cycle (METHONTOLOGY). When developing a small ontology with clear precise requirements, a water fall approach may be sufficient, whereas when it comes to building large ontologies with evolving requirements to begin with, the evolving prototype life cycle may be more appropriate. In real world scenarios, it is rarely the case that the requirements are clearly defined at the

outset. It is for this reason that an evolving prototyping life cycle may be more reasonable to use rather than a waterfall or an iterative model.

None of the described methodologies addresses the issue of usability. In some part, this is due to the fact that for most of the ontologies that have been developed, the users and developers are the same people. In other words, ontologies are built to be used within the developer's institution and little to no regard to traditional usability issues was given.

Another probable reason for the lack of usability studies in the presented ontologies is the innovative aspect that these ontologies still carry. For the most part, the researchers have focused on the ontology building methodologies as their main research goal. This in turn leads to work that is sometimes truncated beyond the implementation phases. The studies presented here are still in a stage where the main focus is on a research and experimental level and therefore little concern for usability is given in these reports. From what is reported in the studies, it appears that an engineering group was appointed to collect and analyze the requirements from the users and later delivered a final version without further consultation from them.

As a final word, we should note that there is no one exact method to build an ontology. One can always find feasible alternative ways. The most appropriate techniques almost always depend on the characteristics of the domain the ontology is intended to capture and the to-be-developed application(s) [Noy and McGuinness 2001]. We believe that a comprehensive methodology should present guidelines to aid ontology developers to select amongst the available methods, according to the given circumstances.

3 Methodology

As discussed in the previous chapter, it is quite clear that building ontologies is still a matter of craft rather than a well defined engineering process. To this day, there is no agreed upon standard methodology for building ontologies. For our intended purposes, we have adapted different methodologies for the different steps in the ontology development process.

This chapter begins by presenting the steps taken to building our ontology and the adopted techniques used for each step. The first step is a general explanation of how knowledge acquisition is carried out throughout the process. This is followed by a description of our conceptual ontology creation process. Later, we present the formalization and instantiation of the conceptual ontology by utilizing a tool called Protégé. This tool was selected from several available, due to characteristics described in this chapter. A comparison of the available tools is presented in this chapter as well.

3.1 Selected Approach

In our opinion, the ontology building process can be divided into the following stages:

- Identifying the purpose, scope, and users
- Ontology Building
 - Domain analysis and knowledge acquisition
 - Integration of existing ontologies
 - Building a conceptual (informal) ontology model
 - Transferring the conceptual (informal) ontology into a formal (machine readable) ontology
- Ontology evaluation

For our implementation, we have chosen an evolving prototype for ontology development. In this model, every step forms part of an evolving process. Each stage can have more than one iteration. At each stage, it is possible to go back to any previous stage of the development process, in order to satisfy emerging requirements. This makes the evolving prototype useful for developing any ontology from scratch. Figure 3-1

illustrates how these steps are related, and in what order they can be performed to complete the ontology building process.

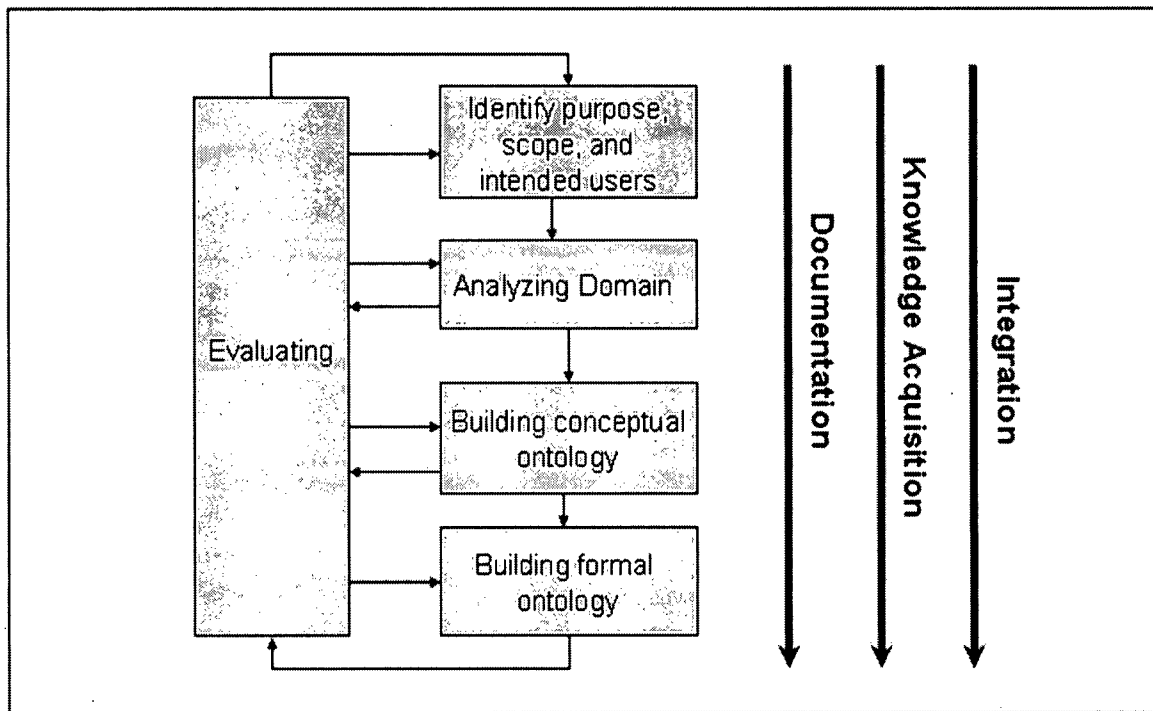


Figure 3-1 Our Ontology Development process

This diagram explains the steps taken to build our ontology. We used an evolving prototype to build our ontology. Integration, Knowledge Acquisition, and Documentation are carried out throughout the entire development process.

We consider the following ontology design criteria for each and every stage of the development process [Gruber 1995; Swartout, Patil et al. 1996].

- **Clarity:** An ontology should provide a precise definition of terms. The terms should make sense to their intended users and not seem arbitrary or ambiguous. The definition must be objective and independent of context. Where possible, definitions should be complete, and not partial. Additionally, every definition should be documented using natural language.
- **Coherence:** An ontology should be coherent. This means that it should allow making consistent inferences of definitions. For example, if a sentence that can be inferred from the axioms contradicts a definition or example given informally, then the ontology is incoherent.

- **Extensibility:** An ontology should be designed to support revision and extensions. It should offer a conceptual foundation for a range of anticipated tasks. Also, it should allow inclusion of new terms (both for generalization and specialization) based on the existing vocabulary without requiring revision of the existing definitions.
- **Minimal encoding bias:** An ontology's conceptualization should be specified at the knowledge level and be independent of any particular representation language. Minimal encoding bias enables knowledge-sharing and ontology reusing amongst different applications, which might use different representation languages.
- **Minimal ontological commitment:** An ontology should make as few claims as possible about the world being modeled; this allows the involved developers liberty to specialize and instantiate the ontology as necessary. These claims, called ontological commitments, facilitate intended knowledge sharing activities. One way to minimize the ontological commitment is to define only those terms that are essential to the communication of knowledge.

These criteria often present conflicting interests. We have considered this while developing our ontology and tried to maintain a balance between all the criteria presented here.

3.1.1 Identify the purpose, scope, and intended users

In this stage of the ontology development, the purpose, scope and intended users of the ontology are identified. In order to do so, we provide answers to the following list of questions:

1. For what purposes is the ontology being developed? (Purpose)
2. What is the domain that the ontology will cover? (Scope)
3. What types of questions must the information in the ontology answer? (Purpose and Scope)
4. Who will use the ontology? (identify and characterize the range of intended users)
5. What is the level of formality required for the ontology? (Scope)

The result of this phase is a set of answers to the stated questions, or in other words, “the requirement specification document.”

Our approach answers the selected questions in a manner similar to that proposed by Uschold and Gruninger [Uschold and Gruninger 1996].

In order to answer question 1 (*For what purposes is the ontology being developed?*), we provide a list of activities that the ontology is intended to support. The following is a list of purposes for building ontologies taken from literature on the subject [Uschold 1996; Noy and McGuinness 2001]:

- Facilitating communication between people *
- Allowing interoperability amongst systems
- Providing Reusability *
- Knowledge Acquisition *

The selected set of purposes to be supported by our ontology are explained in section 4.1.

Question 2 (*What is the domain that the ontology will cover?*) and question 4 (*Who will use the ontology?*) relate to the scope and intended users of the ontology. By scope we mean the domain that the ontology will cover. This includes identifying the granularity and generality of this domain. This is further investigated in the next stage of development when the domain analysis and knowledge acquisition are done.

In order to answer question 3 (*What types of questions must the information in the ontology answer?*) we identify fairly general motivating scenarios and competency questions and use these to help clarify specific applications and activities to be supported by the ontology. We use a set of motivating scenarios to define the problems that the ontology addresses and investigate possible solutions for these problems [Gruninger and Fox 1995]. This is followed by creating a set of informal competency question that we expect the ontology to be able to answer. At this stage, these questions are fairly general and do not introduce all the aspects or concepts of the ontology.

Later, these motivating scenarios and competency questions can be used to extract the concepts, terms and semantic of the ontology [Uschold and Gruninger 1996]. For our

ontology, these scenarios and questions are obtained from books in the history domain, brainstorming sessions, informal analysis of our text and inspecting available applications with similar purposes [Lopez, Perez et al. 1997]. The fifth and final question that we need to provide an answer for is “*What is the level of formality required for the ontology?*” Our approach has both informal and formal representations at different stages of development. First, we develop an informal model, which is later transformed into a formal representation, by using an ontology development tool.

3.1.2 Ontology Building

Some research has been done towards automatically extracting ontological knowledge from natural language text. The main purpose of these studies is to reduce the manual effort involved in building ontologies [Alani, Kim et al. 2003]. Up to now, most ontologies have been built manually. We use a manual approach since automatic extraction is still under development.

This step in the ontology development process concentrates on domain analysis and knowledge acquisition, and building conceptual and formal models of the ontology. These processes are further explained in the following sections.

3.1.2.1 Domain analysis and knowledge acquisition

The output of this phase is a set of concepts and terms covering the full range of information that the ontology must characterize to satisfy the requirements identified previously. In this phase, we use a glossary to gather the set of terms that must be included in the ontology [Lopez, Perez et al. 1997].

We use knowledge acquisition techniques such as brainstorming, in conjunction with informal analysis of the text to gather all potential relevant terms into a glossary of terms [Blazquez, Lopez et al. 1998]. The informal analysis of the text is carried out by providing more in-depth motivating scenarios and expanding the number and refining the granularity level of the competency questions.

Within these brainstorming sessions, no term is excluded. Therefore, we must have some way to trim the set of terms down to a desirable and manageable size. There are two main reasons for removing a term: lack of relevance or duplication [Uschold and King 1995].

The result of these sessions is an informal natural language text terminology. Since our model is an evolving process, in the first attempt, we extract a comprehensive list of terms without worrying about overlap between the concepts they represent, relations amongst the terms, or any properties that the concepts may have. At later iterations, this list can be modified.

In addition to identifying the set of terms, we also produce informal definitions for these terms. The procedure through which we obtain these definitions helps us address the problem of handling irrelevant, ambiguous or duplicated terms. We also have to make sure that all terms and the provided definitions for them purvey the desired meaning in the intended domain. The level of detail provided by these definitions must be consistent. In order to do this we use the following guideline [Uschold 1996]:

- Develop natural language text definitions, being as precise as possible.
- Maintain consistency with terms already in use, in particular:
 - make use of dictionaries and other technical glossaries
 - avoid introducing new terms where possible
- Indicate relationships with other terms similar to the ones being defined.
- Avoid circular references.
- Provide any additional information that may help in understanding the definition.
- Give examples where appropriate.

It is important to remember that Knowledge Acquisition is an activity which is carried out iteratively throughout the ontology development process [Lopez, Perez et al. 1997].

3.1.2.2 Investigate the possibility of using an existing ontology

Throughout the ontology building stages, we query existing libraries of ontologies, such as Ontolingua [Ontolingua], DAML [DAML], and the SUMO [SUMO] ontology libraries to search for similar or related ontologies which might be useful for us. The idea behind this is to look for relevant ontologies that can be integrated with our own ontology in order to speed up the development process as well as to gain a better insight of how to

build a particular area or set of concepts within our ontology. We also survey the ontology literature for relevant work in this area.

3.1.2.3 Building an informal ontology model

In this phase we structure the domain knowledge into a conceptual model. This means that the terms obtained from the previous step get organized and their correlations explained. This stage can be sub-divided into the following activities:

- At this stage we expand our glossary of terms contains all concepts, relations, their definition or meaning, and additional information such as examples to clarify the meaning where appropriate. This terminology represents the knowledge we intend to capture with our ontology forms our glossary of terms. The output of this stage evolves as the ontology development activity proceeds. Table 4-7 shows a part of the glossary of terms obtained during of our development process.
- Once we have a relatively complete glossary of terms, we identify concepts, relations within the concepts, and their attributes. We use the guideline provided in [Noy and McGuinness 2001] to do so. A short description of the guideline is presented here:
 - To define the concepts, we select the terms that express things having an independent existence rather than terms that describe these things. These terms are the concepts in our ontology and become anchors in the concept hierarchy.
 - After selecting all the concepts from the glossary of terms most of the remaining terms are likely to be properties of these concepts. For each property we must decide which concept it describes. These properties either specify an attribute of a concept or its relation with other concepts.
 - Attributes for these concepts can be identified by using motivating scenarios and competency questions in conjunction with the glossary of terms. These attributes define the internal structure and characteristics of the concepts.
 - The relations are mostly the verbs in the competency questions and motivating scenarios. These terms represent the associations that hold amongst the concepts in a domain.

The results are stored in document tables called the Concept Dictionaries [Blazquez, Lopez et al. 1998]. Table 4-8 illustrates a part of one of our concept dictionaries. At this stage, the concepts are structured into naturally occurring groups. For example, concepts most related to one another are placed within the same Concept dictionary. This helps us build a conceptual graph of the ontology where we can visualize these relations and groups.

- For the next step, we use the previously generated concept dictionaries and a middle-out approach to develop our graphical conceptual ontology model. The middle-out approach operates by identifying the most important concepts first and then generalizing or specializing these concepts within the group from that point. This approach was chosen amongst the other available approaches, since the concepts “in the middle” tend to be the more descriptive concepts in the domain [Noy and McGuinness 2001]. Our conceptual model not only represents the concept hierarchy taxonomy but also the other (other than taxonomic) relations that hold amongst the concepts within our domain.

A taxonomic relation can be interpreted as a “is-a” or “kind-of” relation. We organize the concepts into a hierarchical taxonomy by looking at cases where an instance of one concept is an instance of some other more general concepts. In other words, if a concept *A* is a super-concept of concept *B*, then every instance of *B* is also an instance of *A*. This means the class *B* represents a concept that is “kind-of” or a “type-of” *A*. Additionally, all sub-concepts of a concept inherit the properties of that concept.

We adapt a simple Entity Model (ER) to illustrate the concepts and relations amongst them at this stage. We call this graphical representation of concepts and relationships “the conceptual ontology.” Figure 4-6 illustrates a part of our conceptual ontology.

3.1.2.4 Building a formal ontology model

The next step in our approach is to build a formal ontology based on the conceptual model developed in the previous section. In order to do this we utilize an ontology development environment. We analyzed several well-known available ontology development tools to choose the one that best suits our purpose and selected criteria. A

comparison of these tools and the criteria used to select amongst them is presented in this section as well.

3.1.2.4.1 Selecting a development tool

A large number of ontology development tools are available nowadays. In order to select the most appropriate tool, it is necessary to evaluate these tools against a set of criteria important to the user. Some studies have been conducted to evaluate ontology development environments previously, those presented in [Duineveld, Stoter et al. 2000] and [WebOnto 2002] are the most comprehensive ones. In [Duineveld, Stoter et al. 2000] a comparative study of several ontology development tools is presented based on a proposed evaluation framework. This evaluation, called the WondetTools project, was the earliest systematic survey which evaluated and compared the ontology building tools. In [WebOnto 2002] a comprehensive and general study of ontology-based environments is presented. This survey intends to cover tools which support activities such as creating, integration, evaluation, and storage and querying. An evaluation framework for each type of tool is proposed. Eleven different ontology-based tools were compared against their corresponding framework.

These studies define an extensive set of criteria to be used for evaluating these tools.

Some of the aspects that this framework compared include the following:

- The ability to interact with other tools and to import and export ontologies in different formats;
- the expressiveness of the knowledge model;
- scalability and extensibility;
- usability.

Table 3-1 to Table 3-6 and Figure 3-2 show the results of the comparison and analysis.

Feature	Developers	Availability	Extensibility	Ontology Storage
Apollo	KMI (Open University)	Open Source	Plug-ins	Files
LinkFactory	Language & Computing nv	License on site or ASP	Yes	DBMS
OILEd	University of Manchester	Open Source	No	File
OntoEdit Free	Ontoprise	Freeware	Plug-ins	File
OntoEdit Professional	Ontoprise	Software license	Plug-ins	File DBMS
Ontolingua	KSL (Stanford University)	Free Web access	None	Files
Ontosarus	ISI (University of Southern California)	Open source and free Web access to evaluation version	None	Files
OpenKnoME	University of Manchester	Freeware	None	File
Protégé-2000	SMI (Stanford University)	Open Source	Plug-ins	File DBMS (JDBC) XML
SymOntoX	LEKS (IASI CNR)	Free Web access	No	XML
WEbODE	Ontology Group (UPM)	Software license and free Web access	Plug-ins	DBMS (JDBC)
WebOnto	KMI (Open University)	Free Web access	No	File

Table 3-1 a general description and design features of the tools within the OntoWeb framework (based on the work in [WebOnto 2002])

Feature	Graphical taxonomy	Graphical prunes (views)	Zooms	Collaborative working	Ontologies libraries
Apollo	Yes	Yes	No	No	Yes
LinkFactory	Yes	Yes	Yes	Yes	Yes
OILEd	No	No	No	No	Yes
OntoEdit Free	No	No	No	No	No
OntoEdit Professional	No	No	No	Yes	Yes
Ontolingua	Yes	No	No	Yes	Yes
Ontosarus	No	No	No	Yes	No
OpenKnoME	No	No	No	Yes	Yes
Protégé-2000	Yes	Yes	Yes	Yes	Yes
SymOntoX	Yes	Yes	No	Yes	Yes
WEbODE	Yes	Yes	No	Yes	No
WebOnto	yes	Yes	No	Yes	yes

Table 3-2 Usability features supported (WebOnto framework)

This table presents the different usability aspects supported by the different available ontology building tools. (adapted from [WebOnto 2002])

Feature	Built-in inference engine	Other attached inference engine	Constraint / Consistency checking
Apollo	No	No	Yes
LinkFactory	Yes	Yes (frozen ontologies)	Yes
OILED	Yes (FaCT)	No	Yes
OntoEdit Free	No	No	Yes
OntoEdit Professional	Yes (OntoBroker)	No	Yes
Ontolingua	No	Yes (ATP)	No
Ontosarus	Yes	Yes	Yes
OpenKnoME	Yes	No	Yes
Protégé-2000	Yes (PAL)	Yes Jess FaCT Flogic	Yes
SymOntoX	Yes	No	Yes
WEbODE	Yes (Prolog)	Yes (Jess)	Yes
WebOnto	Yes	No	Yes

Table 3-3 Inference services provided by the tools (WebOnto framework)
(adapted from [WebOnto 2002])

Feature	Interoperability with other ontology tools	Imports from languages	Exports to languages	KR paradigm of knowledge model	Axiom language	Methodology support
Apollo	No	Apollo meta- language	OCML CLOS	Frames (OKBC)	Unrestricted	No
LinkFactory	FastCode TeSSI	XML RDF(S) DAML+OIL	XML ,RDF(S) DAML+OIL HTML	Frames + FOL	Proprietary	Yes
OIEd	FaCT	RDF(S) OIL DAML+OIL	OIL , RDF(S) DAML+OIL SHIQ dooty HTML	DL (DAML + OIL)	Yes (DAML + OIL)	No
OntoEdit Free	OntoAnnotate Ontobroker OntoMat Semantic Miner	XML RDF(S) FLogic DAML+OIL	XML RDF(S) FLogic DAML+OIL	Frames + FOL	Yes (FLOGIC)	Yes (Onto- Knowledge)
OntoEdit Professional	OntoAnnotate Ontobroker OntoMat Semantic Miner	XML RDF(S) FLogic DAML+OIL	XML , RDF(S) FLogic DAML+OIL SQL-3	Frames + FOL	Yes (FLOGIC)	Yes (Onto- Knowledge)
Ontolingua	Chimera CML Model Fragment Editor Equation Solver Data structures inspector Expressions Evaluator OKBC	Ontolingua IDL KIF	KIF, CLIPS CLIPS sentential format CML ATP EpiKit, IDL KSL rule engine LOOM OKBC syntax PROLOG syntax	Frames + FOL (Ontolingua)	Yes (KIF)	No
Ontosarus	--	LOOM, IDL ONTO, KIF C++	LOOM, IDL ONTO, KIF C++	DL (LOOM)	Yes (LOOM)	No
OpenKnoME	GCE (GALEN CASE Environment) SPET	GRAIL GALEN IR	GRAIL, CLIPS HTML GALEN IR	DL (GRAIL)	Yes (GRAIL)	Yes (GALEN)
Protégé 2000	PROMPT OKBC JESS FaCT	XML RDF(S) XML schema OWL	XML, RDF(S) XML schema Flogics, CLIPS Java, HTML OWL	Frames + FOL + Metaclasses	Yes (PAL)	No
SymOntoX	--	--		OPAL	OPAL	Yes (OPAL)
WebODE	JESS, PICSEL OIEd ODEMerge ODE-KM	XML RDF(S) CARIN		Frames + FOL	Yes (WAB)	Yes (Methontology)
WebOnto	PlanetOnto ScholOnto	OCML		Frames + FOL	Yes (OCML)	No

Table 3-4 Interoperability, knowledge representation, and methodological support comparison
(adapted from [WebOnto 2002])

1. General
1. Evaluate the clarity of the interface
2. Evaluate the consistency of the interface
3. Evaluate the speed of updating after new data is inserted
4. Is there a good overview of the ontology?
5. Is the meaning of the commands clear?
6. Are the changes identifiable by a certain command clear to the user?
7. Evaluate the stability of the tool (crashes, etc.)
8. Does the tool require a local installation?
9. Evaluate the help-system
2. Ontology
1. Is it possible to use multiple-inheritance?
2. Is it possible to create exhaustive and/or disjoint decomposition? (+ease of doing this)
3.1 Does the tool check new data for consistency with the ontology?
3.2 At what level? (types, disjointness, etc.)
4. Are there example-ontologies available in the tool?
5. Does the tool provide libraries of ontologies that can be re-used? Through what operation? (inclusion, union, etc.)
6. Are there high-level primitives?
7. Is there information about the terms used in constructing an ontology in the help-system?
3. Cooperation
1. Does the tool allow synchronous editing of the same ontology by different users?
2. Are there ways to lock the ontology?
3. Is it possible to browse an ontology if it is locked?
4. Are the changes made by other users easy to recognize?
5. Is it possible to export the ontology's code in various formats?
6. Is it possible to import an ontology-description from another tool?

Table 3-5 Evaluation framework of the WonderTools Project
(adapted from [Duineveld, Stoter et al. 2000])

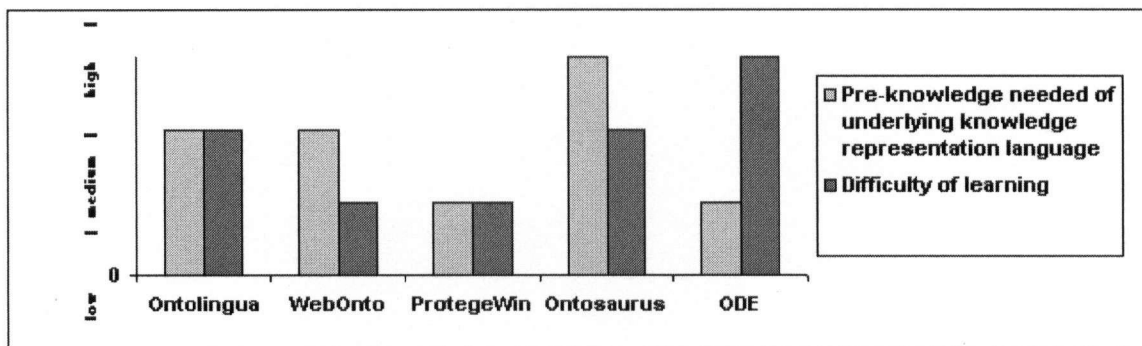


Figure 3-2 Results for ease of use comparison of the selected tools (WonderTools project)
(adapted from [Duineveld, Stoter et al. 2000])

Criteria	Ontolingua	WebOnto	ProtégéWin	OntoSaurus	ODE
1. General					
1.1 interface clarity	-	+	+	-	-
1.2 interface consistency	+	+	+	+	+
1.3 speed of updating	-	0	+	-	+
1.4 overview	0	+	+	-	-
1.5 meaning of commands	+	+	+	+	0
1.6 identifiability of changes	0	0	0	0	0
1.7 stability	+	+	+	+	-
1.8 local installation	No	No	Yes	Yes/no	Yes
1.9 help-system	+	-	+	+	-
2. Ontology					
2.1 multiple inheritance	Yes	Yes	Yes	Yes	Yes
2.2 decomposition types	+	+	-	+	+
2.3.1 consistency checking	+	+	+	+	+
2.3.2 level of checking	?	0	0	?	+
2.4 example ontologies	+	+	0	+	0
2.5 reusable ontologies	+	+	-	+	-
2.6 high-level primitives	+	+	-	+	+
2.7 ontological help	-	-	-	-	+
3. Cooperation					
3.1 synchronous editing	+	+	-	+	-
3.2 ontology locking	+	+	-	+	-
3.3 browsing when locked	+	+	NA	+	NA
3.4 change recognition	-	-	-	-	-
3.5 export facilities	+	-	-	+	0
3.6 import facilities	+	-	-	+	+

Table 3-6 Summary of the results obtained in the WonderTools Project

(adapted from [Duineveld, Stoter et al. 2000])

In this chart, a plus sign (+) means a positive result for the feature specified. In a similar manner, a zero (0) means a reasonable result, a minus (-) represents a negative result, and 'NA' stands for not applicable. We should consider that these results are based on a study carried out in 1999 and since then many of these tools have evolved. As an example of this, the later version of protégé-win, protégé-2000, supports cooperation and has import and export facilities.

The results gained from these evaluations helped us narrow down our list of candidates to: Ontolingua [Ontolingua], OntoEdit [OntoEdit], WebODE [WebODE], and Protégé-2000 [Protégé]. We followed this evaluation by developing a very simple ontology using these four environments. We eventually decided to use Protégé-2000 for the following reasons:

- Availability
- Ease of learning

- Flexibility
- Customizability
- Large and active user community
- Help availability (mailing lists, e-mail, FAQ, manual, annual workshops)
- Possibility of importing and exporting ontologies in different formats

The following section gives a short overview of Protégé-2000.

3.1.2.4.2 Protégé-2000

Protégé-2000 is the latest generation of the Protégé toolset. It was developed at Stanford University to facilitate knowledge acquisition activities [Grosso, Eriksson et al. 1999]. Protégé-2000 is an extensible, platform-independent environment for developing and editing ontologies and knowledge bases [Noy, Fergerson et al. 2000; Noy, Sintek et al. 2001]. Protégé was originally developed 16 years ago to support knowledge acquisition for a rather specialized medical expert systems. It has gradually gained popularity and now is used for many other purposes in different fields [Gennari, Musen et al. 2003]. It currently has more than 12,000 registered users all around the world [Protégé]. Protégé is open source and freely available for download under the Mozilla open source license.

There are several features that make Protégé stand out amongst ontology development environments [Protégé FAQ]:

- Intuitive and easy-to-use graphical user interface

Protégé-2000 provides a highly usable graphical and interactive user interface for the ontology development process. It allows ontology developers and domain experts to perform knowledge-management tasks such as creating and modifying reusable domain ontologies, customizing knowledge acquisition forms, and entering domain knowledge [Noy, Sintek et al. 2001].

- Scalability

Protégé-2000 uses a database as back-end and a cache mechanism to support the development of ontologies. This facilitates creation of large scale ontologies without suffering from performance loss as the ontology grows larger. There are examples of ontologies built using Protégé-2000 with 150,000 elements.

- Extensible plug-in architecture

One of the major advantages of the Protégé-2000 architecture is its modular, open construction. Protégé-2000 can be easily extended using plug-ins created to perform specific user-required tasks. This plug-in architecture allows developers to add customized components to provide new functionality. Plug-ins can be classified in the following categories[Grosso, Eriksson et al. 1999; Knublauch 2003]:

- Small user-interface components, called slot widgets, are designed to suit specific display or input requirements for a particular domain (Table 3-7).
- Custom back-end plug-ins, called storage back-end, allow users to import, export, and store ontologies in different formats (Table 3-8).
- Tab Plug-ins, called tab widgets, allow the inclusion of external applications to be used with Protégé-2000. An example of this is an application that presents a visualization of the ontology components (Table 3-9).

The Protégé Plug-in Library has a collection of contributions from developers all over the world. Table 3-7, Table 3-8, and Table 3-9 show a list of the available plug-ins and a brief description for each of them.

Slot Widgets	Developer	Description
Bayesian Network	Rafael Peñaloza	Make approximate inferences over ontologies where classes are arranged in "DAG" form
Calendar	KM center	Provides date entry via a calendar
Date	fMRI Data Center	Provides flexible date entry
Date	Fullbeing Hu	Generates the current date and time in a predefined format
Date	Phillip Cheng	Automatically fills in a slot value with current date and time
Display GIF Image	ICS (University of California)	Display GIF images from the file system in Protégé forms
Display Indirect Instances	Emotional Brain	Automatically displays all the indirect instances of a class
Get Author	Stanford University	Automatically fills in a slot value with a user name
Java Function Calls Slot Widgets and Storage Backend	Ifgi University of Munster	A group of plug-ins that allows the description of things in flux, activities, and processes which cause a knowledge base to change over a time scale
Measurement	fMRI Data Center	Provides flexible measurement entry
Media	University of South Hampton	Include and display video and audio files
Numeric Inference Engine	KM Center	Performs simple numeric inferences such as unit conversion and multiplication
Power-plant Control	Henrik Erikson	Edit control rules for a simple nuclear power plant simulator
Swap Values	Stanford University	Swap slot values between slots

Table 3-7 Slot widgets in Protégé-2000

This table presents a sample of the different *slot widgets* available in Protégé-2000. Slot widgets are small user-interface plug-ins designed to facilitate the input or display requirements of Protégé users.

Storage back-end	Description
DAML+OIL	Create and edit DAML+OIL ontologies with Protégé
Java Function Calls Slot Widgets and Storage Backend	A group of plug-ins that allows the description of things in flux, activities, and processes which cause a knowledge base to change over a time scale
OWL	Load, save, and edit Web Ontology Language (OWL) ontologies in Protégé.
RDF	Create, import, and save RDF(S) files in Protégé
UML	Store a Protégé knowledge base in UML.
XMI	Store a Protégé knowledge base as XMI files. XMI is a standard format for metadata exchange supported by OMG, the group that is responsible for standards such as UML, CORBA and the Common Warehouse Meta-model
XML Ontology	Store a Protégé knowledge base in XML. Classes are represented in a Schema file as types and instances are output into an XML document instance conforming to the generated schema
XML Schema	Store a Protégé knowledge base in XML. The backend will generate an XML Schema file which conforms to the Protégé knowledge model and an XML file which contains classes and instances.

Table 3-8 Storage back-end plug-ins in Protégé-2000

These plug-ins allow importing, exporting, and storing ontologies in different formats within Protégé.

Tab Plug-in	Description
Algernon	A rule based inference system implemented in Java and interfaced with Protégé. Performs forward and backward rule-based processing of frame-based knowledge bases
Bean Generator	Generate FIPA/JADE compliant ontologies from Protégé projects.
BeanShell	Interactively use the Protégé Knowledge-Base API.
CLIPS	Use the CLIPS Rule Engine from within Protégé
DataGenie	Read data from an arbitrary database into Protégé
Eligibility Screening	Find a set of clinical trial protocols in breast cancer for which a patient might be eligible.
ezOWL	Visual OWL (Web Ontology Language) editor for Protégé.
EZPal	Facilitate acquisition of Protégé Axiom Language (PAL) constraints without knowing the PAL language
Facet Constraints	Identify and fix Instances that have constraint violating facets.
FCA	Demonstrate the benefit of applying FCA while building an ontology
Flora	A query tab based on F-Logic
InstanceTree	Provides a tree view of frames referenced directly or indirectly by an Instance.
Jambalaya	Visualize Protégé ontologies with SHnMP (Simple Hierarchical Multi-Perspective).
JessTab	Allows the use of Jess and Protégé together.
OIL Classifier	Classify OIL ontologies with the FaCT descriptions-logic classifier
OKBC Tab	Import and export ontologies to and from OKBC servers via the OKBC interface
OntoViz	Visualize Protégé ontologies with the help of Graphviz graph drawing software
PAL	Express constraints about a knowledge base and make logical queries about the contents of a knowledge base.
Prolog	An integration of GNU Prolog for Java with Protégé
PROMPT	Manage multiple ontologies, merge separate ontologies to create a single coherent ontology, extract a part of an ontology, and move frames from an included to an including project
PSMLibrarian	Browse a Problem-Solving Methods library
Relations	Browse reified relations in the same way you can browse regular relations
Remote KB	Create your own tab to browse a remote knowledge source using a standard API and user interface
String Search	Search all values of type String in a knowledge base. Search includes classes, slot values, and metaclasses slots, instances,
TGViz Tab	Visualize Protégé ontologies using the TouchGraph library
TMTab	Build an ontology which may be exported as a topic map in XTM syntax
UMLS Tab	Search the Unified Medical Language System (UMLS) and annotate your current Protégé ontology with terms, concept ids, synonyms, relations, and other information from UMLS
WordNet Tab	Search WordNet and annotate your current Protégé ontology with terms, concept ids, synonyms, relations, and other information from WordNet
XML Tab	Extract Protégé ontologies from XML files and create XML files from Protégé ontologies

Table 3-9 Tab plug-ins in Protégé-2000

These plug-ins facilitate the inclusion of other applications into Protégé-2000.

As mentioned earlier one of the most important features in Protégé-2000 is its ability to provide interoperability with other knowledge representation systems. To achieve this, Protégé uses a knowledge model compatible with Open Knowledge-Base Connectivity protocol (OKBC). Protégé-2000 users can easily import and export ontologies using OKBC-compatible knowledge representation systems [Noy, Fergerson et al. 2000]. The Open Knowledge-Base Connectivity protocol (OKBC) is a common query and construction interface for frame-based systems. A knowledge base in a frame-based

system is built around the notion of frames. A frame is a primitive object that represents an entity in the domain of discourse [OKBC]. The knowledge model used by Protégé-2000 is frame-based.

A Protégé ontology consists of classes, slots, facets, and axioms [Noy, Fergerson et al. 2000; Noy and McGuinness 2001]. Classes describe the concepts in the domain of discourse. Slots refer to various attributes or properties of these concepts and the relations amongst them. Facets present the properties and some restrictions of slots. Axioms specify additional constraints on these slots. Within Protégé, an ontology along with a set of individual instances of classes with specific values for slots, form a knowledge base. In this work we will refer to an instantiated ontology as a knowledge base [Noy, Fergerson et al. 2000; Noy and McGuinness 2001; Knublauch 2003]. Figure 3-3 shows a screenshot of the Protégé-2000 ontology-editing environment. In the left pane, the class hierarchy is presented. The middle pane shows the list of instances for the selected class. The right pane displays some information about the selected class in detail such as class' slots and their values.

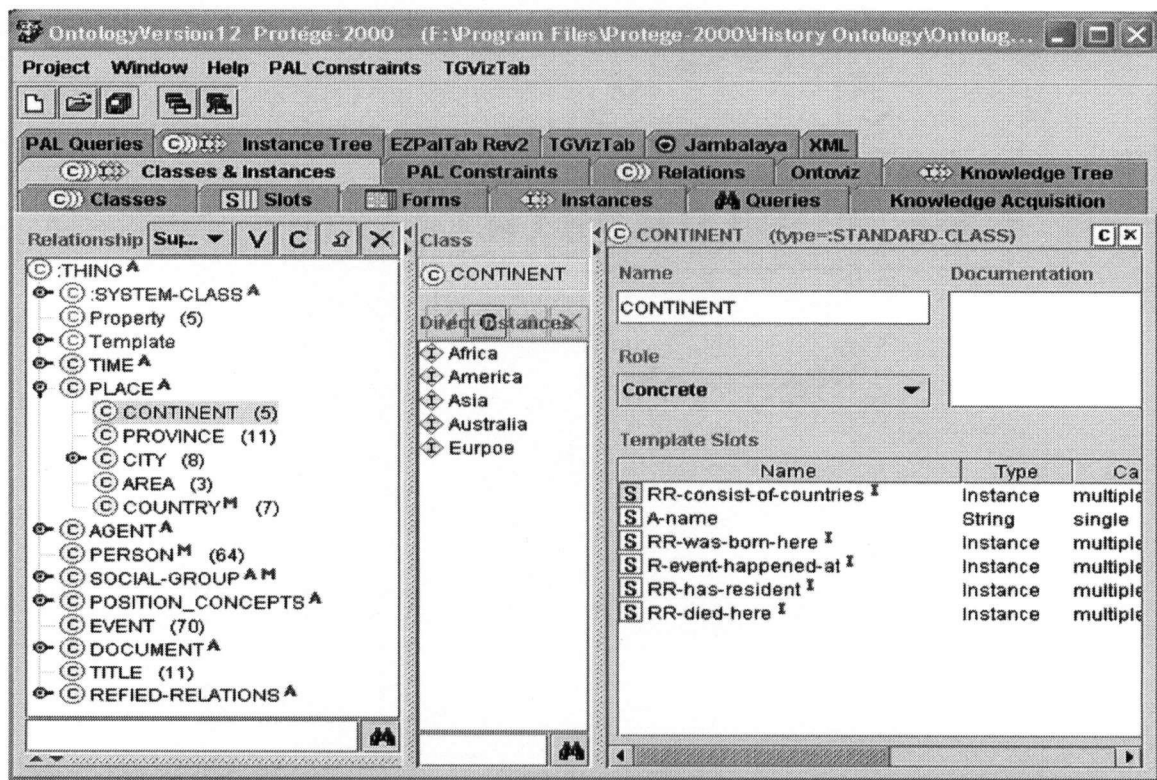


Figure 3-3 Screenshot of the Protégé 2000 development environment

An in-depth explanation for these terms is given in the following paragraphs.

- Classes

Classes represent concepts in the domain of discourse. They can have attributes and relations. In Protégé-2000 Classes are arranged in a taxonomic hierarchy. This tool represents the sub-class relation in a tree and also supports multiple inheritances. The root of a class hierarchy is the built-in class called “THING” (Figure 3-3).

- Slots

In Protégé, slots describe properties of classes and instances. They can also be used to build relations and associations between these classes and instances. Slots can be defined independently of any class; this means that they can exist without being assigned to a class. For example a slot called “name” can be defined and later attached to several different classes. Figure 3-4 illustrates an example of a slot which attached to more than one class.

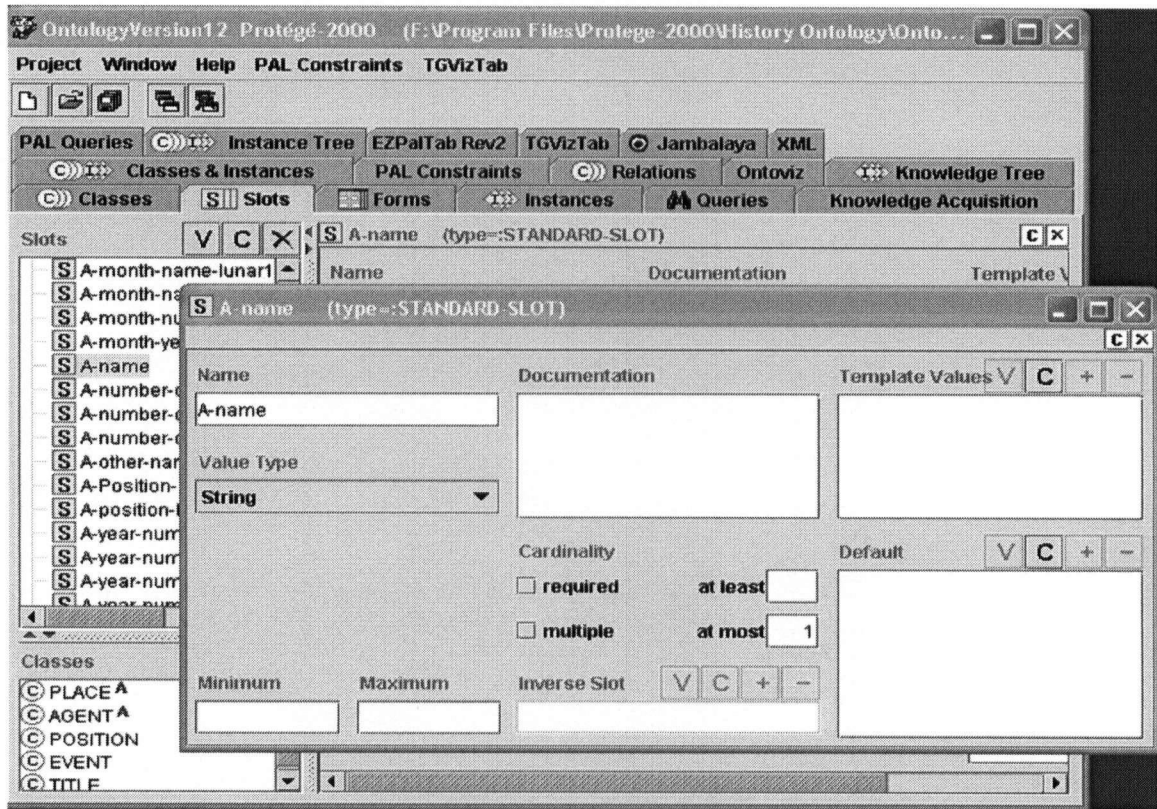


Figure 3-4 Slots in Protégé-2000

In this example, the slot “A-name” (in the left of the screen) is assigned to 5 classes (bottom left). The foreground window presents the information that can be defined for the slot.

- Facets

Facets specify constraints on slot values such as slot's cardinality (number of values that slot can have), slot value type (e.g. integer, string, float, instance of a class), maximum and minimum values for numeric slots, etc. For example, a slot "a-year-number" can be assigned 2 different maximum values depending on the class it is attached to. In Figure 3-5, the maximum value of the slot "a-year-number" can be either 2003 or 1382 depending on the chosen calendar (class it is attached to).

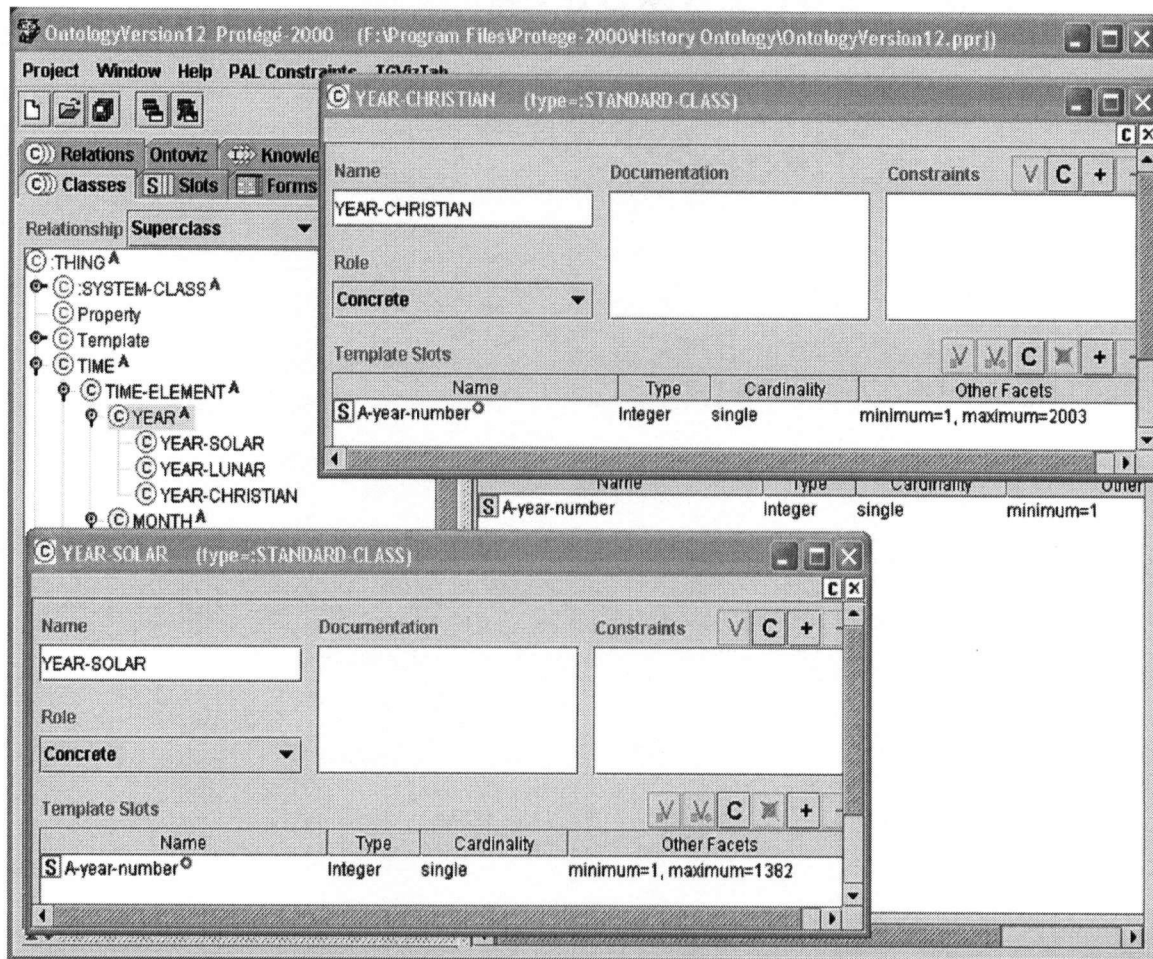


Figure 3-5 Facets in Protégé-2000

The facet maximum value applied to two classes for the slot "a-year-number".

- Axioms

Protégé-2000 does not allow for user defined axioms and rules. Constraints in this environment are limited to those provided by facets. In other words, Axioms and rules

cannot be explicitly represented in Protégé [PAL Documentation]. This could be an issue since, while creating the ontology it is necessary to be able to make general assertions about the fundamental concepts and be able to later test out and ensure that the assertions hold across the entire knowledge-base.

For example, consider an ontology in the history domain. This ontology may consist of concepts like Person, Place, Event and it will inherently involve dates and times. It is reasonable that we be able to assert the following, common-sense constraints:

- All instances of Person have exactly one *birth-date*.
- A Person's *birth-date* must precede the *death-date*.
- Every Event in which a person is involved, must take place between his/her *birth-date* and *death-date*.

These constraints ensure a certain level of ontology consistency [PAL Design Rational Document]. The primary purpose of the Protégé Axiom Language (PAL) is precisely to support the definition of such arbitrary logical constraints on the frames of a knowledge base. PAL constraints are built with special-purpose frames and can be stored as part of a knowledge base. The PAL constraint-checking engine can be run against the knowledge base to detect frames that violate those constraints [PAL].

In PAL, a constraint consists of a set of variable range definitions and a logical statement that holds on those variables (Figure 3-6). The language used by PAL is a limited predicate logic extension of Protégé-2000. This language supports the definition of the aforementioned ranges and statements. The syntax of PAL is a variant of the Knowledge Interchange Format (KIF) [PAL Documentation]. KIF is an interchange format, designed to be easy for computers to parse.

The PAL implementation of the constraint: "All instances of a Person have exactly one birth-date", is presented in Figure 3-6.

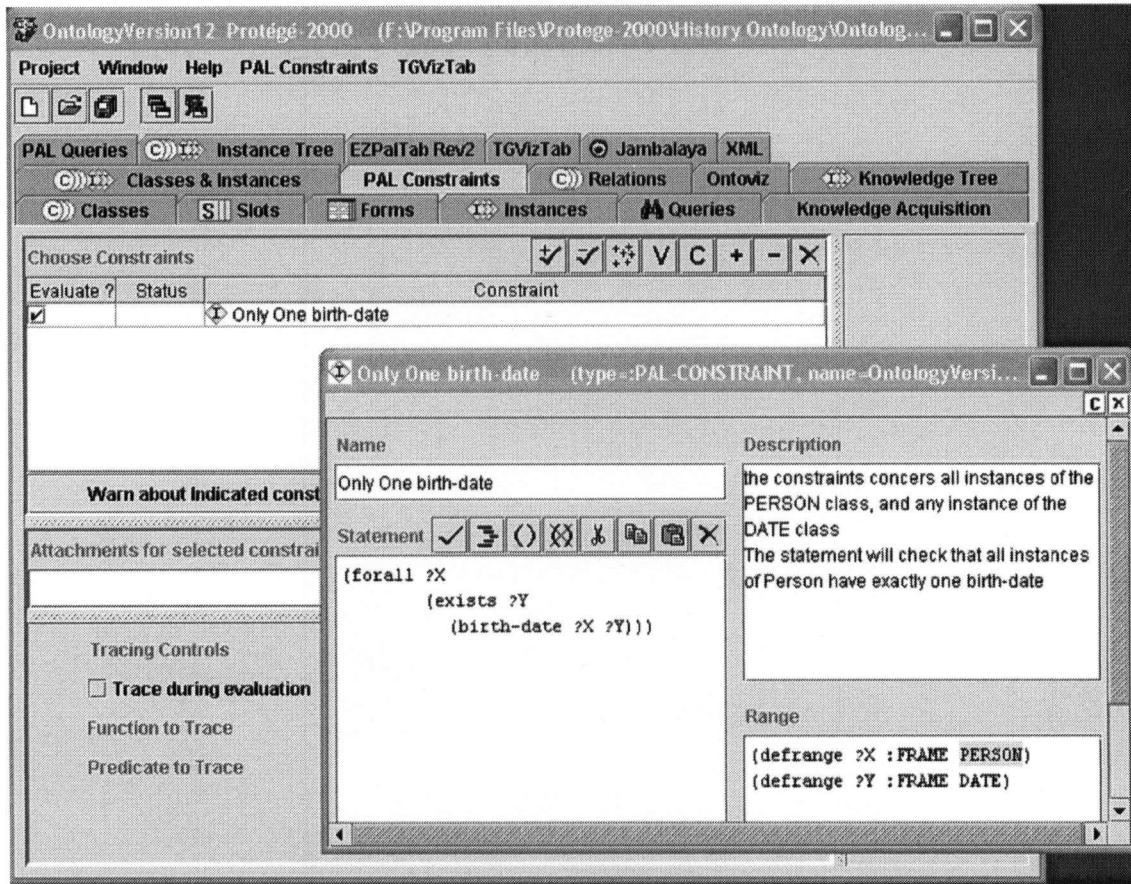


Figure 3-6 Defining axioms in Protégé-2000

In this figure, we define two variables: ?X and ?Y. ?X ranges over instances of Person. ?Y ranges over instances of Date. We then assert that associated to every ?X, there is a ?Y.

- Knowledge-acquisition forms

Protégé-2000 uses knowledge-acquisition forms to attain information about instances. Protégé relies on a form-based interface as the central user-interface metaphor [Grosso, Eriksson et al. 1999]. A user can define a class and attach template slots to it. Protégé will then automatically generate a form to acquire instances of that class. The slots for this class, their cardinality and value type will determine the default layout and content of the generated form. Users can later customize this automatically generated form for each class to better suit the requirements of the specific class. Figure 3-7 shows an example of these forms.

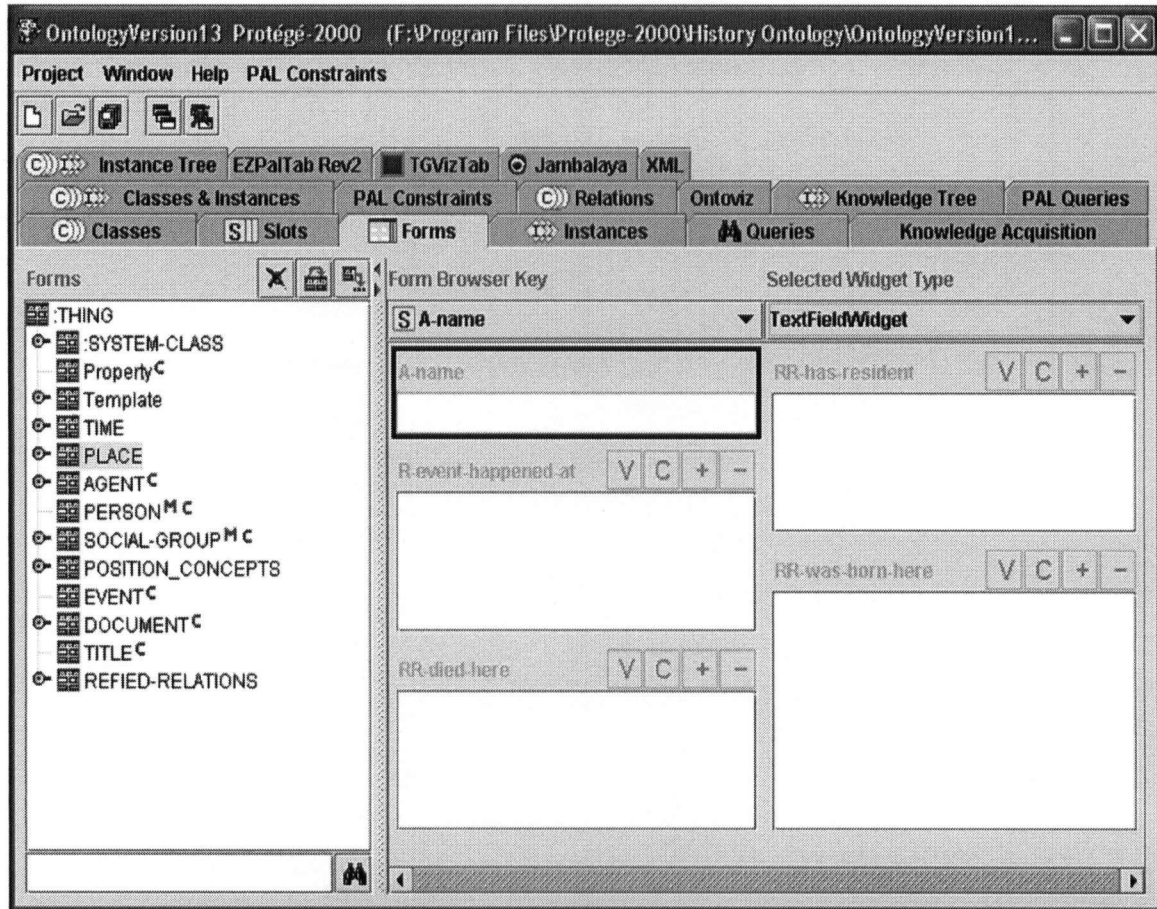


Figure 3-7 A knowledge acquisition form in Protégé-2000

This figure presents the generated form for the class “PLACE”.

- Meta-classes

Protégé-2000 supports a flexible meta-class architecture which uses configurable templates to define new classes in the knowledge base. The use of meta-classes allows for extensibility and enables interoperability with other knowledge models [Noy, Fergerson et al. 2000].

A meta-class is a class whose instances are themselves classes. A meta-class is a template for classes that are its instances, it describes how a class that instantiates the template will look. In Protégé-2000, every class is both a subclass of a class in the class hierarchy, its super-class, and it is an instance of another class, its meta-class. By default, Protégé classes are an instance of the `:STANDARD-CLASS` meta-class (Figure 3-8).

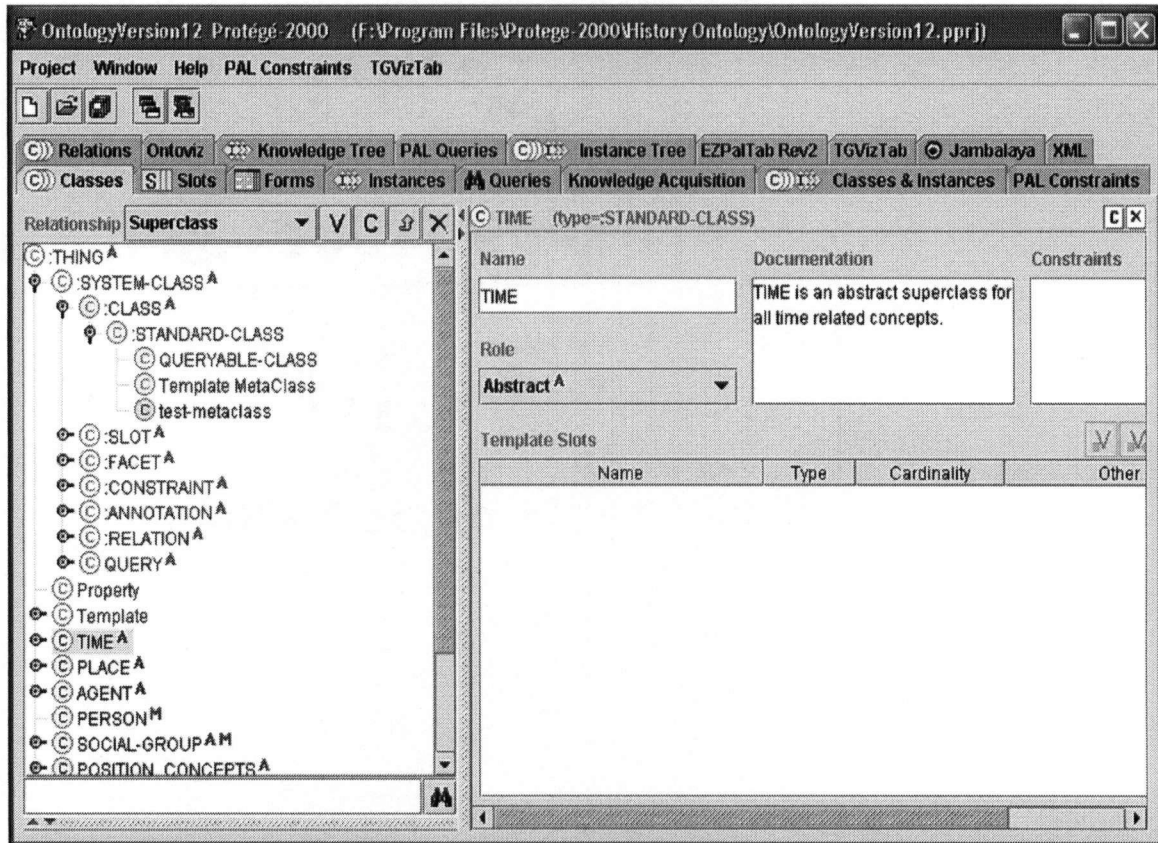


Figure 3-8 Meta Class “:STANDARD CLASS” in Protégé-2000
The class TIME is a sub-class and instance of STANDARD CLASS.

Protégé-2000 allows users to define their own meta-classes and to define new classes as instances of these user-defined meta-classes. Users can later customize the forms to acquire instances of these meta-classes, which are new classes in the ontology, effectively creating new ontology editors [Noy, Fergerson et al. 2000]. Figure 3-9 shows a user customized class form to acquire the name of the author of the class in addition to the included information from the STANDARD CLASS.

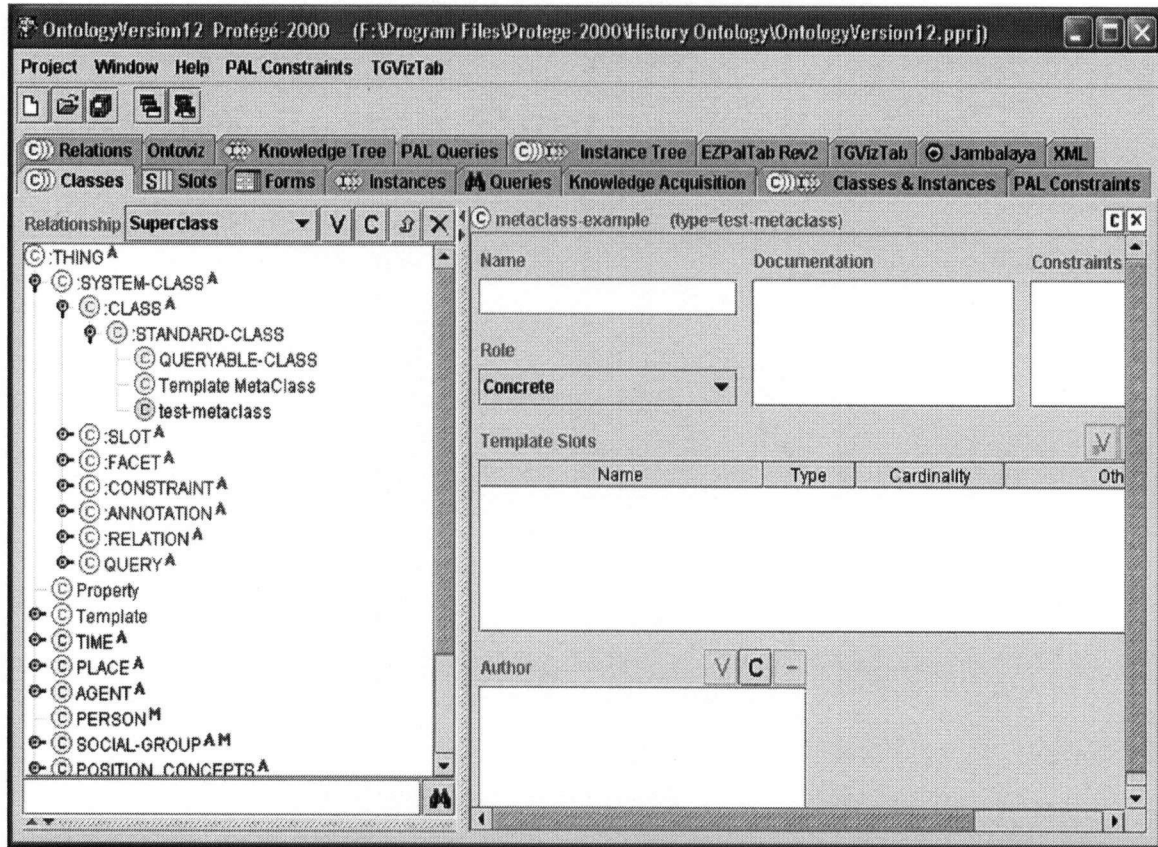


Figure 3-9 An example of user-defined meta-class in Protégé-2000

The meta-class architecture in Protégé-2000 allows developers to adapt the tool to create and edit knowledge bases with knowledge models that are different from the Protégé-2000 knowledge model. An example of this is the adaptation of Protégé-2000 to become an editor for Resource Description Framework (RDF) Schema and RDF instance data [Protege-2000/RDF 2001]. Protégé-2000 can translate an RDF knowledge base created in Protégé-2000 into standard RDF syntax; effectively making Protégé-2000 an editor for RDF documents [Noy, Ferguson et al. 2000; Knublauch 2003]. RDF is a knowledge representation standard being defined by the World-Wide Web Consortium [W3C/RDF 2000]. The main purpose behind developing RDF is to make information available in the Web not only human-readable but also machine-readable.

Protégé-2000 also complies with the new OWL (Web Ontology Language) knowledge model, which is developed by the World-Wide Web consortium and is emerging as the standard for defining metadata for encoding machine-readable semantics on the Web [Protege-2000/OWL 2003].

3.1.3 Ontology Evaluation

In order to evaluate the correctness and completeness of the created ontology, we use the query and visualization facilities provided by Protégé-2000 along with the set of motivating scenarios and competency questions which were developed in the domain analysis stage.

In order to do so, we first instantiate the ontology and examine whether the ontology provides support for the scenarios and is able to answer each and every competency question. We use the built-in query engine in Protégé for the simple query searches and use the PAL query plug-in to create more sophisticated searches. We also use protégé-visualization plug-ins to brows the ontology and ensure its consistency.

4 Build a history ontology

In order to build an ontology for a historical document, we chose to follow a series of stages as described in the previous chapter. In this chapter, we will describe the procedure undertaken to build the ontology and provide the results obtained for each step carried out throughout the ontology development process. These steps are exactly those outlined in chapter 3 and include: *identify the purpose and intended users, domain analysis and knowledge acquisition, building a conceptual ontology, building a formal ontology, and ontology evaluation*. A brief explanation and the results obtained for each of these stages of our ontology building process are presented in the following sections. Since our chosen methodology is based on an iterative-evolving process only the final results for each stage is presented. Every step of the process is explained briefly since it has already been described in detail in chapter 3.

4.1 Identify the purpose, scope, and intended users

The purpose of this stage is to identify the purpose, scope, and intended users for the ontology that we are about to build. In order to do this we provided answers to each of the following questions:

Question 1: For what purpose is the ontology being developed?

Considering the conventional ontology development purposes stated in the previous chapter we designed this ontology to facilitate the following:

- Allow communication between people who have an interest in historical documents. By this, we mean that people within a community should be able to use the same terminology to refer to the concepts with the same meaning within the domain of discourse. Furthermore users of this model should have access to the semantic of a document to refer to these concepts.
- Enable interoperability amongst systems. By using the same knowledge model computer systems will be able to use this ontology and apply their own applications on it.

- Ensure reusability. The ontology is designed so it can be shared amongst people and systems and also designed in a way that facilitates later integration into other ontologies and thus allow its reuse.
- Facilitate knowledge acquisition. This is one of the main goals our work. Knowledge found in documents can be hard to extract. Much of the meaning behind passages and sections of documents can be hidden amongst tons of words. One of our goals is to represent the semantic within a historical document. In other words, we mean to identify and represent the main concepts in this document along with their relations. Another goal of this work is to represent the temporal aspect of the knowledge. This refers to capturing the dynamicity of the stated relations within a time-line.

Question 2: What is the domain that the ontology will cover?

In order to test our ideas, we selected a history book as our test base. The book, “History of Iranian Constitution”, is written by “Ahmad Kasravi” and printed in Farsi. It talks about the events that led to the Iranian constitution. The book talks about the people who were involved in these events, the places where they happened, and their consequences. It also discusses the evolution of the governmental hierarchy and how people in this hierarchy have taken over each other’s positions throughout time.

In short, the domain that this ontology aims to represent is that of a historical document, specifically a short range of the history of the Iranian constitution that spans for about 50 years.

Question 3: What type of questions must the information in the ontology answer?

In order to answer this question we devised a set of competency questions and motivating scenarios (section 3.1.1). Table 4-1 shows a list of the competency questions that we kept in mind when we started this project. This list was later modified to that shown in Table 4-3 to Table 4-6.

List of Competency Questions (first draft)	
1.	Who is <i>person P</i> ?
2.	In what <i>events</i> has <i>person P</i> been involved?
3.	What was <i>person P</i> 's <i>role</i> in <i>event E</i> ?
4.	What is the <i>relation</i> between <i>person P1</i> and <i>person P2</i> ?
5.	What is the <i>relation</i> between <i>person P1</i> and <i>Group G</i> ?
6.	At <i>time T1</i> who is <i>person P1</i> related to?
7.	What <i>statements</i> did <i>person P1</i> make?
8.	To which <i>event</i> <i>person P</i> 's <i>statement S</i> was related to?
9.	What <i>positions</i> has <i>person P</i> held?
10.	When did <i>person P</i> hold these <i>positions</i> ?
11.	Who was taking over <i>person P</i> 's <i>position PO</i> ?
12.	Who were <i>person P</i> 's superior and/or inferior?
13.	What was <i>event E</i> about?
14.	Who was involved in <i>event E</i> ?
15.	Where did <i>event E</i> take place?
16.	When did <i>event E</i> take place?
17.	What were the <i>consequences</i> of <i>event E</i> ?
18.	Is there <i>any relation</i> between <i>events E1</i> and <i>E2</i> ?
19.	What <i>events</i> happened at <i>location L</i> ?
20.	What is the geographical hierarchy of <i>location L</i> at <i>time T</i> ?
21.	Who was from <i>location L</i> ?

Table 4-1 First draft of the competency questions

These are the questions that the ontology must be able to provide answers for.

Some of the motivating scenarios contemplated whilst developing this ontology are presented in Table 4-2.

Motivating Scenarios
A person doing a <i>biography</i> of any person represented in the book will be interested in this person's date of birth, place of birth, events in which he/she participated, as well as the different roles played within the different organizations that he/she belonged to.
For the <i>general public</i> a user might be interested in a person's ranking in the government or social status in any given timeframe and how this ranking or titles might have changed throughout history. These users might want to find out about a particular person and this person's involvement in a particular event.
An <i>archeologist</i> might be interested in a particular place or location; it would be useful for him to know which events happened in this place and who lived there through out time. Another aspect which might be of importance to an archeologist is the geographical hierarchy of places throughout time.
In the case of a <i>historian</i> , it might be of interest to know about a particular person P who held position PO in the government. The user might be interested in people whom this person was involved with, in particular those directly above or below in the hierarchical government structure. This user might also be interested in getting an overview of the hierarchical structure of the organization during the time that person P was involved in it and how this hierarchical structure changed over time. A historian might also be interested in the hierarchy of the royal family: who was the king, who were his sons, when any of these took power. Yet another aspect of interest might be who took over the different social positions, and during which time or event.

Table 4-2 Motivating Scenarios

Some of the scenarios that the developed ontology must provide support for

Question 4: Who will use the ontology? (Identify and characterize the range of intended users)

Since this ontology is intended as a proof of concept and not necessarily meant to be used in practice, the selected intended audience is that of the general public who might be interested in reading the book but want to obtain more information than that which is attainable by using conventional keyword search. Other people who might benefit from this ontology are researchers or historians interested in the semantic behind the words in the book.

Question five: What is the level of formality required for the ontology?

Our approach utilizes both an informal and a formal representation at different stages of the development process. First, we develop an informal model (conceptual ontology), which is later instantiated and transformed into a formal ontology by utilizing the Protégé-2000 development environment (section 3.1.2.4.1).

4.2 Ontology building

In this stage, we first analyzed the domain and performed several knowledge acquisition methods (as described in 3.1.2.1) to extract the knowledge required to build the ontology. We then used this knowledge to build our conceptual ontology which was later formalized, instantiated, and evaluated using Protégé-2000.

4.2.1 Domain analysis and knowledge acquisition

In this stage, we first expanded our competency questions to cover all the aspects of the information that must be included in our ontology.

Table 4-3 to Table 4-6 show our extended version of the competency questions. This list of competency questions is organized by the general or top-level notion that they refer to. These top-level notions were defined later in the development process however they are used here for clarity purposes.

List of Competency Questions (PLACE)	
1.	<u>What type of place</u> is place P? (<i>continent, country, province, city, area</i>)
2.	Where is <i>place</i> P <u>located</u> ? (in which <i>continent, country, country, province, city...</i>)
3.	<i>Who</i> <u>was born in</u> <i>place</i> P?
4.	What <i>events</i> <u>happened at</u> <i>place</i> P?
5.	If <i>place</i> P is a <i>country</i> , which are the neighbors?
6.	What <u>changes have occurred</u> to <i>place</i> P (its name, ownership, changes in geographical hierarchy?!!!!)

Table 4-3 Competency questions related to the notion "PLACE"

List of Competency Questions (EVENT)	
1.	What was <i>event</i> E about?
2.	<u>When/Where</u> did <i>event</i> E <u>take place</u> ?
3.	Who (which <i>countries</i>) <u>was/were involved</u> in <i>event</i> E/ and in what <u>role</u> ?
4.	What were the <u>consequences</u> of <i>event</i> E?
5.	What were the <u>precedents</u> of <i>event</i> E?
6.	What other <i>events</i> is <i>event</i> E <u>related to</u> ? (what are the relations?)
7.	What is the <u>relation</u> between <i>event</i> E1 and E2?
8.	What other <i>events</i> <u>took place</u> at the same <i>time</i> as <i>event</i> E?
9.	Is <i>event</i> E a <u>composite</u> of other events? If so what are those events?
10.	Who <u>holds</u> <i>position</i> PO during <i>event</i> E?
11.	What <i>statements</i> were <u>published</u> <u>related to</u> <i>event</i> E?
12.	<i>Who</i> was in opposition to / in favor of <i>event</i> E?

Table 4-4 Competency questions related to the notion "EVENT"

List of Competency Questions (PERSON)	
1.	Who is <i>person</i> P?
2.	<u>Where/when</u> was <i>person</i> P <u>born</u> ?
3.	<u>Where/when</u> did <i>person</i> P <u>die</u> ?
4.	What <i>title-of-honor</i> did <i>person</i> P <u>have</u> ? and for <i>how long</i> ?
5.	What <i>positions</i> did <i>person</i> P <u>hold</u> ?
6.	<i>Who</i> else lived in the <i>same era</i> as <i>person</i> P?
7.	What was the <u>relation</u> between <i>person</i> P1 with <i>Person</i> P2 or <i>Group of people</i> G? <u>How long</u> did the relation <u>last</u> ?
8.	<i>Who</i> <u>belongs</u> to <i>group</i> G?
9.	What type of group is group G? (social group? Religious group? Political group?)
10.	For <i>How long</i> did <i>person</i> P <u>hold position</u> PO? Start-date? End-date? Duration?
11.	<i>Who</i> else <u>had</u> the same <i>position</i> PO as <i>person</i> P?
12.	Who are the inferiors and superiors of <i>person</i> P in the hierarchy?
13.	When was <i>person</i> P <u>dismissed</u> from <i>position</i> PO? <i>Who</i> <u>took over position</u> PO?

Table 4-5 Competency questions related to the notion "PERSON"

List of Competency Questions (DOCUMENT)	
1.	What was <i>document D</i> about?
2.	<i>Who made document D?</i>
3.	When was <i>document D</i> <u>made</u> ?
4.	What type of <i>document</i> is document D?
5.	In <u>relation to</u> what <i>event(s)</i> was <i>document D</i> done?
6.	What are other <i>documents</i> with the same subject or about the same <i>event</i> as <i>document D</i> ?

Table 4-6 Competency questions related to the notion "DOCUMENT"

We used brainstorming sessions in conjunction with these competency questions and motivating scenarios to extract all potentially relevant terms for our ontology. These terms were gathered into a glossary. This glossary of terms includes the terms, their definition or description, and may include additional information such as examples that help understanding these definitions. In order to provide a definition for the terms, we consulted dictionaries such as the Merriam Webster Dictionary [Merriam] and the Oxford Dictionary [Oxford] as well as general purpose ontologies such as SUMO [SUMO], Ontolingua [Ontolingua] , and WordNet [WordNet]. We must note that this was an iterative process and as we proceeded in our development, these lists (competency questions, motivating scenarios and glossary of terms) evolved. Table 4-7 shows a partial view of our term glossary.

Glossary of Terms		
Term	Definition	Resource
Agent	"Something or someone that can act on its own and produce changes in the world."	SUMO
Person	An individual, someone, somebody. An agent with certain rights and responsibilities and the ability to reason, deliberate, make plans, etc. This is essentially the legal/ethical notion of a person	WordNet & SUMO
Group-of-People	A number of individuals assembled together or having some unifying relationship	Merriam Webster
Religious Group	A Group-of-People whose members share a set of religious beliefs	SUMO
Government	A ruling body of a country	Oxford
Position	A formal position of responsibility within an organization.	SUMO
Event	Something that happens at a given place and time and has a certain duration.	SUMO & WordNet
Place	A particular region, center of population, any geographic area which is associated with some sort of political structure. This notion includes Lands, Cities, districts of cities, counties, etc.	Merriam Webster & SUMO
Time-Point	TIME POINT is a specification of a single point in historical time. A time-point is not a measurement of time, nor is it a specification of time.	Ontolingua & SUMO

Table 4-7 Partial view of the glossary of terms

4.2.2 Building an informal ontology

Once we obtained a relatively complete list of competency questions and glossary of terms, we identified concepts, relations within the concepts, and their attributes. We did this according to the guideline presented in the previous chapter (section 3.1.2.3).

In our lists of competency questions (Table 4-3 to Table 4-6) concepts are shown in *italics* and the relations are underlined. The concepts and their related properties are stored into concept dictionaries which are structured into naturally occurring groups. This approach is a combination of the approaches introduced in [Noy and McGuinness 2001] and those of [Lopez, Perez et al. 1999]. At this stage we categorized our concepts into

five concept dictionaries relating to *people*, *places*, *events*, *documents*, and *time*. Each of these categories holds the concepts that are most related together. In Table 4-8 to Table 4-12, the relations and attributes specific to each particular concept are shown in *italics*. The rest (non-italic) are shared amongst all the concepts in the same dictionary.

Table 4-8 to Table 4-12 illustrate our concept dictionaries. Each of the tables is preceded by a brief description. Further explanation will be provided later when we demonstrate the conceptual model based on these concept dictionaries and whilst transforming the conceptual ontology into a formal ontology.

Table 4-8 shows concepts related to the notion “PLACE”. Within our domain, we identified five different types of geographical places: continent, country, province, city, and an area within a city. The relations that are represented here include those that capture geographical interdependencies amongst the different type of places, those that present a connection between places and events, and those that demonstrate the association between people and places.

Concept Dictionary for the notion "PLACE"		
Concept Name	Property	Relation
CONTINENT	has-name	event-happened-in , people-was-born-in, people-lives-in, people-died-here, <i>countries-included</i>
COUNTRY	has-name	event-happened-in, people-was-born-in, people-lives-in, people-died-here, <i>has-neighbor, consist-of-provinces,</i> <i>located-in-continent,</i>
PROVINCE	has-name	event-happened-in, people-was-born-in, people-lives-in, people-died-here, <i>consist-of-cities, located-in-country,</i>
CITY	has-name	event-happened-in, people-was-born-in, people-lives-in, people-died-here, <i>consist-of-areas, located-in-province</i>
AREA	has-name	event-happened-in, people-was-born-in, people-lives-in, people-died-here, <i>located-in-city</i>

Table 4-8 Concept dictionary related to the notion "PLACE"

Table 4-9 illustrates the concepts related to people. This includes both individual persons and groups of people. In our work, we identified three concepts that are associated with people: individual persons, groups of people, and social/governmental positions that a person may have. As for relations amongst these concepts, we included those that represent the associations between a person and a group of people, those that denote relationships amongst a place, a person, and/or a group of people, and those that are specific to each of these concepts such as a particular relation that shows the positions held by a person or the foundation date of a group. We should note that, this table does not show neither the hierarchical structure of a position concept nor the changes this hierarchy might have undergone throughout time, this will be presented later when we describe the conceptual ontology model.

Concept Dictionary for the notion "PEOPLE"		
Concept Name	Property	Relation
PERSON	name other name description <i>gender</i>	is-related-to-document, made-statement, is-addressee-of, involved-in, <i>is-father-of</i> , <i>is-son-of</i> , <i>is-father-in-law</i> , <i>is-son-in-law</i> , <i>holds-position</i> , <i>is-member-of</i> , <i>is-addressee-of-statement</i> , <i>has-birth-date</i> , <i>has-death-date</i> , <i>has-birth-place</i> , <i>has-death-place</i> , <i>has-title-of-honor</i> ,
POSITION	name description <i>position-level</i>	position-holder, has-time-interval
GROUP-OF-PEOPLE	name other name description <i>group-type</i>	is-related-to-document, made-statement, is-addressee-of, involved-in, <i>has-foundation-date</i> , <i>has-dismissing-date</i> <i>has-member</i>

Table 4-9 Concept dictionary related to the notion "PEOPLE"

Table 4-10 shows the properties and relations assigned to the concept "EVENT". The relations defined, include those that represent the association of an event with other concepts such as places, people, and documents and those that capture the interrelation amongst events themselves. We consider that events can be a precedence, consequence, part of, and /or composite of other events. We also allow representing whether any given event is related to any other event in any other way.

Concept Dictionary for the notion "EVENT"		
Concept Name	Property	Relation
EVENT	name description goal	happened-at-place, happened-at-time-point has-agent, precedence-of, consequence-of part-of, composite-of, has-related-event has-related-document, has-time-range

Table 4-10 Concept dictionary related to the notion "EVENT"

Table 4-11 shows the properties and relations associated with the concept “DOCUMENT”. The relations consist of those that represent the relations between people, events, and other documents to a document. The properties assigned to this concept aim to capture the general information about a document such as its title, a brief description, its content, and its type. Different document types in our domain include: report, letter, telegraph, order, notice, license, contract, memo, agenda, meeting minute, bill, and expense statement.

Concept Dictionary for the notion “DOCUMENT”		
Concept Name	Property	Relation
DOCUMENT	title	publication-date, made-by, made-about-event
	description	related-to-document, related-agent
	content	related-event
	document-type	

Table 4-11 Concept dictionary related to the notion “DOCUMENT”

Table 4-12 presents all the concepts related to the notion “TIME”. These concepts include time elements such as year, month, and day and other time related concepts such as calendar date, time interval, and duration. A calendar date is a time point with a resolution of days. It consists of a day, a month, and a year. A time interval captures a certain period of time which consists of a start time, an end time, and a duration. A duration represents the length of a time interval in form of number of days, months, and years.

Concept Dictionary for the notion "TIME"		
Concept Name	Property	Relation
YEAR	year-number	
MONTH	month-number month-name	
DAY	weekday-name weekday-number month-day-name	
CALENDAR DATE		has-day, has-month, has-year
TIME INTERVAL		has-start-time, has-end-time, has-duration
DURATION	number-of-days number-of-months number-of-years	

Table 4-12 Concept dictionary related to the notion "TIME"

The next step in our process was to use the identified concepts and relations (concept dictionaries) within our domain along with the motivating scenarios to develop our conceptual ontology model. This was carried out in a manner similar to what was presented in the previous chapter (section 3.1.2.3). This conceptual model illustrates not only the concept hierarchy (sub-concepts, super-concepts) but also the interrelations that hold amongst these concepts. We used a middle-out approach to categorize the concepts. The middle-out approach operates by identifying the most important concepts first and then generalizing or specializing the concepts within groups from this point. We also created abstract concepts as organizing features. These concepts are organized as groups due to the similar features that they possess, however they are not directly related as super-concepts and sub-concepts. For example, we grouped the concepts related to time under a super-concept TIME. (Figure 4-3)

The next stage in our process is to build the conceptual model of our ontology. Figure 4-1 to Figure 4-13 represent the different parts of our conceptual model. A detailed explanation is provided prior to the presentation of each of these figures. We adapt a simple Entity Relationship (ER) model to illustrate our conceptual ontology at this stage. Within these figures, a rectangular shape represents a concept, a diamond symbolizes a

relationship, and an oval shape denotes the attributes associated with these concepts or relations. These attributes can be concepts themselves such as concept “TIME” which can be allied with relations. Some of the relations denoted by a diamond contain the symbol “/”, which indicates that the relation is bidirectional. In other words, the stated relationship has an inverse. Throughout the development of our ontology we searched existing libraries of ontologies in order to find similar related concepts or relations useful to us. The individual models we adapted to our own work are cited in the corresponding section.

Figure 4-1 represents the top-level concept hierarchy in our domain. Considering the glossary of terms, competency questions, motivating scenarios, and concept dictionaries produced in the previous steps, we identified five central concepts within our ontology: AGENT, PLACE, EVENT, DOCUMENT, and TIME. Every other concept in this domain is defined around these primitive concepts.

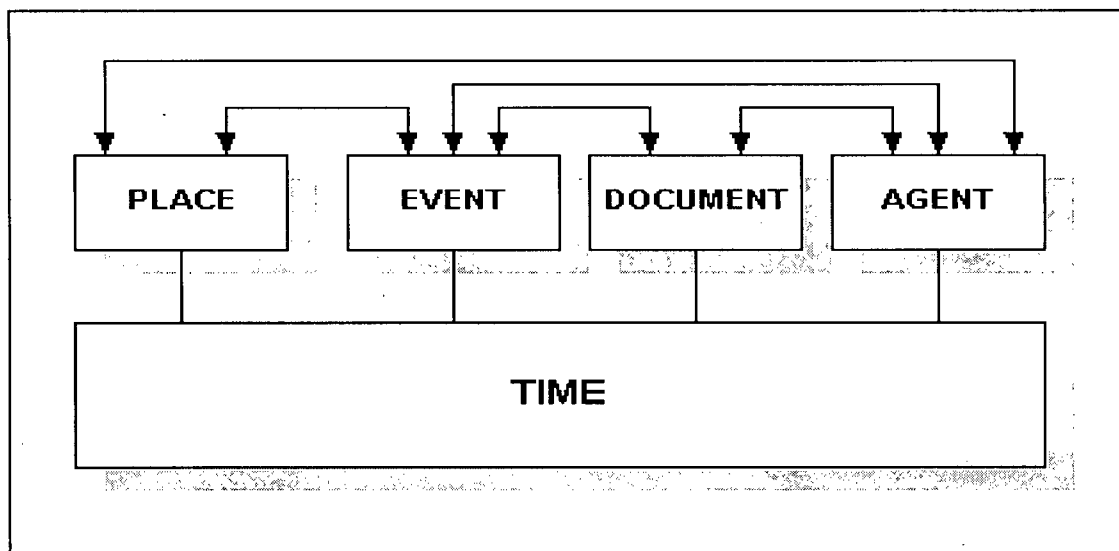


Figure 4-1 Main concepts within our ontology

Figure 4-2 roughly illustrates the relations that hold amongst the main concepts. These concepts and relations are presented in detail when we represent the conceptual model for each of these concepts. At this point, we try to give the reader an overview of how these concepts are related to one another.

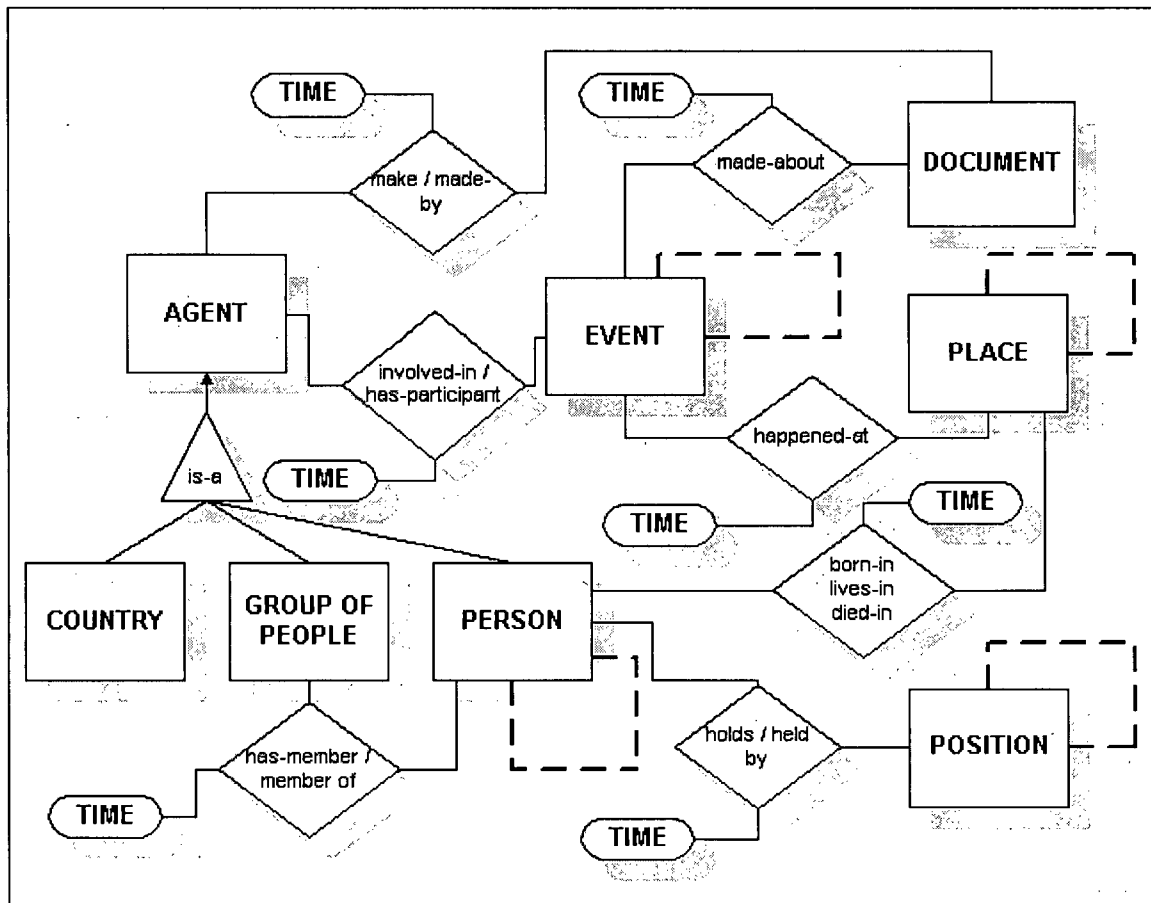


Figure 4-2 Overview of main concepts and relations in our history ontology

The dotted- lines denote the existence of internal relations within a concept. The time concept associated with the relations indicates that a particular relation is time-dependent. The “is-a” relation shows how bottom-level concepts are a type-of top-level concept.

The representation of time is fundamental to any knowledge model that aims to represent change or action. Many ontologies that try to represent time are currently available. Some examples of these include: Simple-Time Ontology from Ontolingua Sever [Ontolingua], Time Ontology from SUMO [SUMO], and the Time ontology developed in the Stanford Knowledge Systems Laboratory [Zhou and Fikes 2002]. In order to develop our time model, we studied these existing time ontologies. We based our work on these ontologies and adapted them to fit our needs. Traditionally time ontologies have a resolution of seconds or even milliseconds, however our time model has a coarser granularity and measures time by day.

Our time model is illustrated in Figure 4-3. It considers time as an infinite continuum. Any single point on this continuum is a TIME-POINT (i.e. Jan 1, 1900). If we have two TIME-POINTS we can describe a TIME-INTERVAL (i.e. from Jan 1, 1900 to Aug 1, 1900) and calculate a DURATION (seven months). Note we can have a DURATION with unknown TIME-POINTS or INTERVAL. In our representation of time “TIME POINT” and “TIME INTERVAL” are the two fundamental concepts. An abstract concept called “TIME” was created and is used as an organizing feature to gather all concepts related to time in a group.

Moving down in the concept hierarchy tree we can find CALENDAR-DATE. This concept is a time point with resolution of days. It is the main element used to capture the temporal aspects of our domain. A “CALENDAR-DATE” contains a day, month, and a year. Since the book that we were extracting the data from represents the dates in three different calendars: *Solar calendar*, *Lunar calendar*, and *Christian calendar*, at later stages we defined additional concepts that allow us to represent these three different calendars (Figure 4-4).

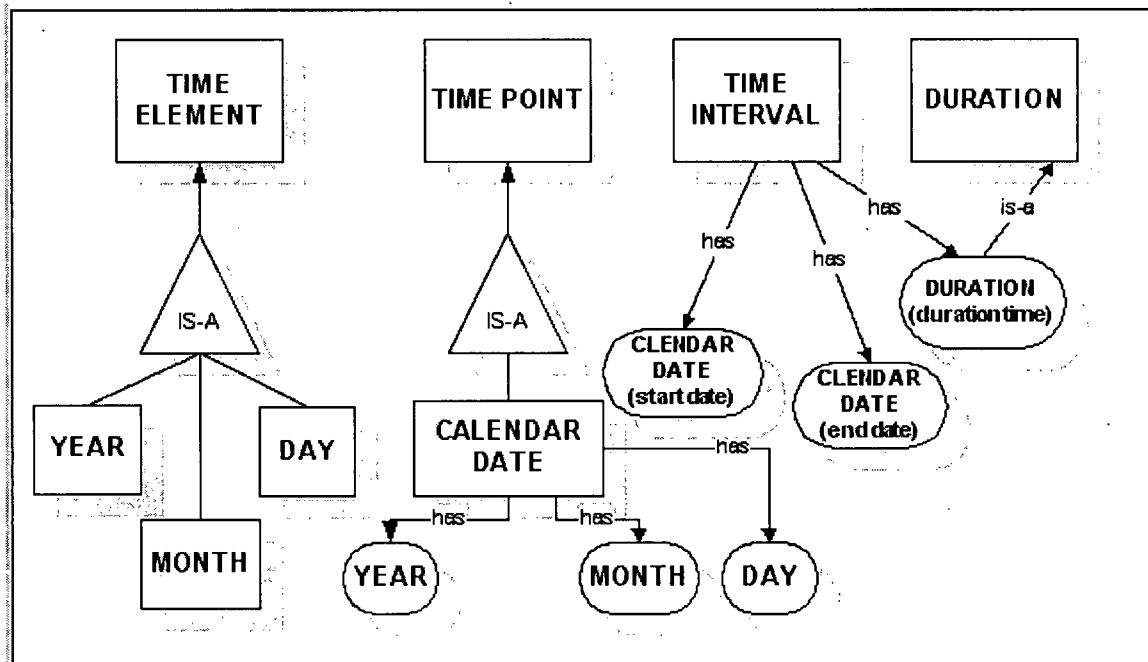


Figure 4-3 Time model in our domain

located in a continent. This consist-of relation is commonly referred to as a part-whole relation. These relations are usually employed to organize knowledge about the concepts in a domain. However; ontologies do not have this relation as a built-in functionality unlike the IS-A relation, thus it has to be defined manually [Noy and Hafner 1997].

Since our model is temporally dynamic these relations have a time property associated with them. By temporally dynamic, we mean that the relations within the concepts are time-based, they might change or evolve over time. Additionally, we might also see some new places appear and others cease to exist after a certain time or either their name or region might be changed. For example the cities in a country might change their name, location or might end up belonging to a different country at different points in time. This will be further explained as we describe the transformation of our informal model to a formal ontology.

Figure 4-6 shows the relations that hold between the concept “PLACE” and other concepts in our domain such as PEOPLE or EVENTS. It should be noted that these relations are time-dependent and thus have a time attribute associated to them.

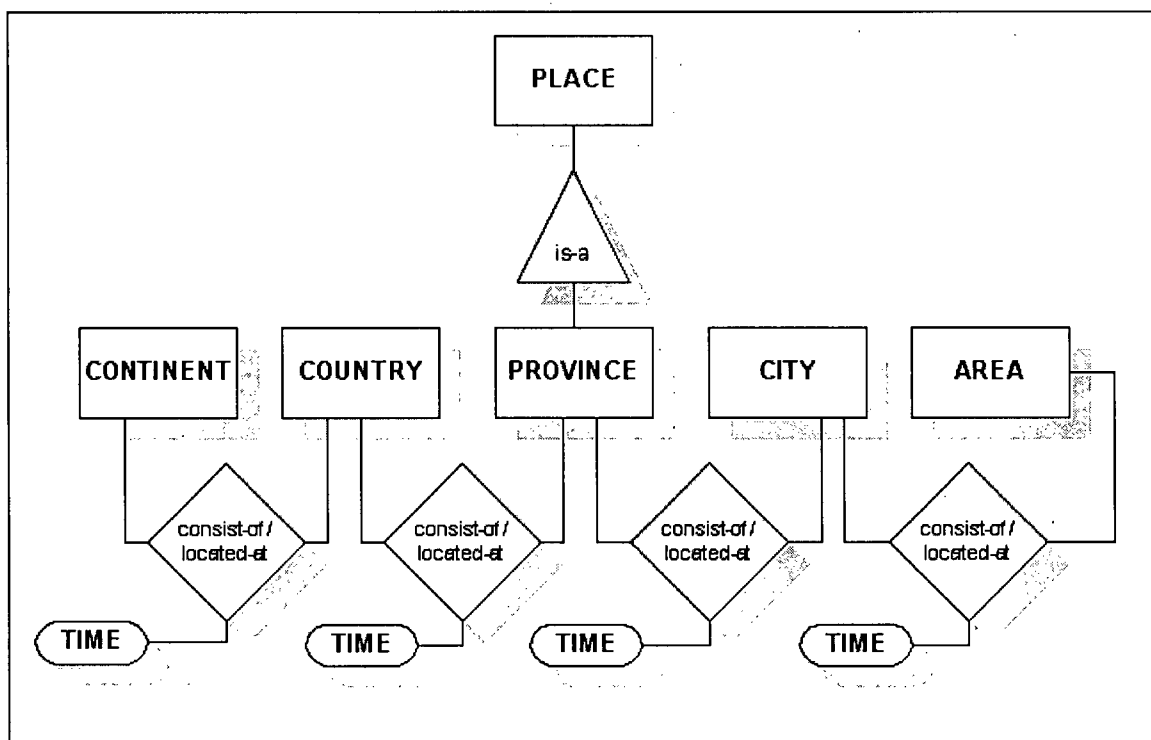


Figure 4-5 Interrelations amongst concepts related to “PLACE”

This figure represents the interrelations amongst the concepts related to place. These relations allow us to capture the hierarchy of places within our ontology.

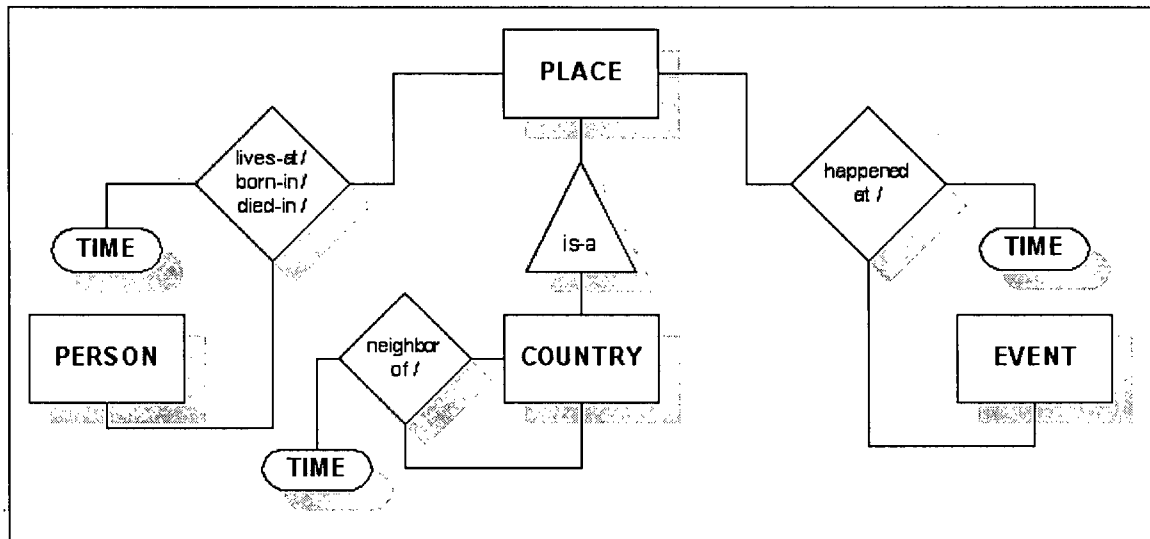


Figure 4-6 Relations between the concept “PLACE” and other concepts within our ontology

Figure 4-7 and Figure 4-8 illustrate the concepts and relations associated to people. This includes both individual persons and groups of people. Figure 4-7 shows an overview of our concept hierarchy. We defined a general concept “AGENT” referred to as: “something or someone that can act on its own and produce changes in the world”. The concept “AGENT” acts as a super-concept for both persons and groups-of-people. As it can be noticed, the relations that are shared amongst all the sub-concepts of “AGENT” are only assigned to the concept “AGENT”. These relations (i.e. a certain person can make a certain statement due to the relation that exists between “AGENT” and “DOCUMENT”) are inherited by sub-concepts of “AGENT” (“PERSON”, “GROUP OF PEOPLE”). We later added the concept “COUNTRY” as a sub-concept for “AGENT”, this was done due to the role that a government of a country might play as the representative of the people within the country in an international affair. A logical consequence of this is to consider the concept “COUNTRY” as a sub-concept of both “PLACE” and “AGENT”.

Figure 4-8 provides a more in depth view of the relationship between a “PERSON” or “GROUP OF PEOPLE” with other concepts in the domain. Two concepts; “TITLE” and “POSITION” are associated with a “PERSON”. The concept “TITLE” captures the titles of honor a person might have for any period of time. These titles are assigned to different

people throughout history. The notion POSITION indicates the positions that a person might have within an organization. This concept will be further explained in the following section.

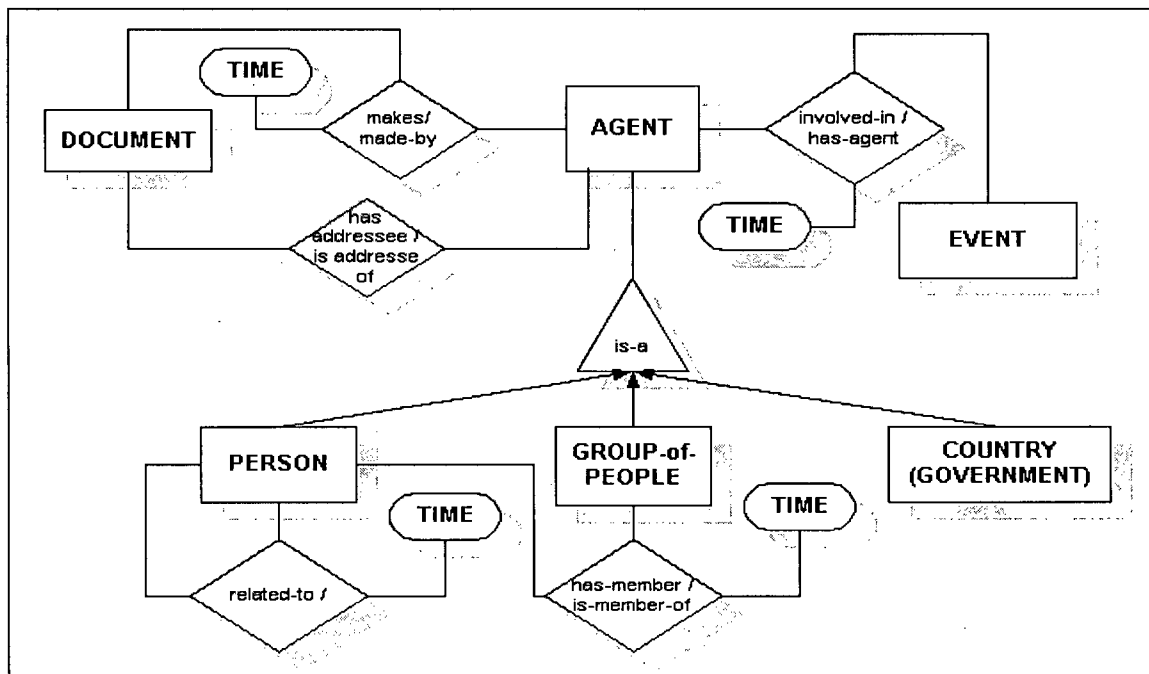


Figure 4-7 Concept hierarchy for "AGENT," "PERSON," "GROUP OF PEOPLE," and "COUNTRY"

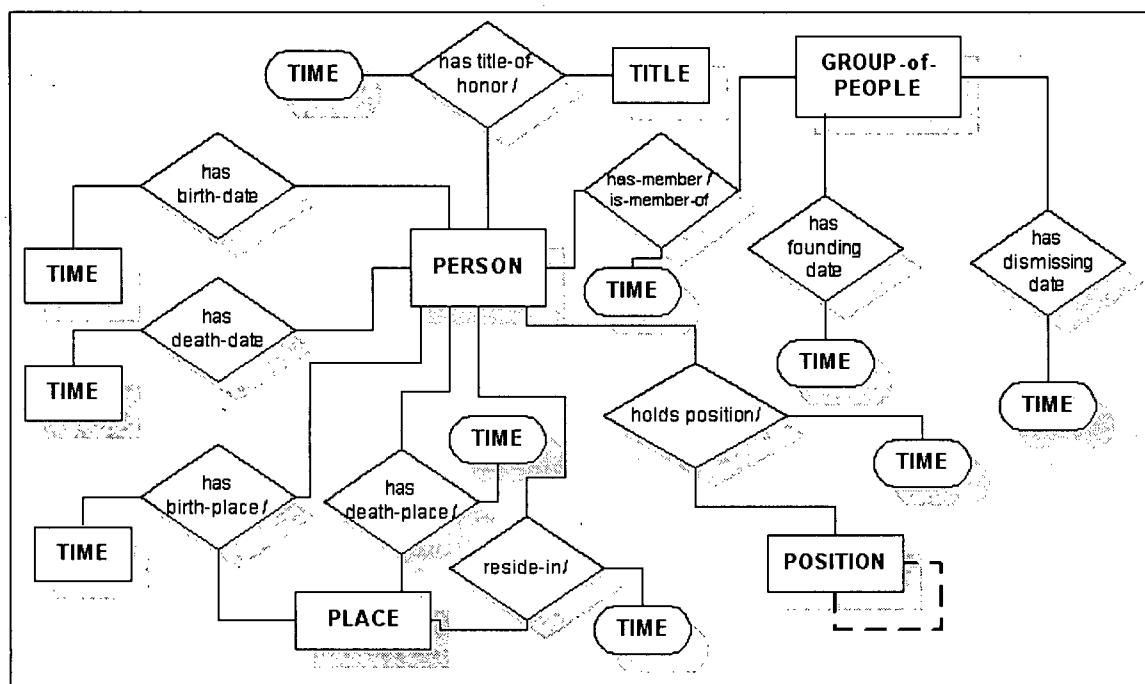


Figure 4-8 Relations between “PERSON” and “GROUP OF PEOPLE” and other concepts in our domain

Figure 4-9 and Figure 4-10 demonstrate the conceptual model for the notion “POSITION”. Within our domain a position is defined as “a formal role with responsibilities that a person can have within an organization.” We identified two types of position hierarchies within our history book: *governmental positions* such as king, prime minister, or governor, and *royal court positions* such as king and crown prince. Figure 4-9 illustrates how we defined our position hierarchy. Each position has a person assigned to it, who holds the position for a period of time. Therefore there exists a triplet of person, position, and time (or more precisely a time interval) which represents each position at any given point in history. For each position there might also exist an inferior and/or superior which itself consists of a person, position, and time. The reason to represent the positions in this manner is to facilitate building a hierarchical structure of positions that captures the dynamicity inherent to these notions. Our model not only captures how each position is filled throughout time but also how the relations between these positions evolve. Figure 4-10 illustrates our approach for capturing position hierarchies in time. In order to capture the dynamicity of the hierarchy of positions and the people that hold them, we defined a different set of positions that exist at the different

time slots. By this we mean that, as soon as there is a change in the position hierarchy (like a position has been added, one is removed, the order is changed) we use a new set of positions which are applicable at the new time range. In this set, the positions are labeled according to their rank in the hierarchy. We then build our position hierarchy based on this set and associate the triplet of person, position, and time to this new hierarchy. This will further explained while we constructing our formal ontology.

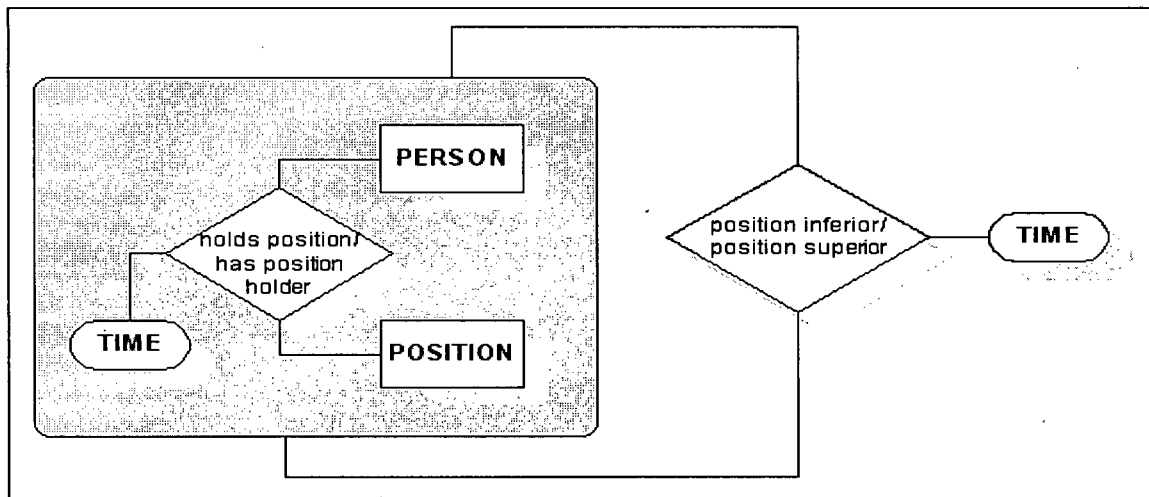


Figure 4-9 Time-based hierarchical relationship between a PERSON and POSITION

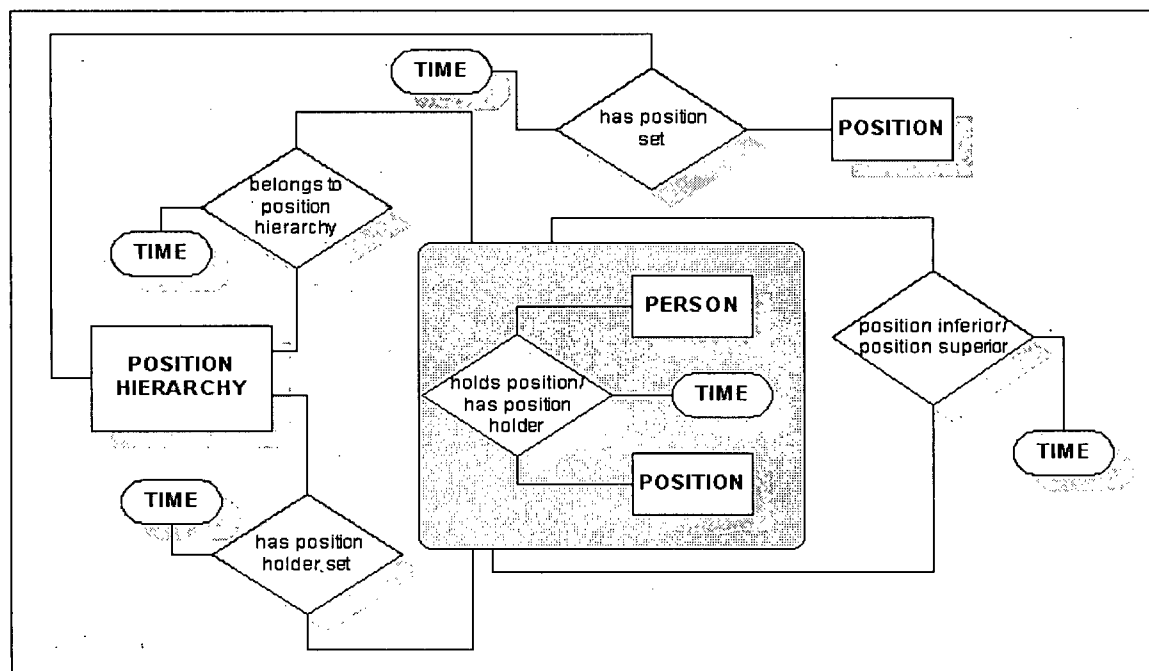


Figure 4-10 Conceptual model for position hierarchy

Our representation deals with EVENTS using a set of relations adapted from John Sowa's work on thematic roles or case relations [Sowa 2000a]. Figure 4-11 and Figure 4-12 demonstrate our "EVENT" conceptual model. The relationships defined for the concept "EVENT" allow us to capture the who, what, when, where, and why (goal) of an "EVENT" (Figure 4-12). This model also captures the relationships between different events. Additionally, the time concept assigned to these relations allows us to model chronological, sequential, series, and parallel events (Figure 4-11).

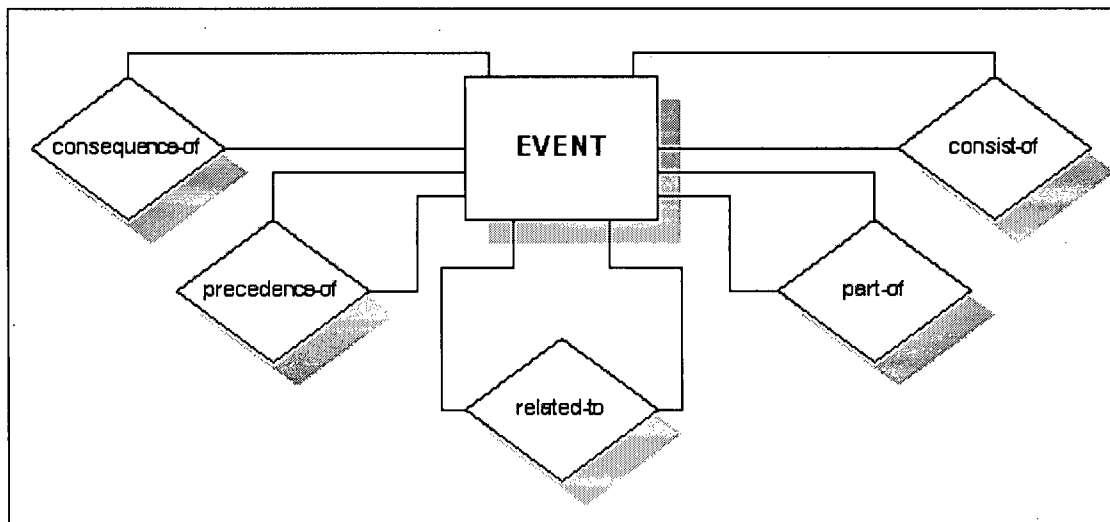


Figure 4-11 Interrelations amongst events

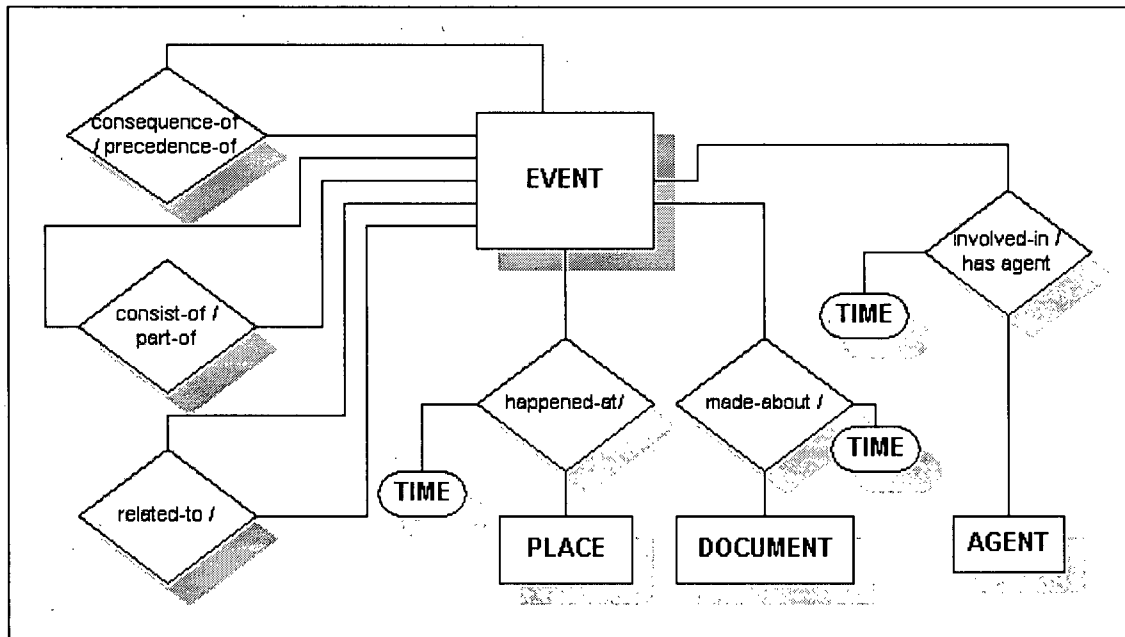


Figure 4-12 Association of an “EVENT” with other concepts in our domain

The conceptual model of the “DOCUMENT” concept is illustrated in Figure 4-13. This model represents the relations that may exist between a document and other concepts in our domain. All these relationships have an inverse (i.e. made-by/makes).

Later in the development process we defined two sub-concepts of “DOCUMENT”, those referring to documents that have addressees (MUTUAL DOCUMENTS) and those without addressees (NON-MUTUAL DOCUMENTS).

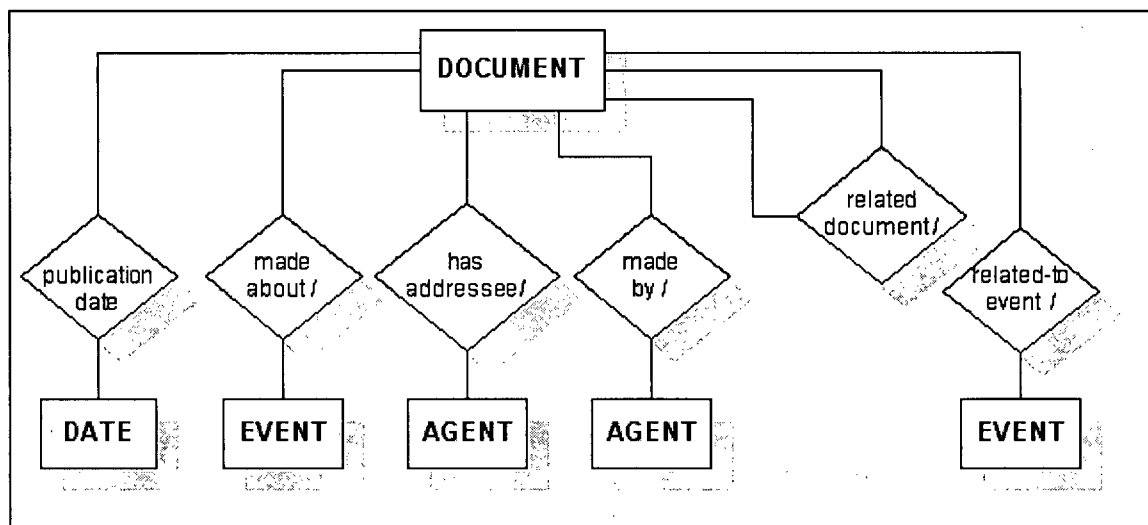


Figure 4-13 Concept “DOCUMENT” and its relation with other concepts in the domain

4.2.3 Building a formal ontology model

Once we have created our conceptual model, we utilized Protégé-2000 to build our formal ontology. The first step in this formalization is to map the created conceptual model to the model provided by Protégé-2000.

Table 4-13 illustrates the mapping between the notions we used to describe an ontology and the representation of these notions in Protégé-2000. An in depth explanation of these notions can be found in the previous chapter (section 3.1.2.4).

Our terminology	Representation in Protégé-2000
Concept	Class
Relation	Slot
Property / attribute	Slot
Constraint / Axiom	Facet
	Axiom
Instance	Instance

Table 4-13 Mapping between the terminology we used and Protégé-2000 terminology

The rest of this section describes each of the steps of the procedure carried out to build our formal ontology.

4.2.3.1 Mapping our domain concepts to a class hierarchy

At this stage we transform our conceptual model to the class hierarchy in Protégé-2000. The class hierarchies are illustrated in Figure 4-14.

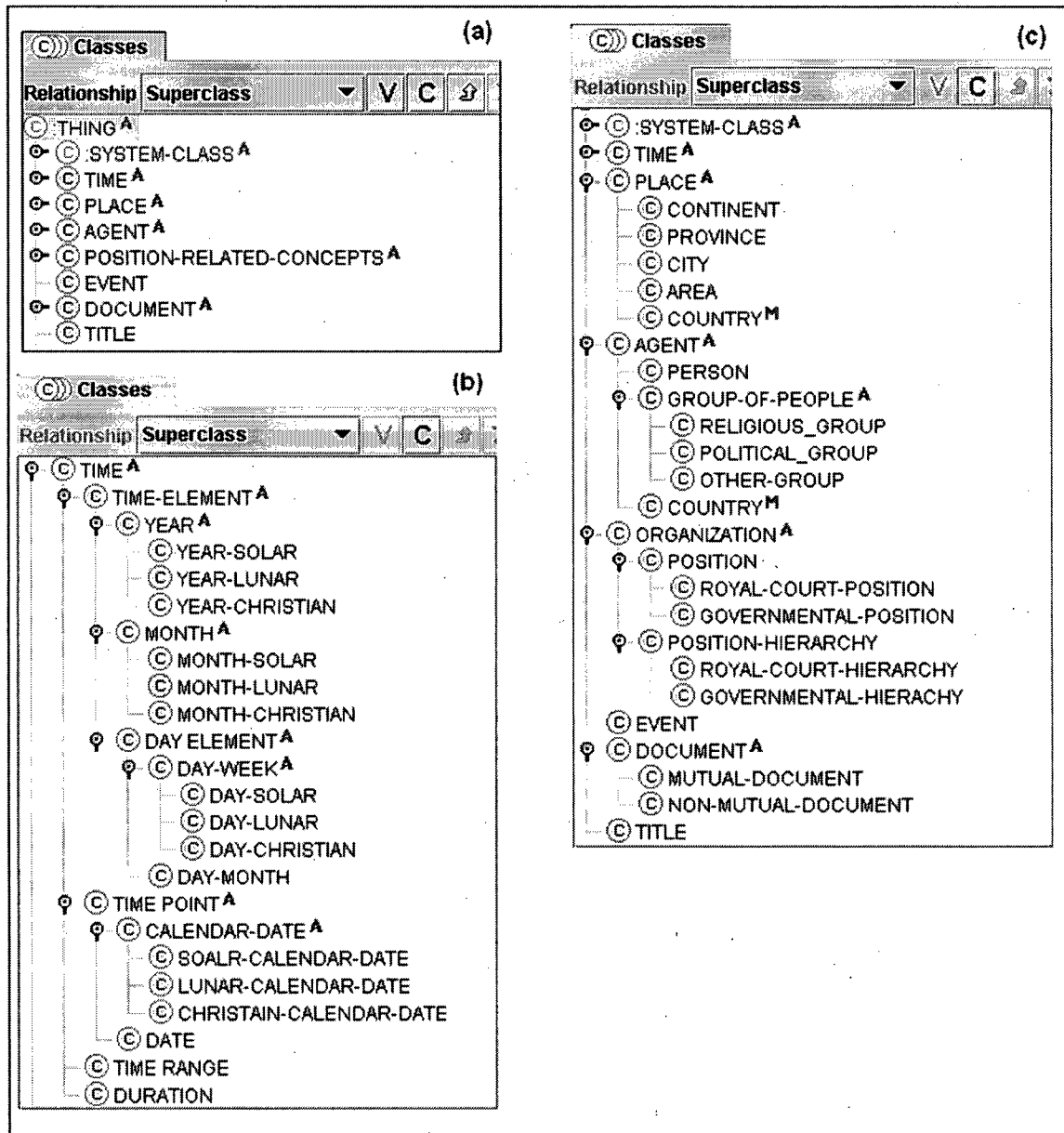


Figure 4-14 Our ontology class hierarchy in Protégé-2000

Within this figure the top left section (a) represents our top-level hierarchy which illustrates only the main concepts in our domain. The bottom left section of the figure (b) shows the TIME hierarchy in our ontology. Finally, the right part of the figure (c) demonstrates the break down of the hierarchy of concepts within our domain.

4.2.3.2 Assigning properties to classes (attributes & relations)

Once we have our class hierarchy, we must assign properties to these concepts. In Protégé-2000, these properties become slots attached to the classes. Slots can represent both attributes and relations associated to a class. All subclasses of a class inherit the slots of that class. For example, all the slots of the class PLACE are inherited by subclasses of PLACE. (Figure 4-15)

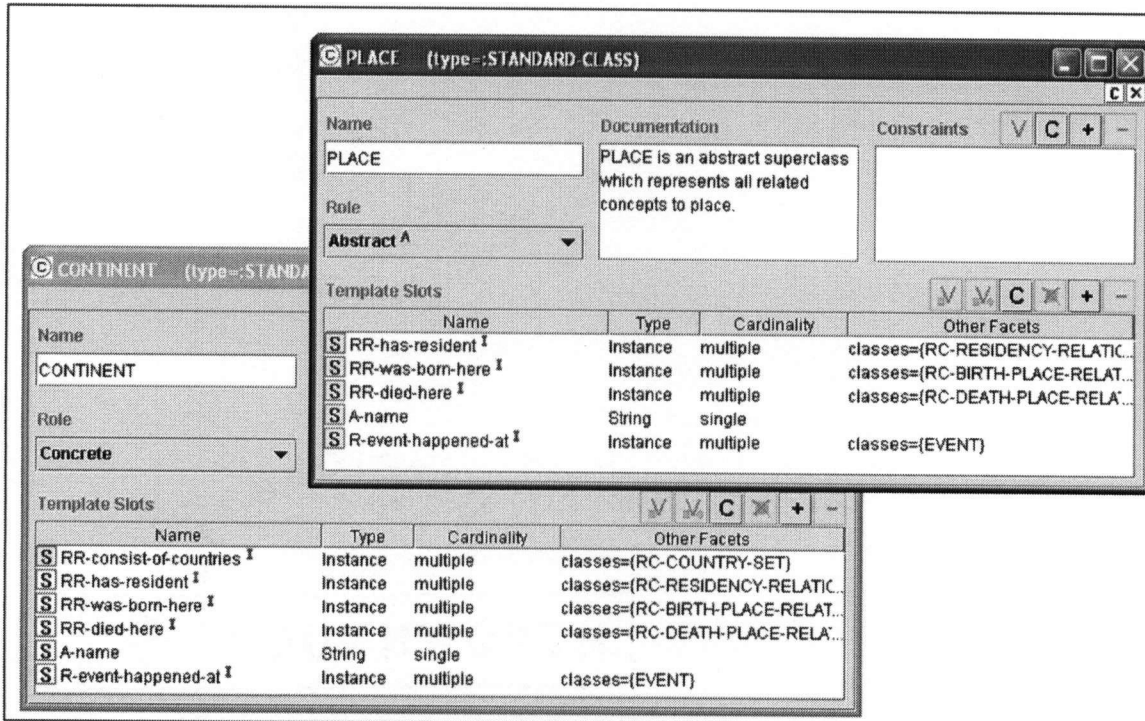


Figure 4-15 Inheritance of properties and relations by subclasses

This figure shows the super-class “PLACE” and its subclass “CONTINENT”. “CONTINENT” inherits all the slots attached to “PLACE”. An attribute “name”, defined as a type string, and a relation “happened-at” relates the concepts “PLACE” and “EVENT” for every sub-class of “PLACE”.

At this stage we used our list of attributes and relations defined in our concept dictionaries in conjunction with the conceptual model derived from the previous stage and attached the appropriate slots to the classes. This includes assigning attributes such as a string or a number value to slots and associating a class to other classes in the ontology by defining slots that represent relations. The remainder of this section explains how we defined relationships between the concepts in order to capture the dynamic temporal aspect of the history domain that we intend to represent.

The knowledge model of Protégé-2000 conforms to that of OKBC [OKBC 1998], where slots attached to classes describe relations via a combination of binary concept-relationship-concept triplets. In other words, in Protégé and OKBC, slots are nominally designed to represent binary relations. Slots also have facets, which specify the built-in constraints on attributes for slots such as value type, maximum and minimum values, cardinality, etc. Facets are a limited form of ternary relations. However, this framework does not facilitate attaching user-defined attributes to a slot (or relation).

There are several possible solutions to work around this situation. The conventional method to model higher order relationships or to assign attributes to a relationship, in any KR system where slots are binary relations (i.e. at least any OKBC-compliant system), is to reify such relations as concepts (classes). In order to do so, we can create new classes with a structure that allows recording multiple concepts of multiple-arity relationships. This process is called *reification* and the resulting concepts are known as *reified concepts or relations*. Naturally, this solution also works if there is a need to have additional attributes assigned to the relationships.

An example of this would be the following:

- PERSON has-membership MEMBERSHIP
- MEMBERSHIP has-interval TIME INTERVAL
- MEMBERSHIP has-group GROUP OF PEOPLE
- MEMEBRSHIP has-member PERSON

When an instance of MEMBERSHIP is created, the participants in the MEMBERSHIP can be specified as well as the time value for that “relation”.

The lack of built-in support for higher order relations makes things such as visualization more cumbersome. In order to overcome this inconvenience, the Protégé development group has introduced a couple of plug-ins.

The “:RELATION” class is intended to be a tag class for reified relationships. This plug-in allows the user to view the reified relationship hierarchy in the same way that one can browse regular relations in Protégé [Relation]. This feature is used in a couple of visualization plug-ins that simplify the visualization of instances of reified relations. This

is done by creating graphs which are connected with edges corresponding to the instances of reified relations.

At the time of our implementation, the RELATION tab and associated visualization plugins were not available yet, thus we created reified relations as concepts in our domain and grouped them all under an abstract concept called “REIFIED RELATION”.

In our design, we applied the reification for two main purposes. The first purpose is to assign attributes such as time to the relations that hold in our ontology. The second purpose is to implement temporal dynamic hierarchical structures such as our position and place hierarchy. Even though reification was an inconvenience for our former purpose, we would not be able to define the latter if we could not encapsulate the triplet of concept-relationship-concept as a new reified concept.

Figure 4-16 to Figure 4-25 illustrate the reified relations intended to capture the time element in a relation. For each reified relation, we present a figure which contains: a diagram illustrating the model designed for transforming the relation to a concept, the implementation of the new concept in protégé, and an example of the instantiated reified relation.

In order to reify a relation, we defined a new concept (class) with a triplet of slots: concept1-concept2-time. In addition to this, we captured any attribute associated with the concept or relations in this new reified class as well.

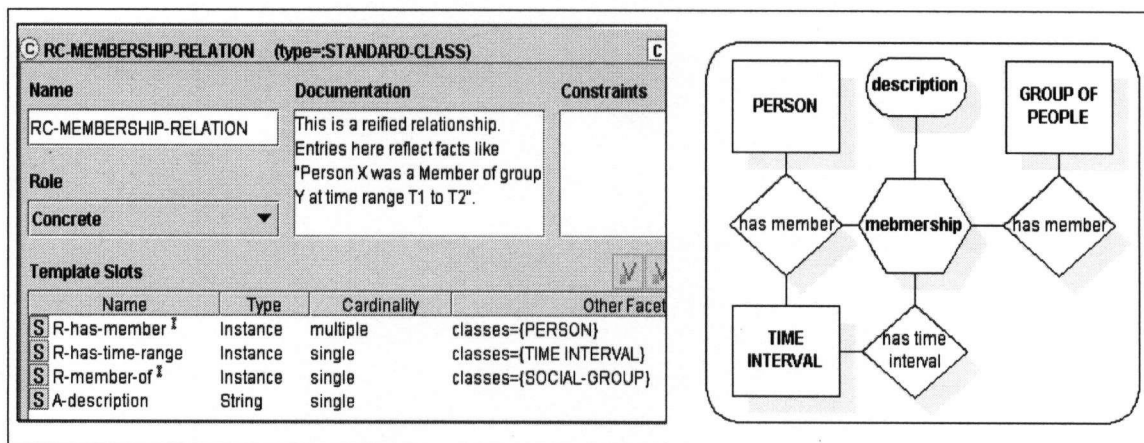


Figure 4-16 Reification of the relation MEMBERSHIP as a concept

This figure illustrates the creation of a new concept MEMBERSHIP which represents the membership relation that holds between a “PERSON” and a “GROUP” in a time range (TIME

INTERVAL). It also captures the attribute “description” which provides extra information about any instances of the class **MEMBERSHIP**. The right part of the figure is our designed model, the left part of the figure is the transformation into Protégé-2000.

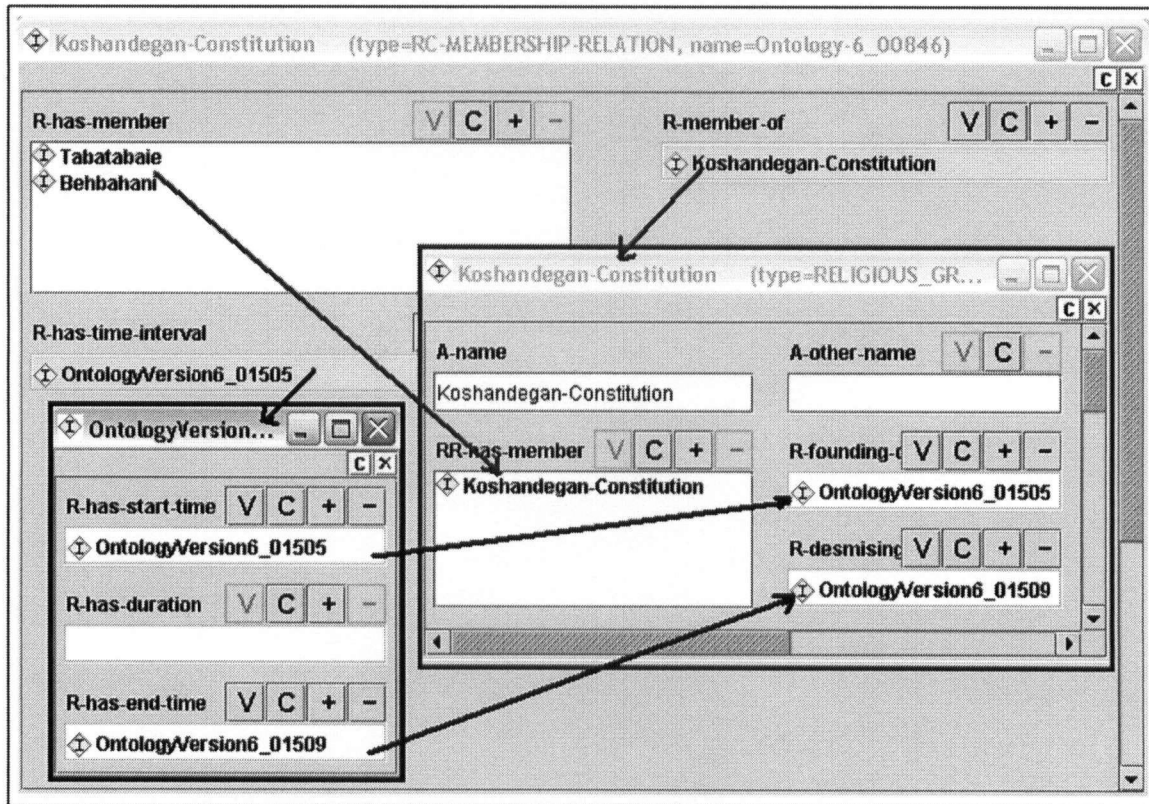


Figure 4-17 An instance of the reified relation “MEMBERSHIP”

This figure shows that two people: “Tabatabaie” and “Behbahani” (top left of figure) were the members of a group called “Koshandegan-Constitution” (window in the middle-right) for a certain time range (bottom left side of screen).

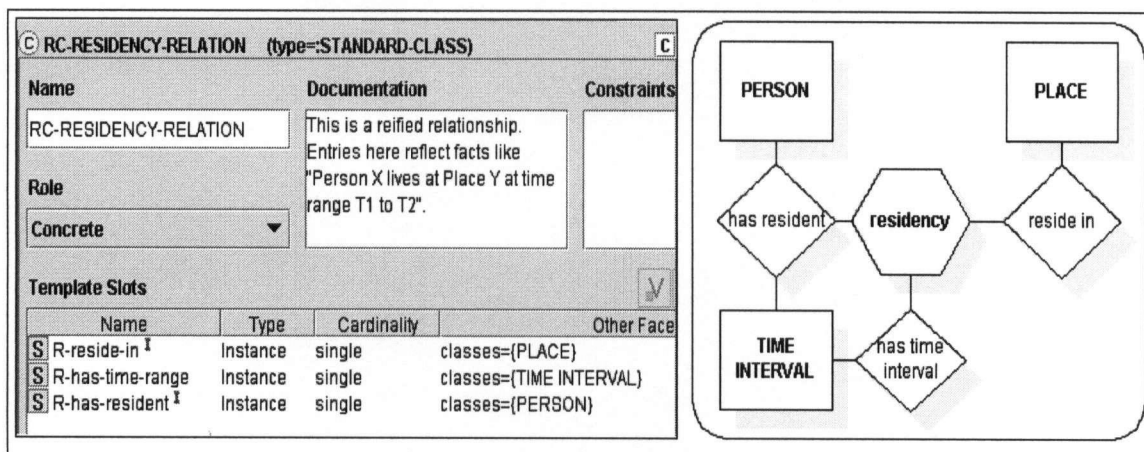


Figure 4-18 Reification of the relation **RESIDENCY** as a concept

This figure illustrates the model used to capture the “RESIDENCY” relation that holds between a “PERSON” and “PLACE” during a “TIME INTERVAL”. The right part of the figure is our designed model, the left part of the figure is the transformation into Protégé-2000.

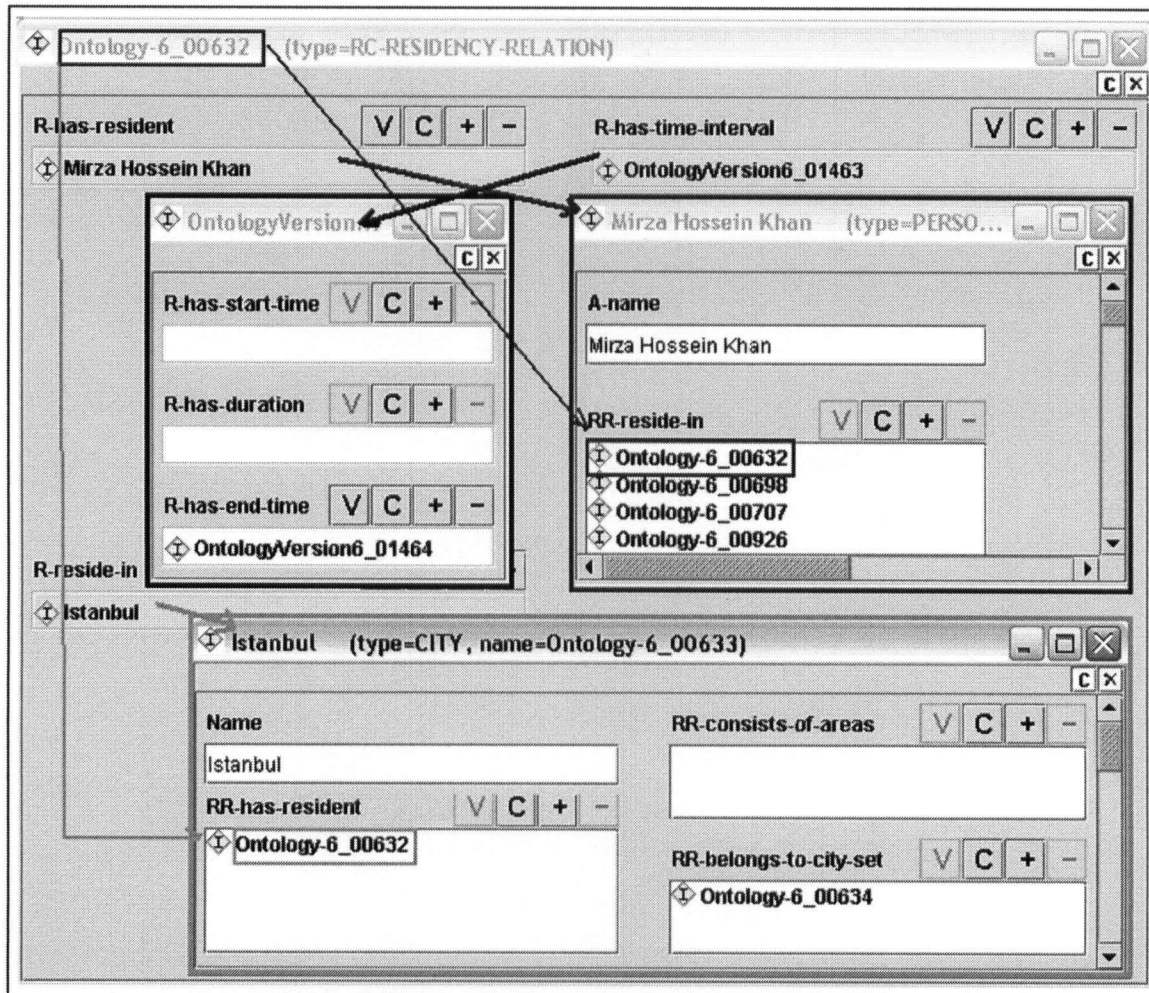


Figure 4-19 An instance of the reified relation RESIDENCY

In this example, PERSON “Mirza Hossien Khan” (top right side of figure) was living in CITY “Istanbul” (bottom window) during a time interval (top left side of figure). As it can be noticed from this figure, for this instance we only knew the time of completion of the RESIDENCY relation, not the time it begun nor its duration.

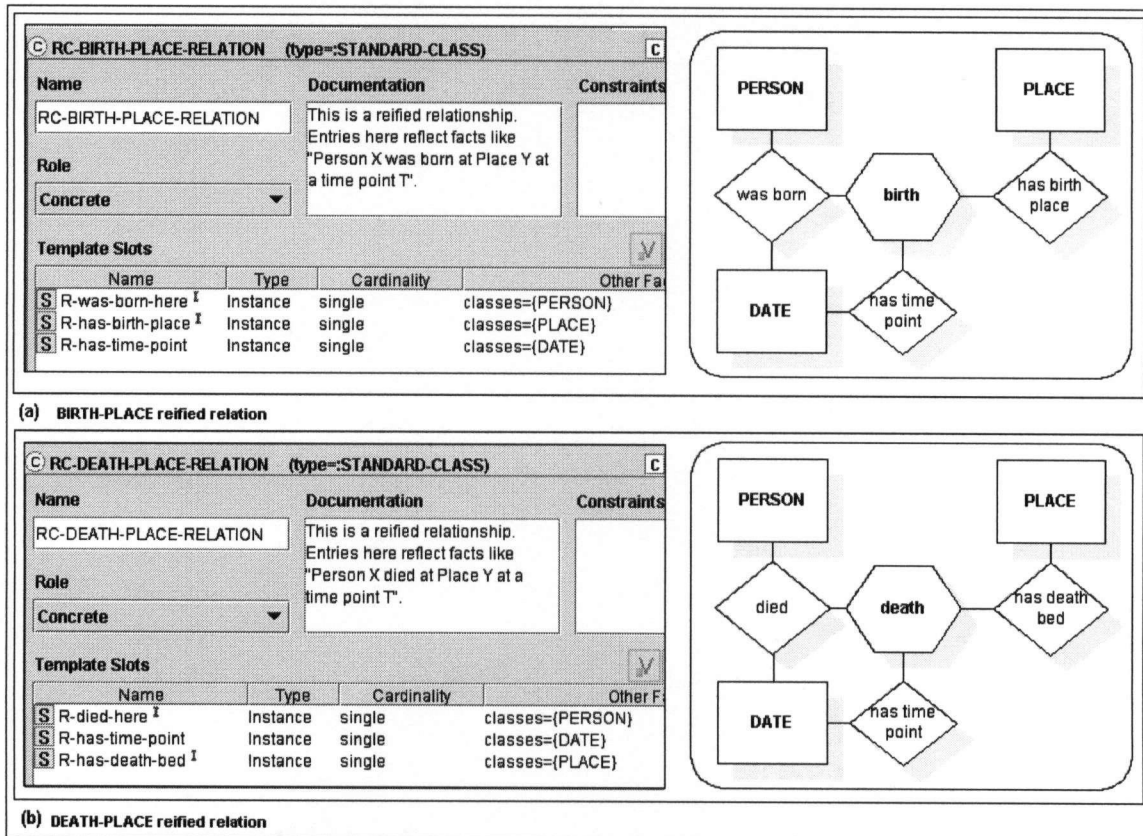


Figure 4-20 Reification of the relations BIRTH-PLACE and DEATH-PLACE as concepts

The BIRTH-PLACE and DEATH-PLACE reified relations are presented as a triplet that consists of a PERSON, PLACE, and a TIME POINT in which the person was born or died. The right part of the figures are our designed models, the left part of the figures are the transformation into Protégé-2000.

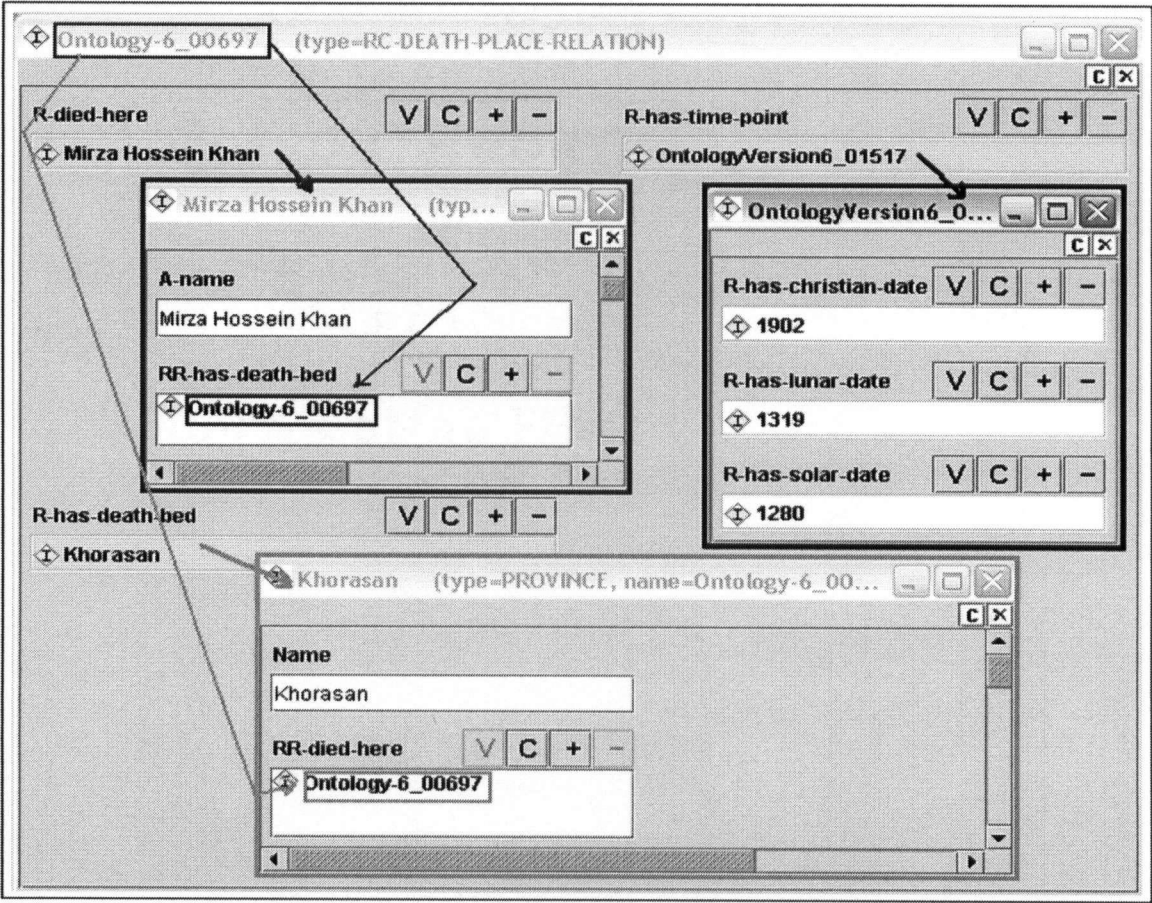


Figure 4-21 An instance of the reified relation DEATH-PLACE

This figure illustrates an example of the DEATH-PLACE relation in our domain. In this example PERSON “Mirza Hossein Khan” (top left window) died in PROVINCE “Khorasan” (bottom of the figure) at a certain time point (top right window). An example for the BIRTH-PLACE reified relation would be identical to this one.

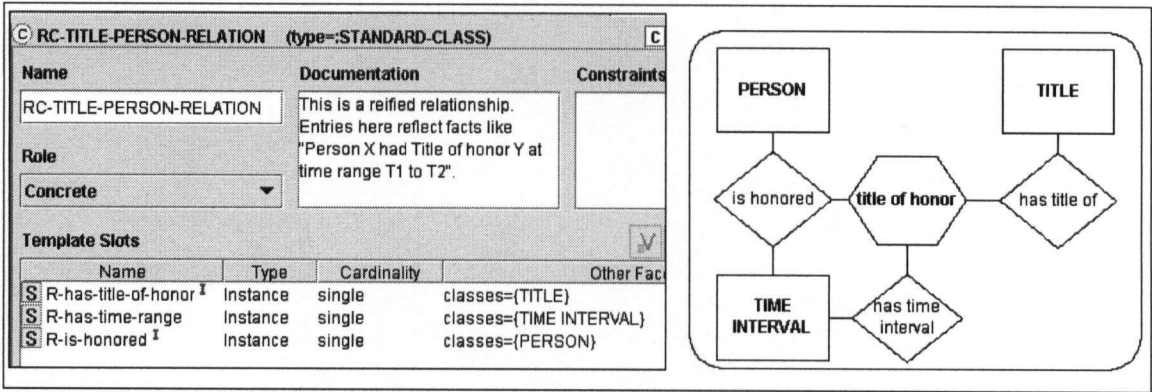


Figure 4-22 Reification of the relation TITLE-OF-HONOR as a concept

The association of a TITLE to a PERSON for a time period is modeled through a reified relation presented as TITLE-OF-HONOR in this figure. The right part of the figure is our designed model, the left part of the figure is the transformation into Protégé-2000.

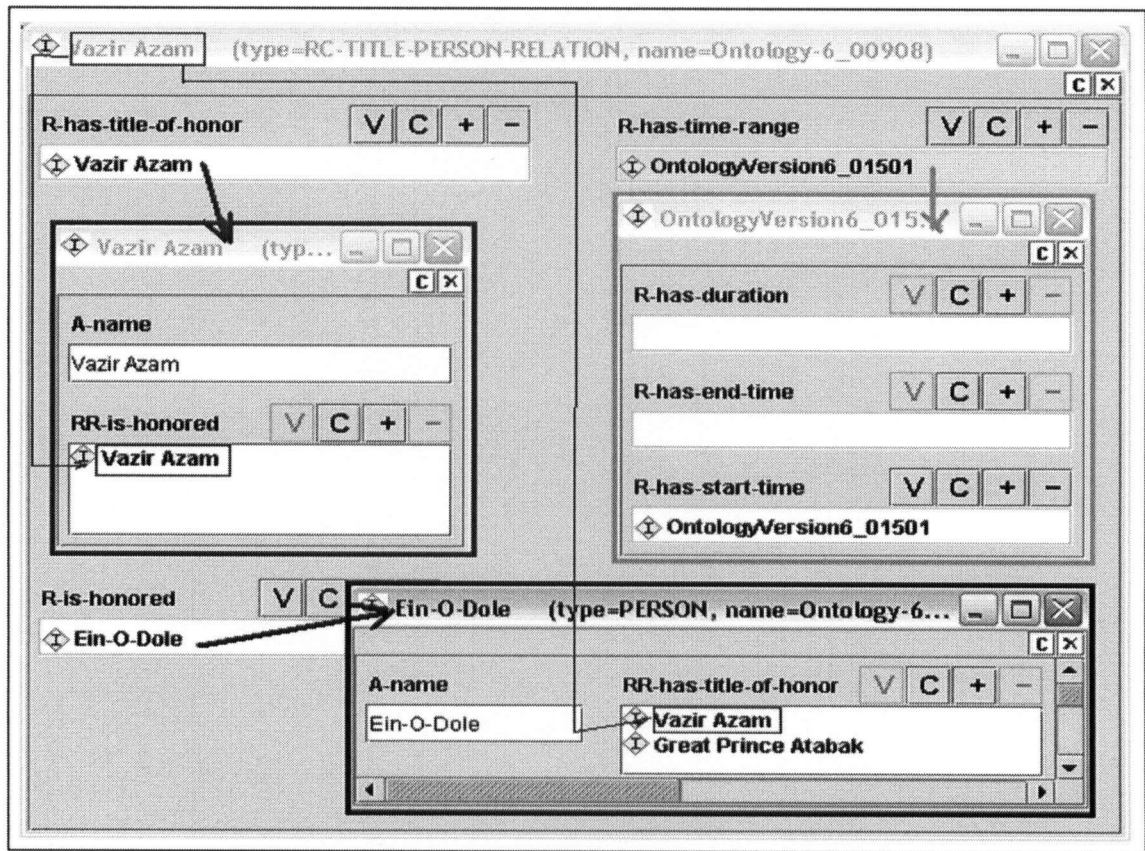


Figure 4-23 An instance of the reified relation TITLE-OF-HONOR

This figure shows an instance of the reified relation TITLE-OF-HONOR in our domain. A PERSON “Ein-O-Dole” (bottom left) was honored with the TITLE “Vazir Azam” (window on the upper left) for a period of time (top right side of the figure). As it is obvious from this figure, we only knew the beginning of the time period in which he held the title, not the end date nor the duration.

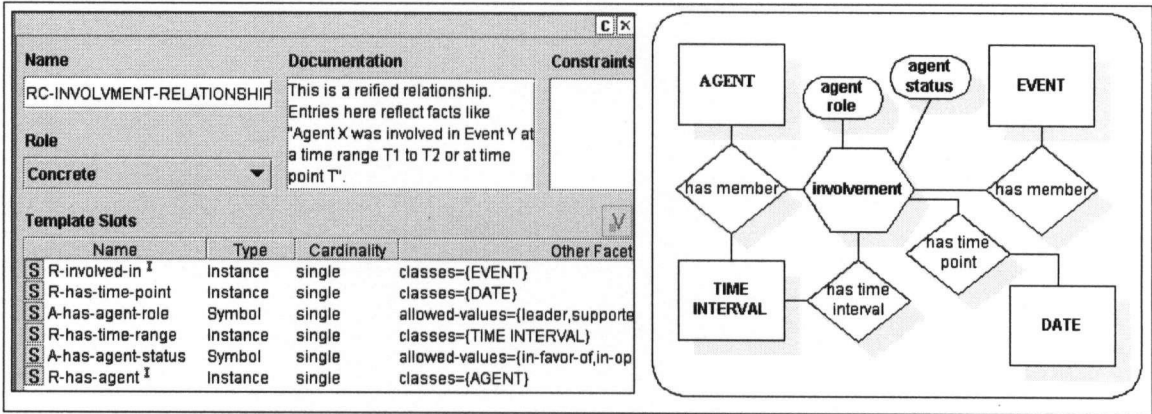


Figure 4-24 Reification of the relation INVOLVEMENT as a concept

The relation INVOLVEMENT is reified as a concept which consists of a PERSON, an EVENT, and the TIME POINT or TIME INTEVAL in which the involvement association is valid. The right part of the figure is our designed model, the left part of the figure is the transformation into Protégé-2000.

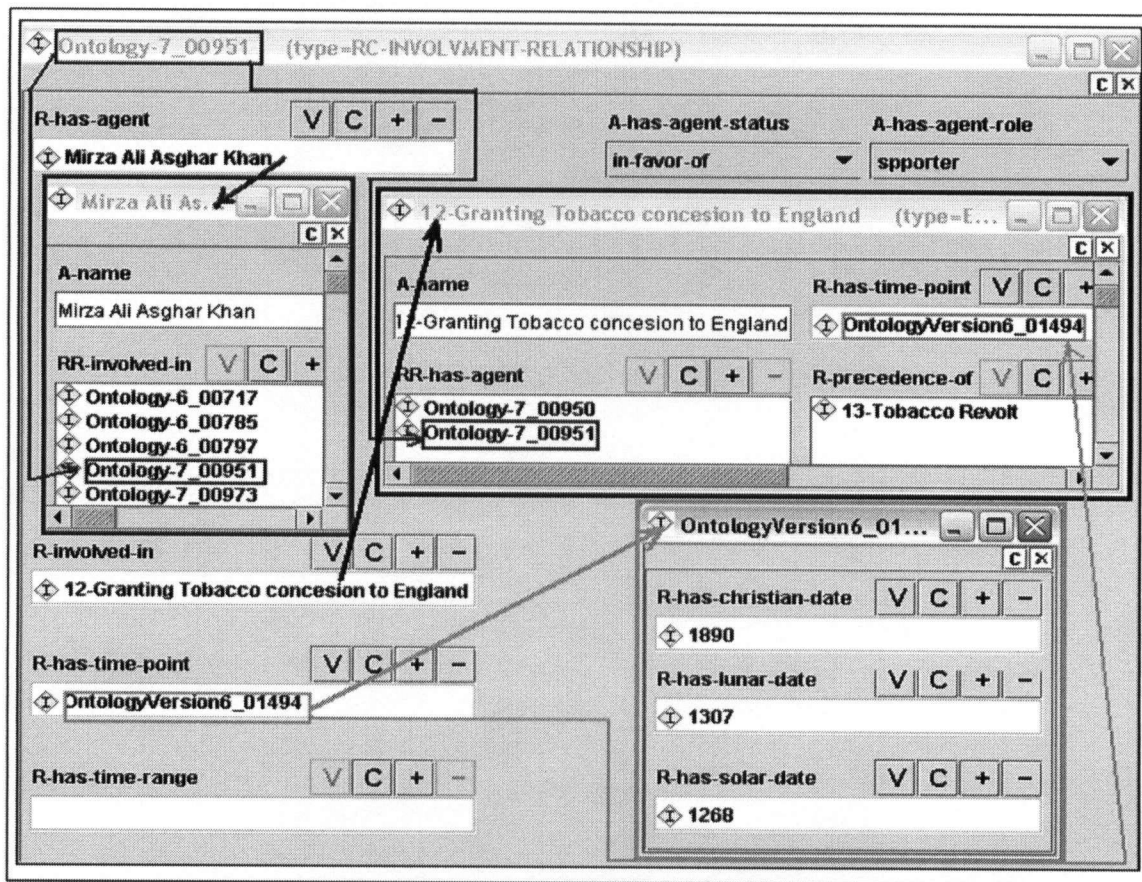


Figure 4-25 An instance of the reified relation INVOLVEMENT

In this example a PERSON “Mirza Ali Asghar Khan” (top left window) is involved in an EVENT “Granting Tobacco concession to England” (top right window) at a TIME POINT in “Christian year 1890” (bottom right window).

Having explained how reified relations were handled, we proceed to explain the definition of our temporally dynamic hierarchy of places and positions.

In our ontology, we defined different types of places (continent, country, province, city, area) which are interrelated. These interrelations (belongs to /is part of) are time dependent. In order to capture this dynamic temporal aspect of our hierarchy of places, we defined a set of reified relations as concepts. These include those concepts represented in Figure 4-26. The concepts in this figure are defined as SET concepts such as COUNTRY SET, PROVINCE SET, CITY SET, and AREA SET.

At any time point, the place hierarchy is conformed by sets of instances with different types of places. For these places, there is a part/whole (consist-of / belongs-to) relation

that holds at each level of our place hierarchy. This means that any place class in the hierarchy consists of instances of the immediate lower level place class at any given time. In other words, the summation of every instance of the concept in the lower level will construct the concept in the higher level within the hierarchy at that certain time period. We used this feature to our advantage in our design and created our reified relations in a manner that captures the aforementioned characteristics.

As an example of this, consider the class COUNTRY. We know that a country is located in a continent; moreover it consists of a set of provinces at any given time. As illustrated in Figure 4-27 and Figure 4-28, for each instance of the concept country we can capture two things that hold at a certain time: to which country set it belongs and which provinces it consists of. The country set to which this particular country belongs to forms a continent, again for a specific time interval.

Figure 4-29 illustrates an example of the aforementioned association. In this figure, Country "Iran" is located in the continent "Asia" which at that particular point of time consists of the following countries: Iran, Osmani, and Russia. We should note that only those countries relevant to our domain were included in our ontology. At this same time point, the country Iran consists of a set of provinces: Espahan, Gilan, Khorasan, and Tehran. We used a "TIME INTERVAL" to capture the temporal aspect of this relation. In this case we only knew the time that the association between these places ended. Many of these times are unknown because they fall outside of the time range that our ontology intends to represent (the time range covered by our history book)

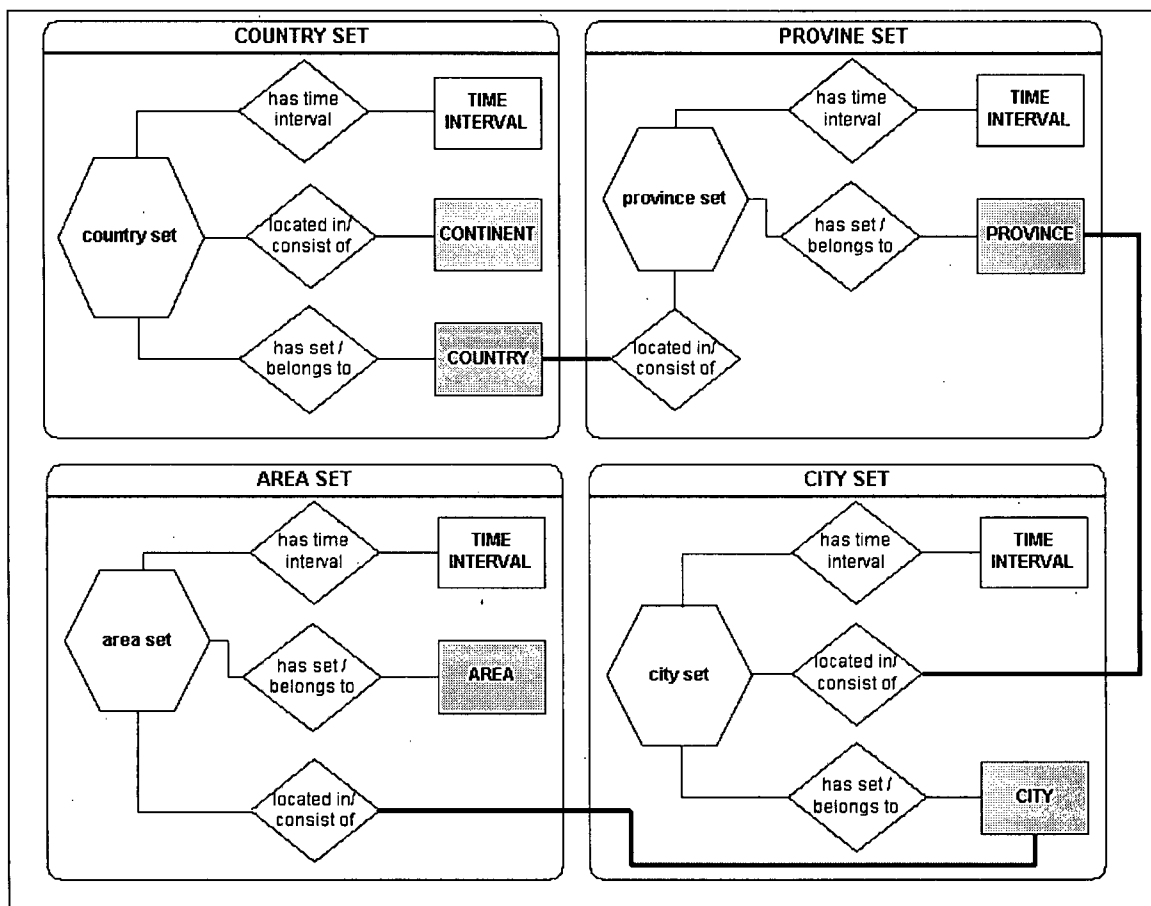


Figure 4-26 Modeling the temporal place hierarchical organization

This figure illustrates the model we utilized to create a time-based hierarchy of places. As shown, there are five different types of places in our ontology: CONTINENT, COUNTRY, PROVINCE, CITY, and AREA. Each of these is conformed by a set of instances of another type of place (at a lower level in the place hierarchy). In order to capture the dynamicity of the place hierarchy, we defined a set of relations such as “located-in”, “consist-of”, “belongs-to”, etc. between the places. These relations are all associated with time.

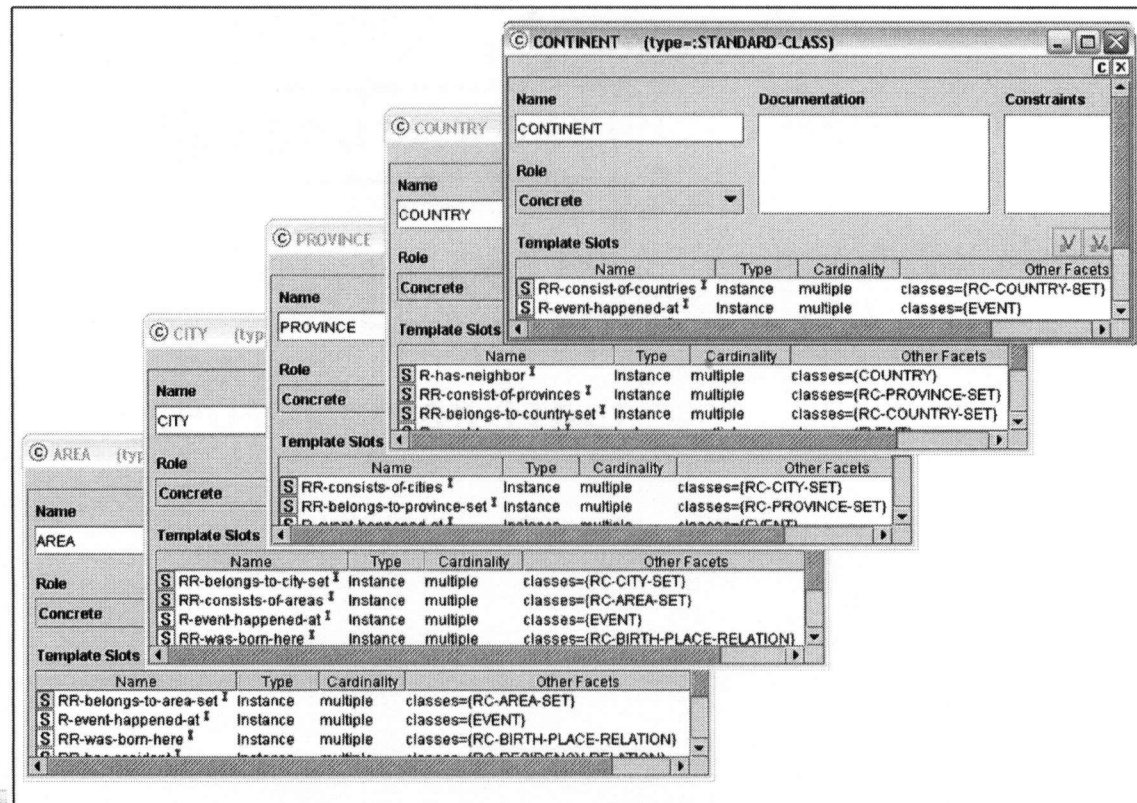


Figure 4-27 Class implementation of main place concepts in Protégé

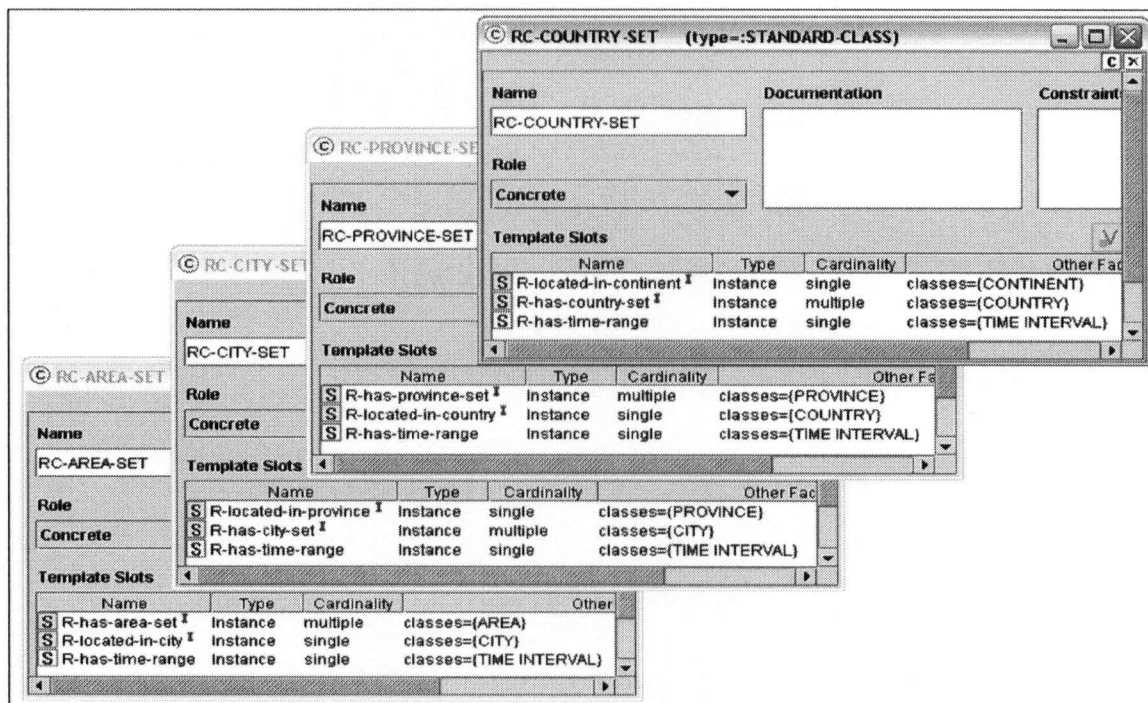


Figure 4-28 Reified relations used to represent the temporal dynamic organization of places in Protégé

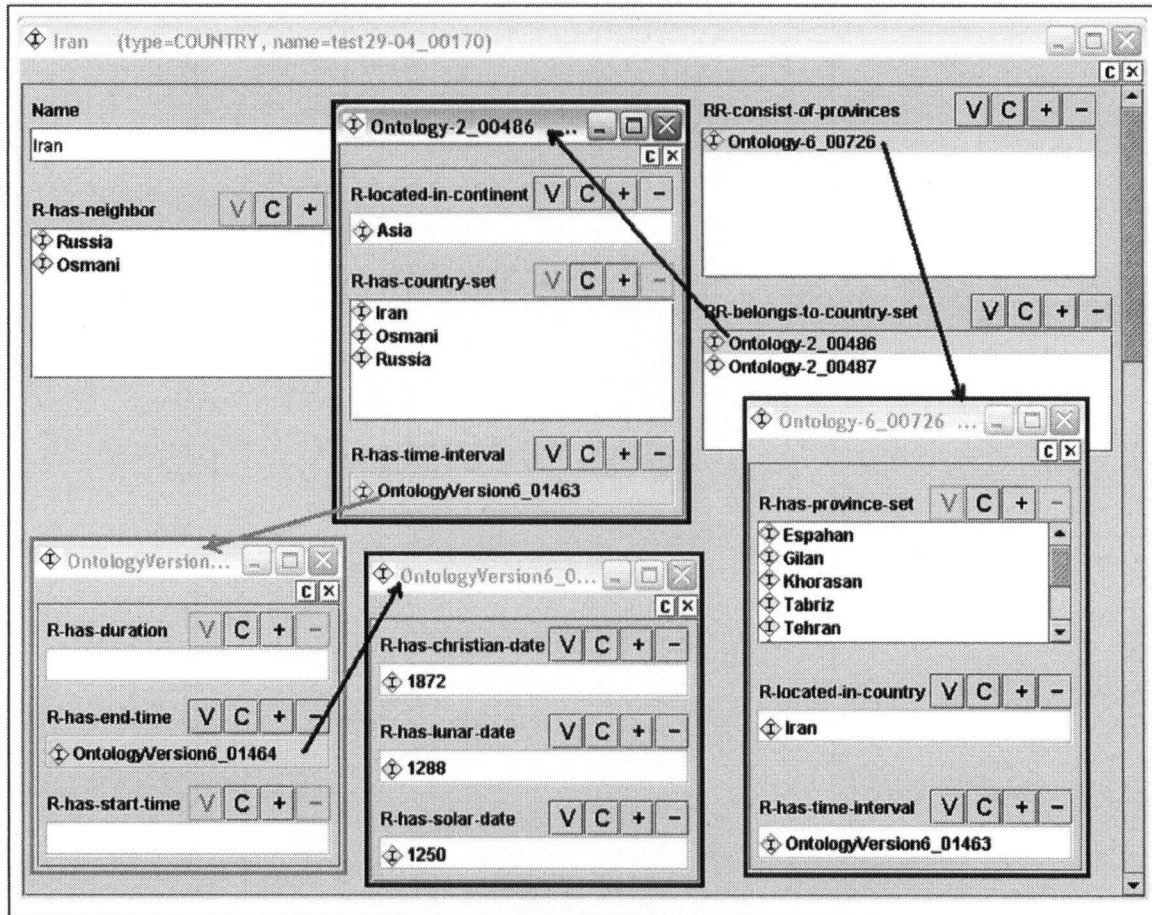


Figure 4-29 An instance of the place hierarchy in Protégé

This figure illustrates that a COUNTRY Iran was located in the CONTINENT Asia (window in the top-middle), which at that time consists of Iran, Osmani, and Russia. The same COUNTRY Iran consists of a set of provinces (bottom-right window). All these relations hold for a time range which is represented by the concept TIME INTERVAL (two windows on the bottom left side of the figure).

In a similar manner that we represented the temporally dynamic relations between places, we represent the governmental and royal court position hierarchies along with the people who hold these positions at any given time. We defined two reified relation as concepts: HOLDS-POSITION and POSITION HIERARCHY (Figure 4-30). The HOLDS-POSITION concept is created to represent the association between a PERSON and the POSITIONS that this person might hold at any given time. This reified concept also captures the inferior and superior of each position in a similar manner (person, place, time). The POSITION-HIERARCHY concept aims to represent the hierarchical structure of positions in an organization which, in our domain, refers to either a governmental or a royal court organization. Figure 4-30 explains our conceptual model associating both the POSITION-HIERARCHY and HOLDS-POSITION reified concepts. The HOLD-

POSITION relation is associated to the POSITION-HIERARCHY through a “belongs-to position hierarchy” relation. This encompasses all the people that hold a position in that hierarchy along with how these positions were related to other positions in the hierarchy at that time. We should note that, all these relations are time based and are thus associated with a time interval.

Figure 4-31 shows how we utilize Protégé-2000 to implement this hierarchy. An instance of this dynamic hierarchy is illustrated in Figure 4-32. In this figure, the position “king” was held by a “PERSON” called “Mozafaredin Shah” for a time interval (TIME INTERVAL). In this figure we can also see how the inferiors and superiors of that person in that specific time period are captured.

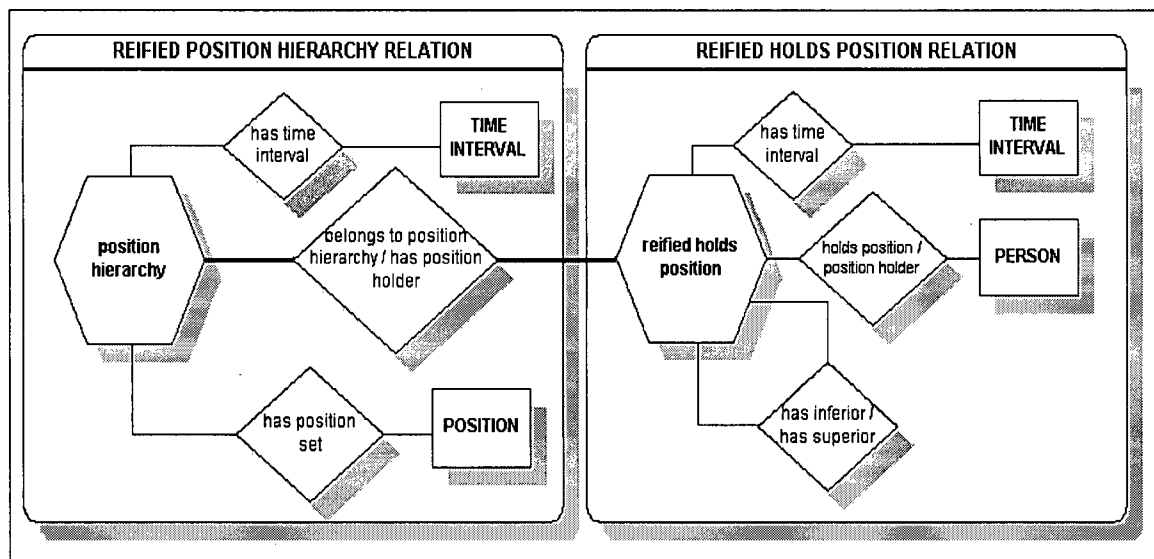


Figure 4-30 Reified relations used to capture the dynamicity of the POSITION hierarchy

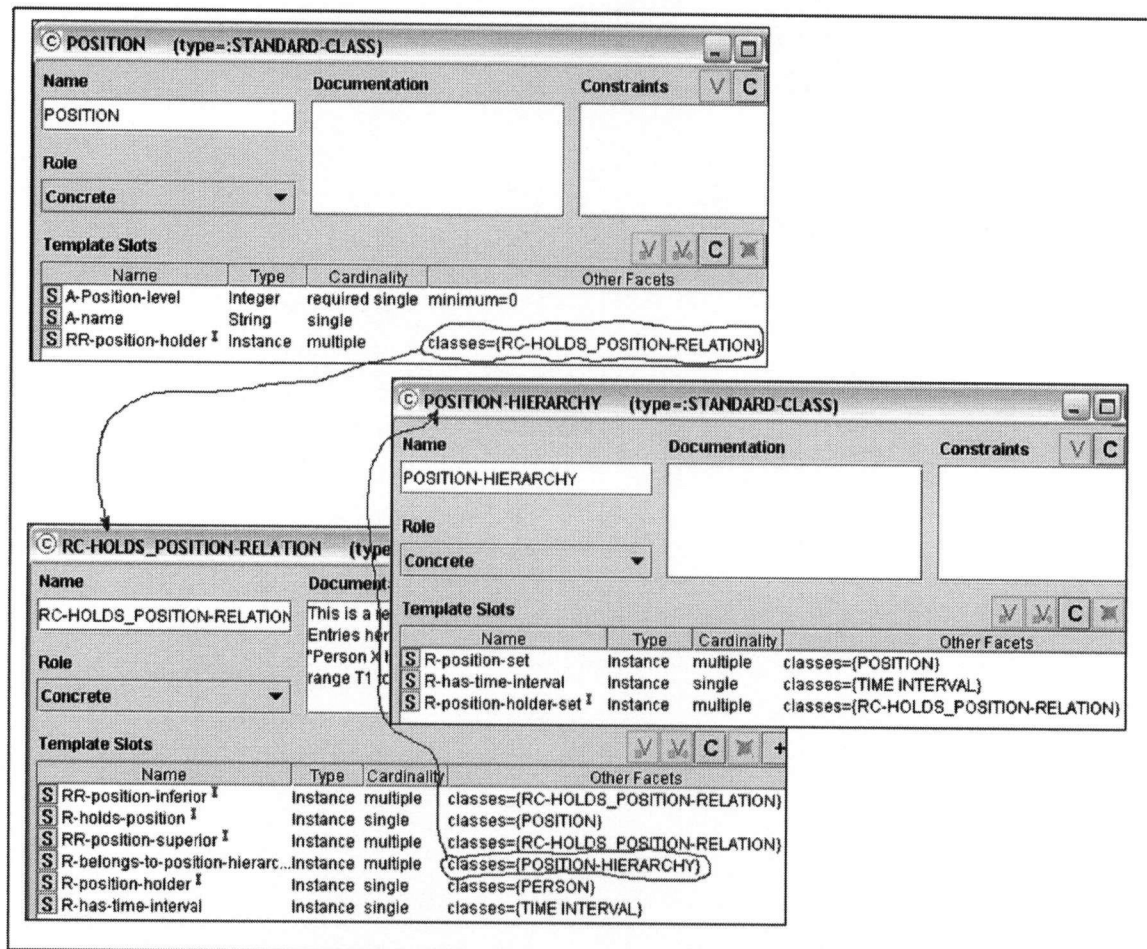


Figure 4-31 Implementation of reified relations for position in Protégé-2000

In this figure the three main concepts relating to position are illustrated: POSITION, POSITION-HIERARCHY, AND HOLDS-POSITION. It also demonstrates how we defined relations used to associate these concepts.

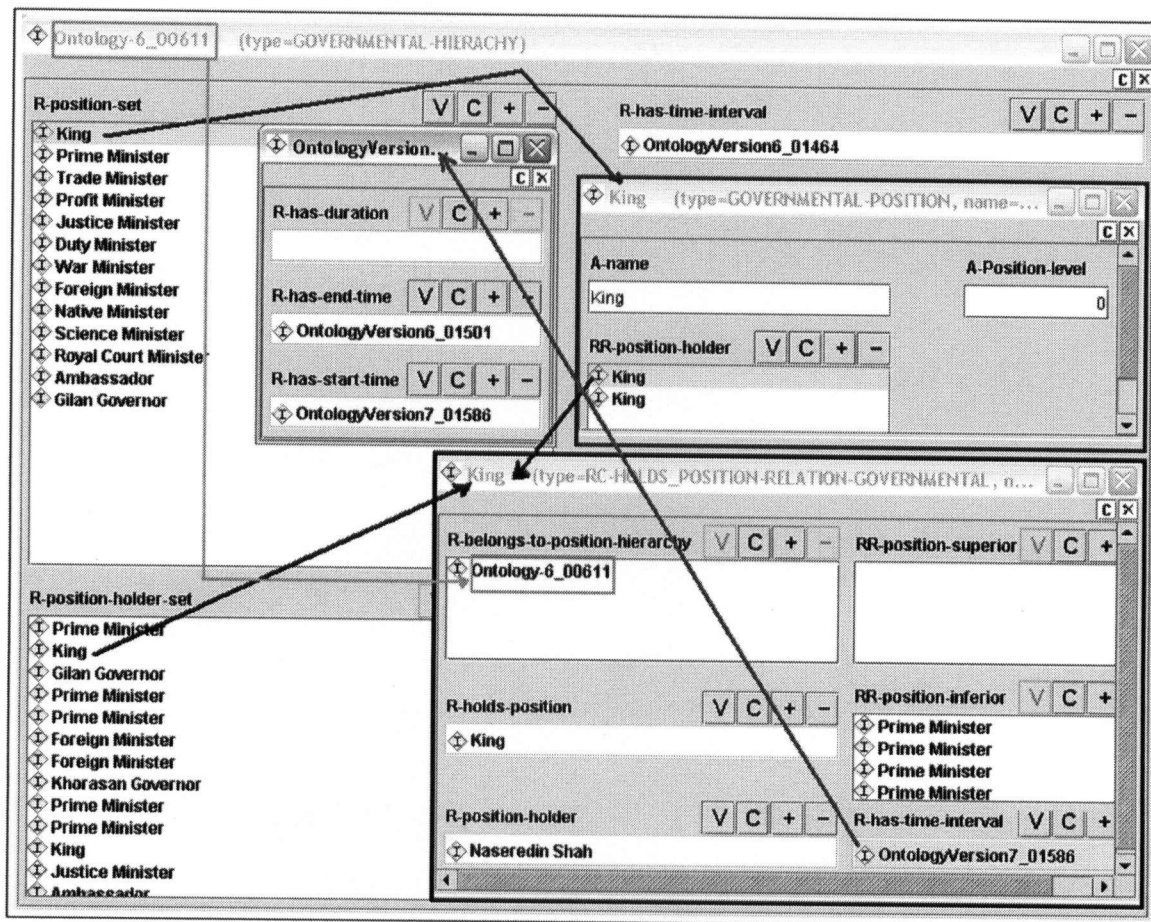


Figure 4-32 An instance of the reified relation POSITION-HIERARCHY and HOLDS-POSITION in Protégé-2000

This figure illustrates an instance of the position hierarchy that holds for a specific time interval in our domain. In this example, the position “king” (top right window) was held by a person called “Mozafaredin Shah” (bottom right window) at that time interval. We can also see the inferiors and superiors of that person in that specific time period (within the bottom right window).

4.2.3.3 Defining Constraints

The next step in our formal ontology development is to specify constraints on the properties (slots) of the classes we defined. We do this, in order to ensure the consistency of our ontology. This includes defining constraints on both attributes and relations that these classes represent. We used built-in facets provided in Protégé to assign constraints on slot values. Amongst the facets we used are: cardinality of a slot (how many values the slot can have), restrictions on the value type of the slot (for example, integer, string, instance of a class), minimum and maximum values for a numeric slot, etc (Section 3.1.2.4.2). In addition to these facets, we used PAL axioms to define more complex user-

specified constraints than those possible with facets (Section 3.1.2.4.2). The PAL axioms are used specifically for consistency checking of the temporal aspect of our ontology. Amongst the things we checked by using these axioms is whether any instance of the relations that associate two concepts, have discordances in their time attributes. For example, we must ensure that for any person the birth-date precede the death-date. These axioms were also used to check for the existence of loops within the ontology. As an example of possible loops in our ontology, we could have a case in the position hierarchy, where person A is at the same time a superior and an inferior of person B.

4.2.4 Ontology Evaluation

After designing, building, formalizing our ontology using Protégé and enforcing constraints on attributes and relations, we used the knowledge acquisition forms provided in Protégé to instantiate our history ontology. Over seven hundred and fifty (750) instances were extracted from the history book and included in our ontology. Amongst these instances we find: *people, places, documents, events, etc.*

Before we started using the ontology, we needed to evaluate the model as a whole to examine whether it satisfied the previously specified motivating scenarios and whether it provided answers to all the competency questions we designed it for.

In order to test each competency question or motivating scenario we employed one or more of the following choices provided by Protégé:

- Using the built-in query engine to provide answers for simple queries. (Figure 4-33) Our implementation allows us to answers those competency questions that are directly related to instances of concepts by using the built-in query engine. In Figure 4-33 we looked for instances of people who held the position “king”.
- Using the PAL query plug-in to create more sophisticated queries. (Figure 4-34). We addressed those competency questions which could not be answered using the built-in query engine by posing them using the PAL query plug-in. This plug-in allows us to make more complex searches like the example presented in Figure 4-34. In this example we search for places where one or more events took place.

- Using visualization facilities provided by Protégé plug-ins in order to browse the ontology to ensure that our representation is accurate and consistent. Figure 4-35 and Figure 4-36 show two of the different available visualization plug-ins in Protégé-2000. In these two figures the place hierarchy related to country Iran is presented. Visualization aids are particularly helpful when trying to understand hierarchical relations.

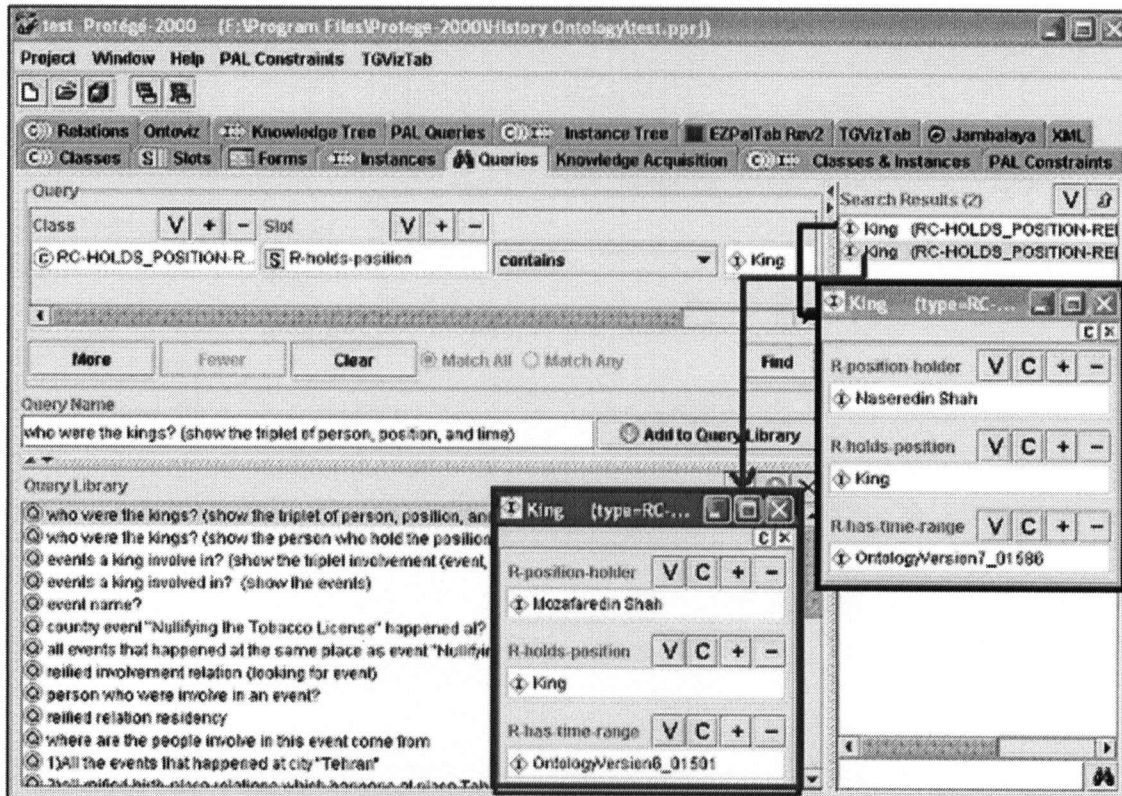


Figure 4-33 Using Protégé-2000 built-in query engine to answer competency questions

In this example we answer the question “Who were the kings?” using the Protégé-2000 built-in query engine. Our model returns two instances of the triplet “person, position, and time” (in the outlined windows). In this case “Mozafaredin Shah” and “Naseredin Shah” held the position “KING” at different time intervals. The bottom left part of this figure shows a few of the questions that our ontology is designed to answer.

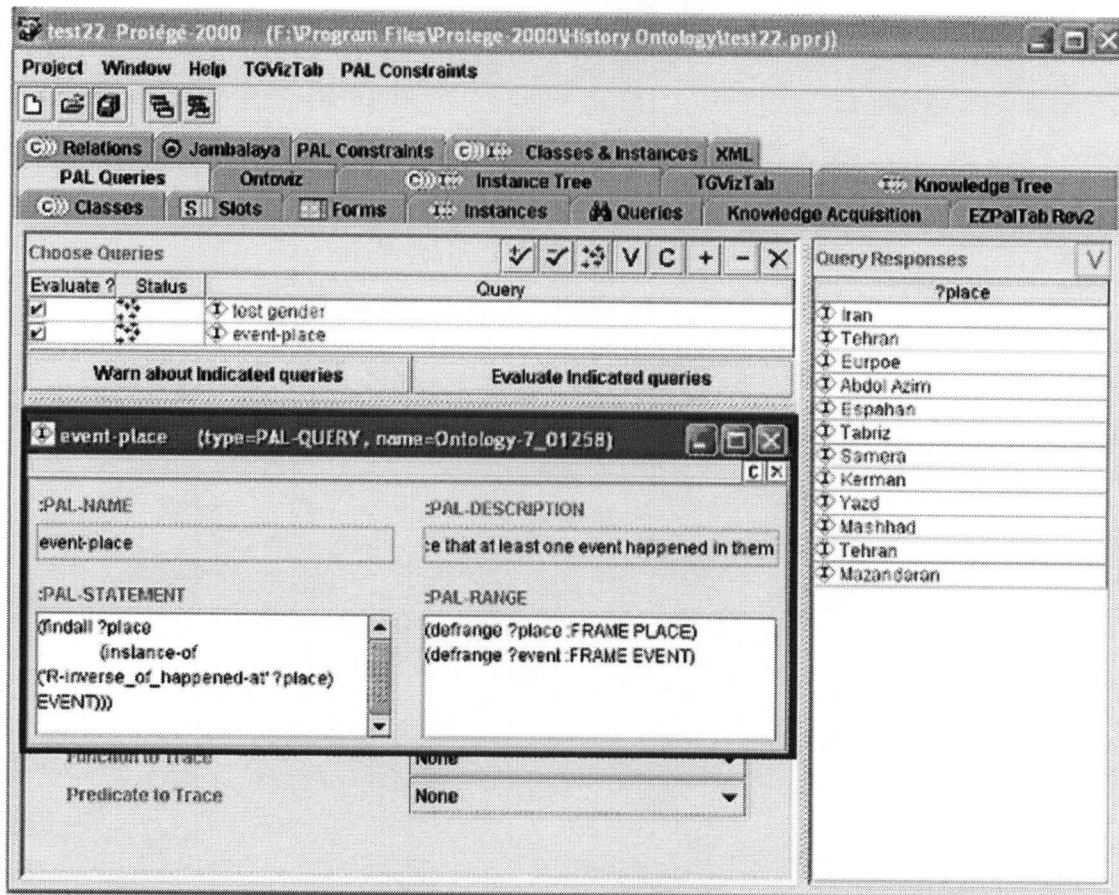


Figure 4-34 Using the PAL query engine to answer competency questions

In this example we look for places where at least one event has happened. The pop-up window (window in the middle of the screen) illustrates the PAL query statement written for this search. The result for this search (right side of screen) is a list of places where one or more events had taken place. If we select any of these instances, detailed information about that particular place is presented.

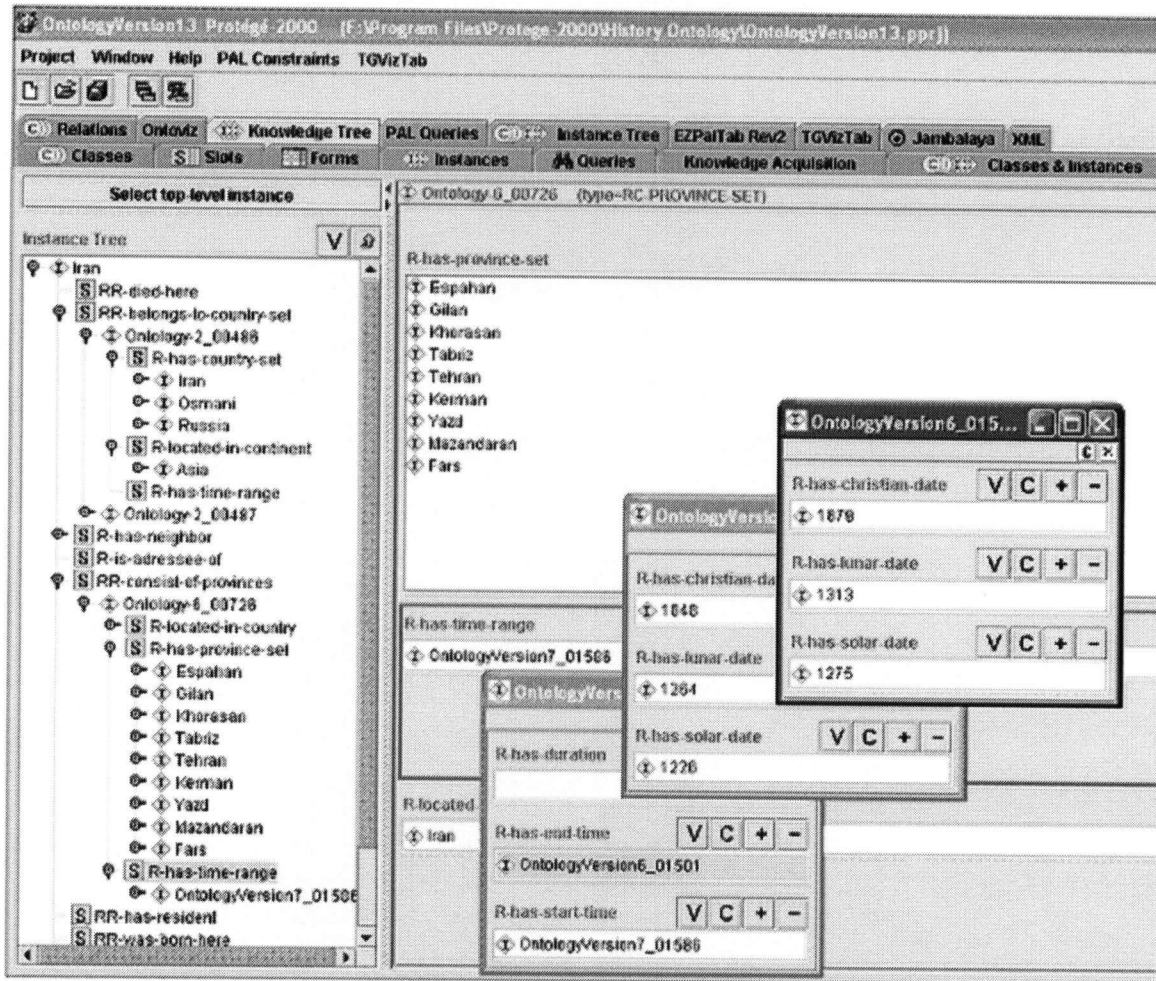


Figure 4-35 “Knowledge Tree” visualization of the place hierarchy for “Iran”

In this example we utilized the “Knowledge Tree” visualization Tab plug-in to demonstrate the hierarchy of places related to “Iran”. On the left side of screen we can see all the notions that relate to country “Iran”. Amongst these notions we may find the country set to which it belongs (Iran, Osmani, and Russia), the provinces it consists of (Espahan, Gilan, Khorasan, Tabriz, Tehran, Kerman, Yazd, Mazandaran, Fars), and the continent it is located in (Asia). All of these relations for a given time range (windows in the right side of screen). This visualization plug-in allows us to brows through all this information in a single screen.

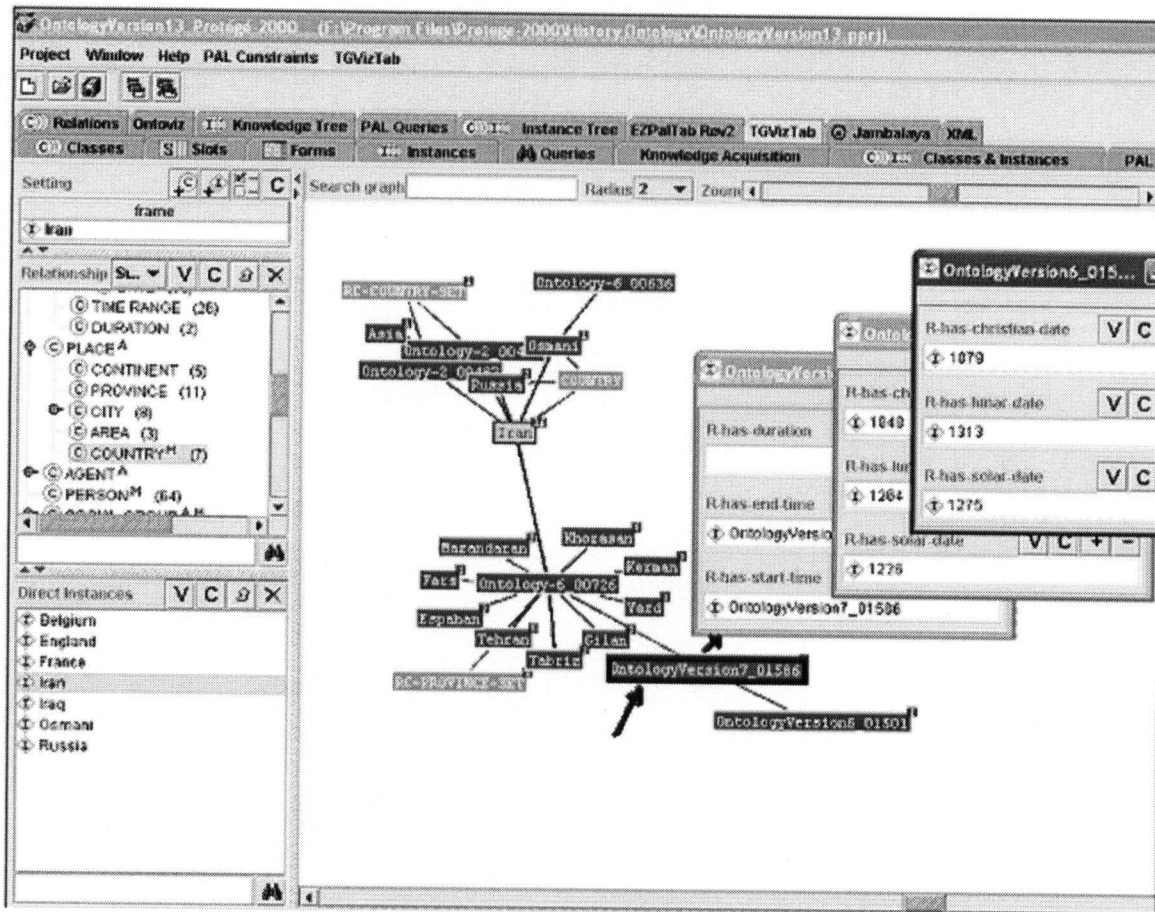


Figure 4-36 Using TGViz visualization tab to browse the hierarchy of places related to country “Iran”

In this example, the hierarchy of places related to country “Iran” is presented in a graphical manner where every node in the graph represents a class or instance and the lines connecting them denote a relation. By selecting any of these nodes additional information related to that particular node is presented. In this example we selected the notion time interval (outlined) and the information about that particular node is presented in pop-up windows (right side of screen). In this case the relations represented by the graph hold from year 1849 to 1879 in the Christian calendar.

We ran an exhaustive set of queries to ensure that the system provided satisfactory answers to all the competency questions and adequate, useful solutions to all motivating scenarios. Our testing yielded satisfactory results. We were able to provide adequate answers to each and every competency question we had designed our system to answer. In addition to this, the system provided facilities to support the requirements we had imposed on it with our motivating scenarios as well.

As an example of this exhaustive set of queries, the following section describes how our implementation provided an answer to the competency questions stated in Table 4-5. For clarity purposes, we performed the set of queries for a particular PERSON “Mirza

Hossein Khan” and a particular GROUP of people “Koshandegan”. To avoid repetitiveness, for the cases where two or more questions were similar, only one question was answered.

As stated before we should note that Protégé-2000 is a frame-based system and as such, it can provide results and answers to queries with a granularity level of a concept (class). Therefore, the output of a query is a frame that represents a concept. In order to browse the properties (detail) of this concept, we have to select the specific concept amongst the results and its full details are presented by Protégé-2000.

The following is the list of questions taken from Table 4-1. For every question, we present a corresponding figure showing the results obtained from querying our model in Protégé.

Question 1: Who was “Mirza Hossein Khan”?

Question 2: Where/When was “Mirza Hossein Khan” born/died/reside?

Question 3: What title of honor he held and for how long?

Question 4: What positions did he hold and for how long?

Question 5: Who was the superior for each position that he held?

Question 6: How was the position hierarchy organized at that time?

Question 7: What documents did he make?

Question 8: In what events was he involved?

Question 9: Who else lived at the same era?

Question 10: Who were the members of group “Koshandegan-(Abdol-Azim refuge)”?

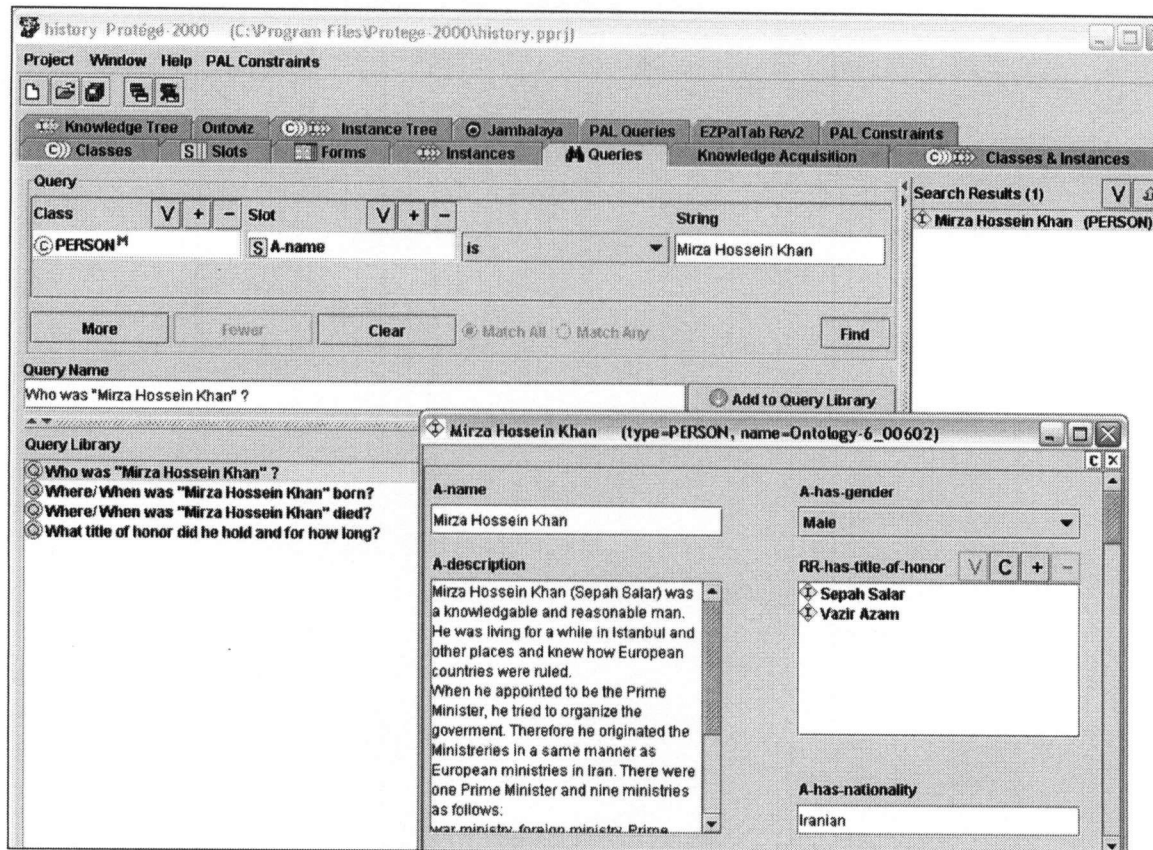


Figure 4-37 Results for Competency Question #1

Question 1: Who is "Mirza Hossein Khan"?

This figure represents a query search for the person called "Mirza Hossein Khan". The top left side of the main window illustrates the query statement. We are looking for an instance of class PERSON whose value in slot "A-name" is equal to "Mirza Hossein Khan". The right pane illustrates the results for this search. The results show we found a person with that particular name. By selecting that instance "Mirza Hossein Khan", a pop-up window appears. This window includes all the properties of the instance of class person "Mirza Hossein Khan". The user is then able to navigate through the information provided. Here we demonstrate the description field which gives an overview of the biography of the person.

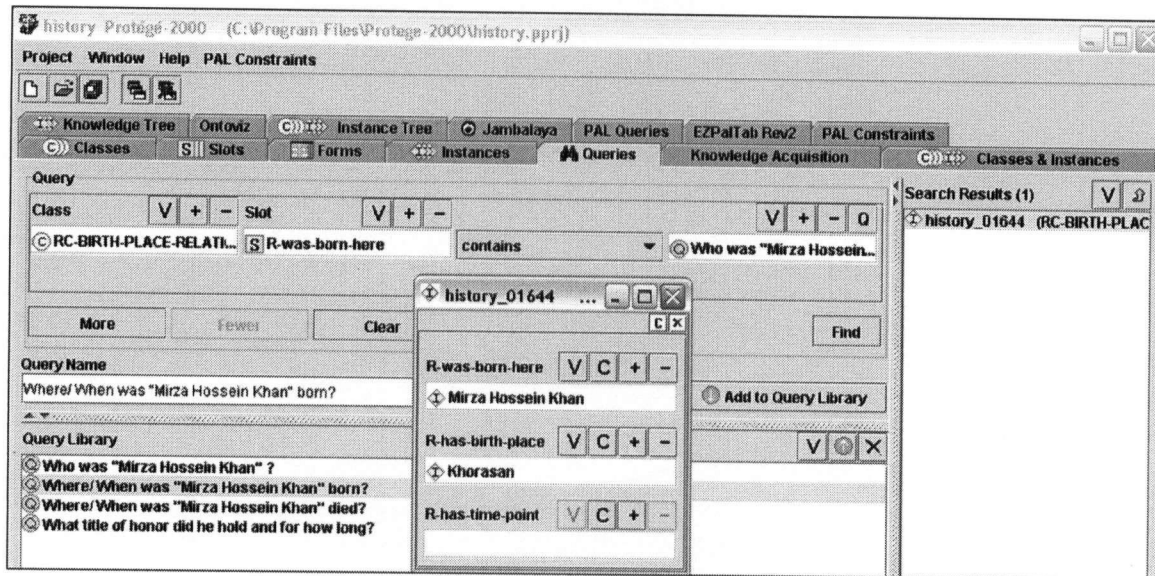


Figure 4-38 Results for Competency Question #2

Question 2: Where/When was “Mirza Hossein Khan” born?

This figure presents the query where we are looking for Mirza Hossein Khan’s “BIRTH-PLACE” and “BIRTH-DATE”. The result is shown in the right side pane. By selecting this instance we can observe that PERSON “Mirza Hossein Khan” was born in PROVINCE “Khorasan” and his birth date is unknown (window in the middle). The query process and results for capturing residency and death place are similar to this example.

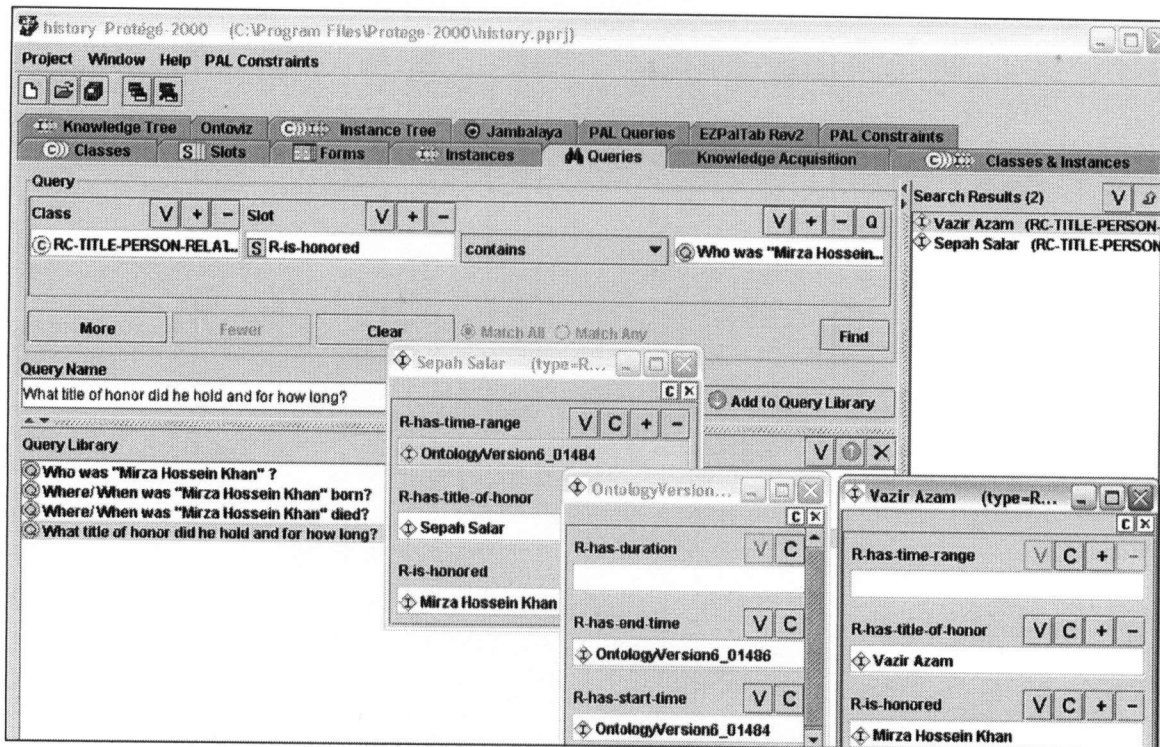


Figure 4-39 Results for Competency Question #3

Question 3: What title of honor did "Mirza Hossein Khan" hold and for how long?

In this query we look for the titles of honor that "Mirza Hossein Khan" held and the duration that those titles had. Two instance of title of honor belonging to "Mirza Hossein Khan" were found (right side pane). The pop-up windows in the figure show the information included in these instances (title of honor, name of person who held it and the duration). As it can be noticed for the title "Vazir Azam" our model does not contain the exact time interval that this title was held. However, for the title "Sepah Salar" we could find this duration.

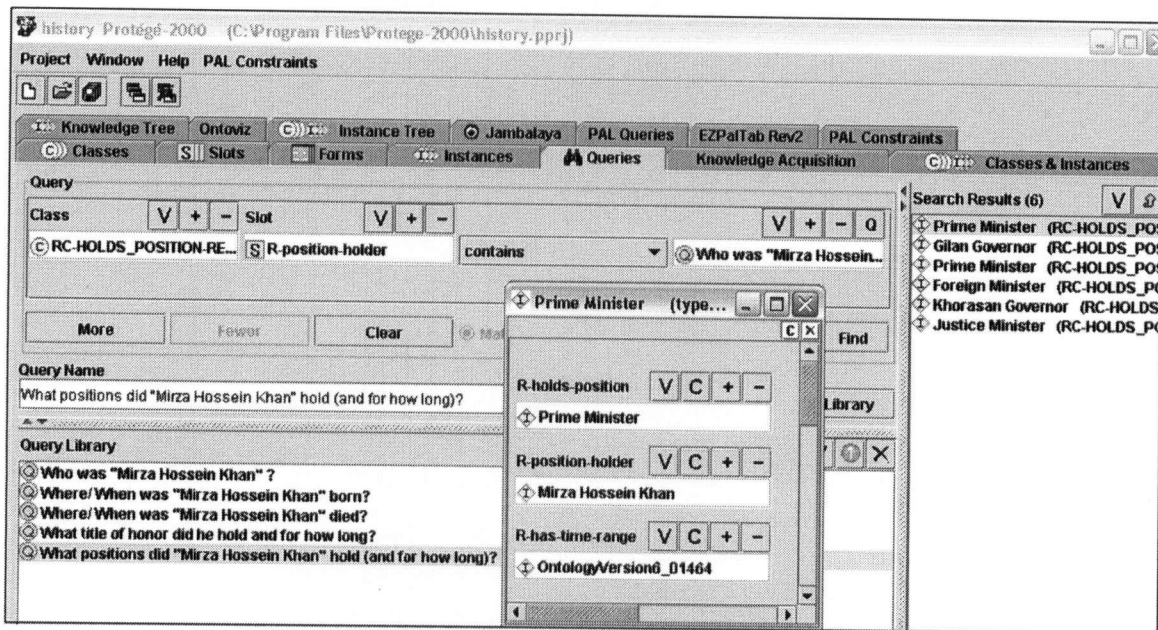


Figure 4-40 Results for Competency Question #4

Question 4: What positions did “Mirza Hossein Khan” hold and for how long?

This figure illustrates the results of the query in the right side pane. It can be seen that this person held six positions throughout time. As an instance of these results the pop-up window in the middle shows that the position PRIME MINISTER was held for a particular time interval by “Mirza Hossein Khan”.

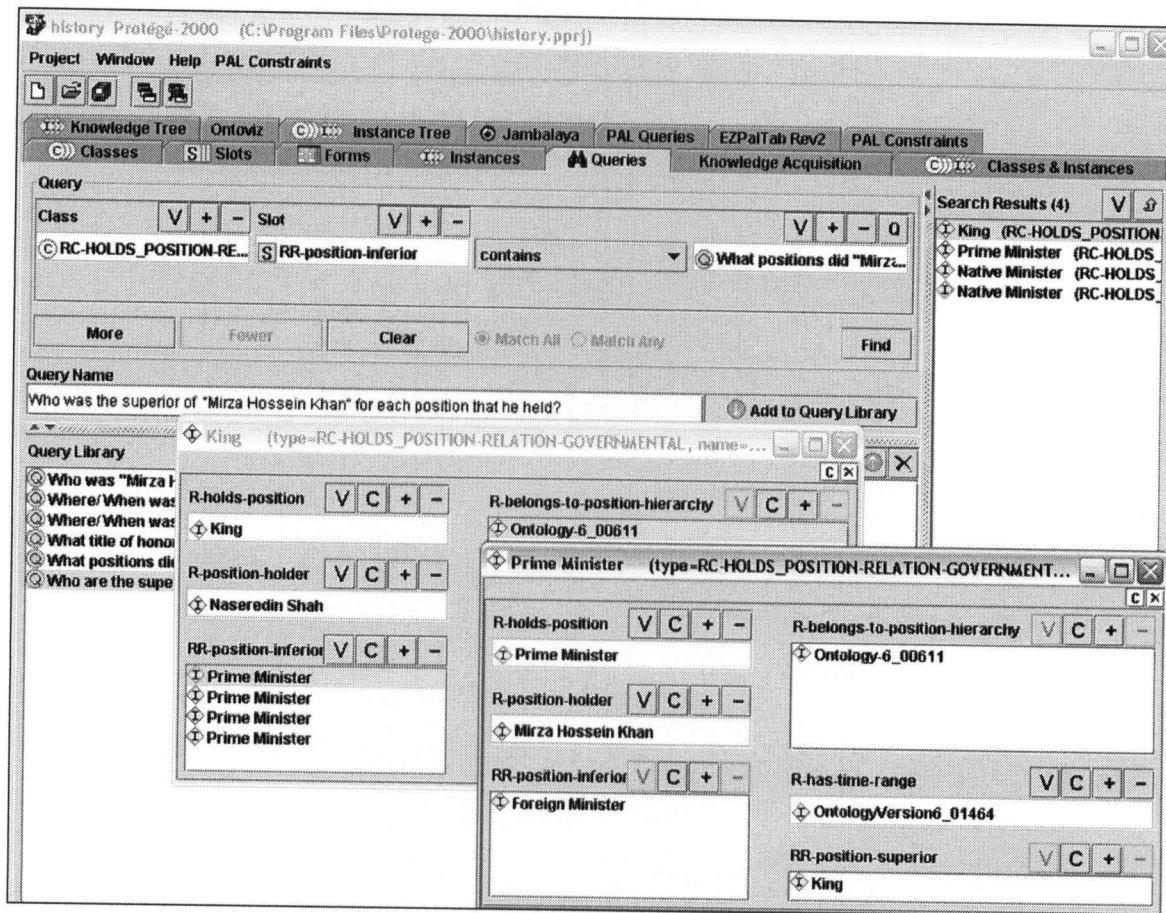


Figure 4-41 Results for Competency Question #5

Question 5: Who was the superior of "Mirza Hossein Khan" for each position that he held?

This figure illustrates the results for the query looking for people who were the superior to "Mirza Hossein Khan" for each position that he held (top-right pane). As a particular example the two pop-up windows in the middle show that king "Nasseredin Shah" was superior to Prime Minister "Mirza Hossein Khan" at a certain time interval. It also demonstrates that these two positions belong to the same position hierarchy "ontolog_6_00611". This hierarchy will be explored in the next figure.

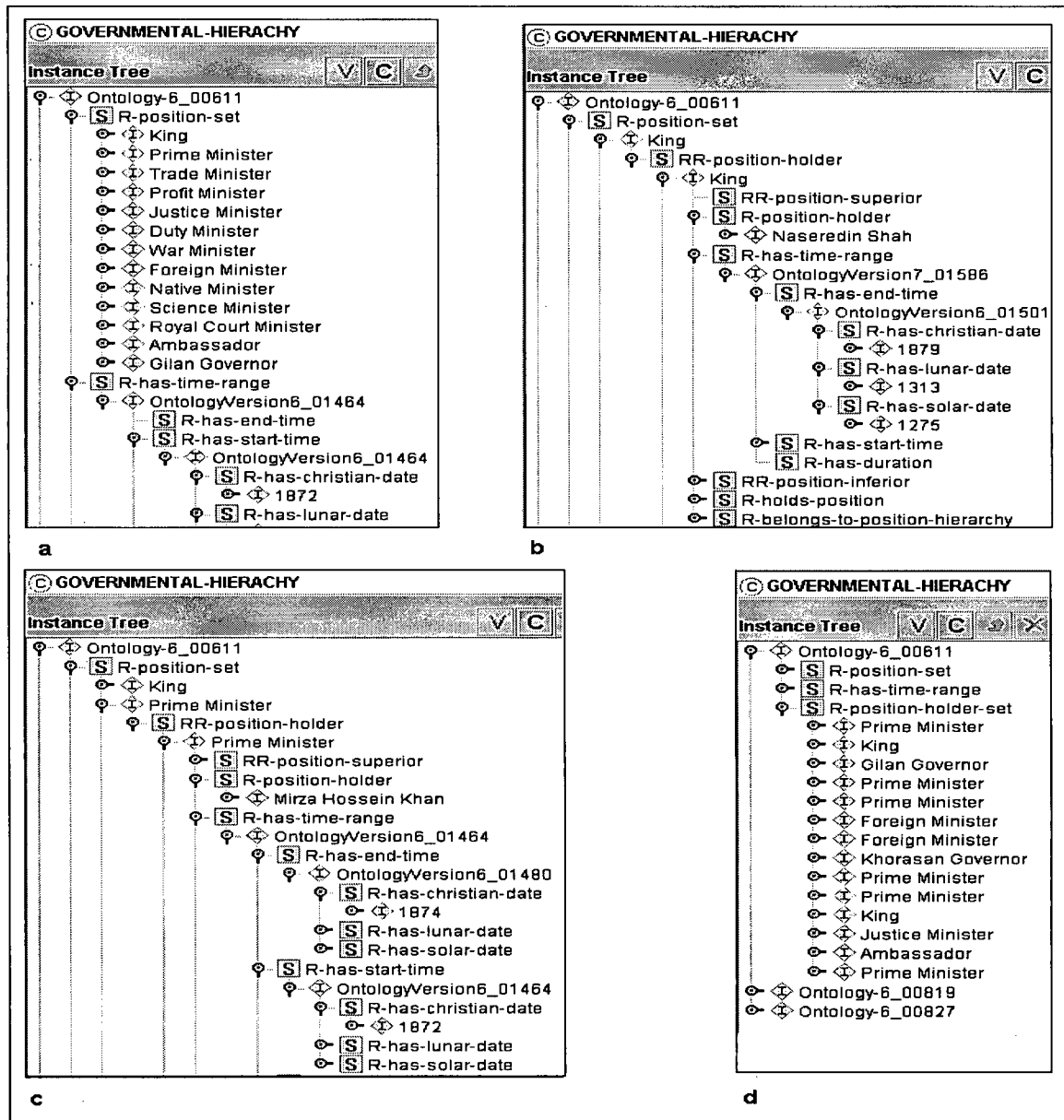


Figure 4-42 Results for Competency Question #6

Question 6: How was the position hierarchy organized at this time?

This figure illustrates the organization of the position hierarchy at the time that “Mirza Hossein Khan” held the position “Prime Minister”. Instance tree (a) shows the hierarchy of positions starting at year 1872. This hierarchy is organized based on the importance of the positions. Every position has a ranking specified by a tag (not shown in this figure). Instance tree (b) illustrates one of the instances of position “king” which was held by “Naseredin Shad” who at that time was the superior of “Mirza Hossein Khan” when he was the Prime minister. As it can be noticed he became the king in 1879 which means that another person held this position before him. Instance tree (c) shows the time during which “Mirza Hossein Khan” held the position of Prime Minister (1872-1874). There are other people who held this position after him. Instance tree (d) demonstrates all the positions that were assigned during the time that this hierarchy was valid. For example the position “Prime Minister” has been assigned to 5 people during this time. During this same time range, there were two kings.

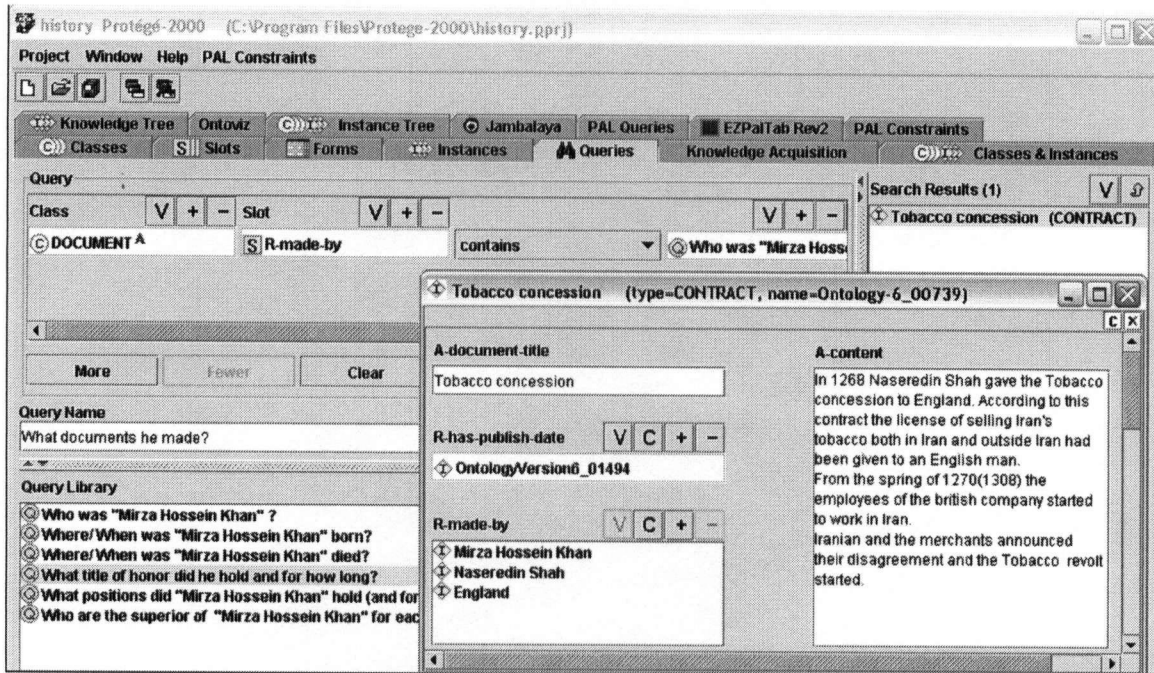


Figure 4-43 Results for Competency Question #7

Question 7: What documents did he make?

This figure illustrates the results obtained from query for the documents made by “Mirza Hossein Khan”. The right side pane shows the only document made by him. The type of the document is a contract and we also can find out the time when it was written and the other agents (people, groups or countries) who were involved in creating this document in the pop-up window.

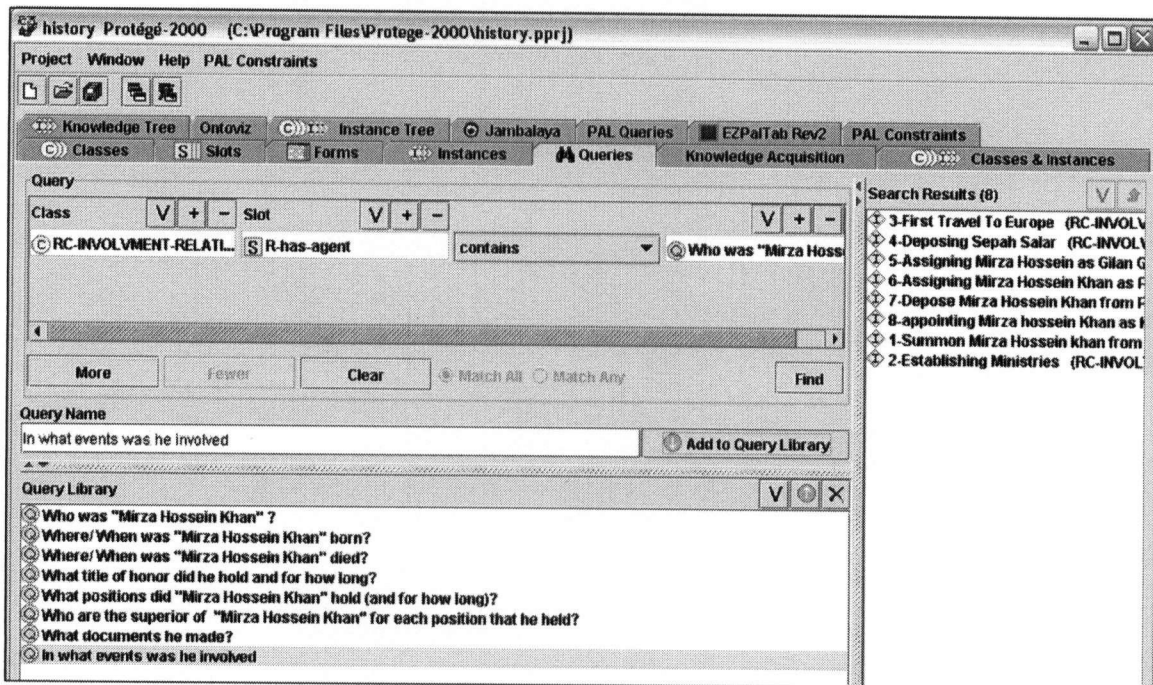


Figure 4-44 Results for Competency Question #8

Question 8: In what events was he involved?

The results of the search for the events where "Mirza Hossein Khan" participated are displayed in the right pane of this figure. By selecting any of the instances of the events listed, one can get additional information related to that particular event. This includes a brief description, people involved, time range, consequences, and documents related to that particular event.

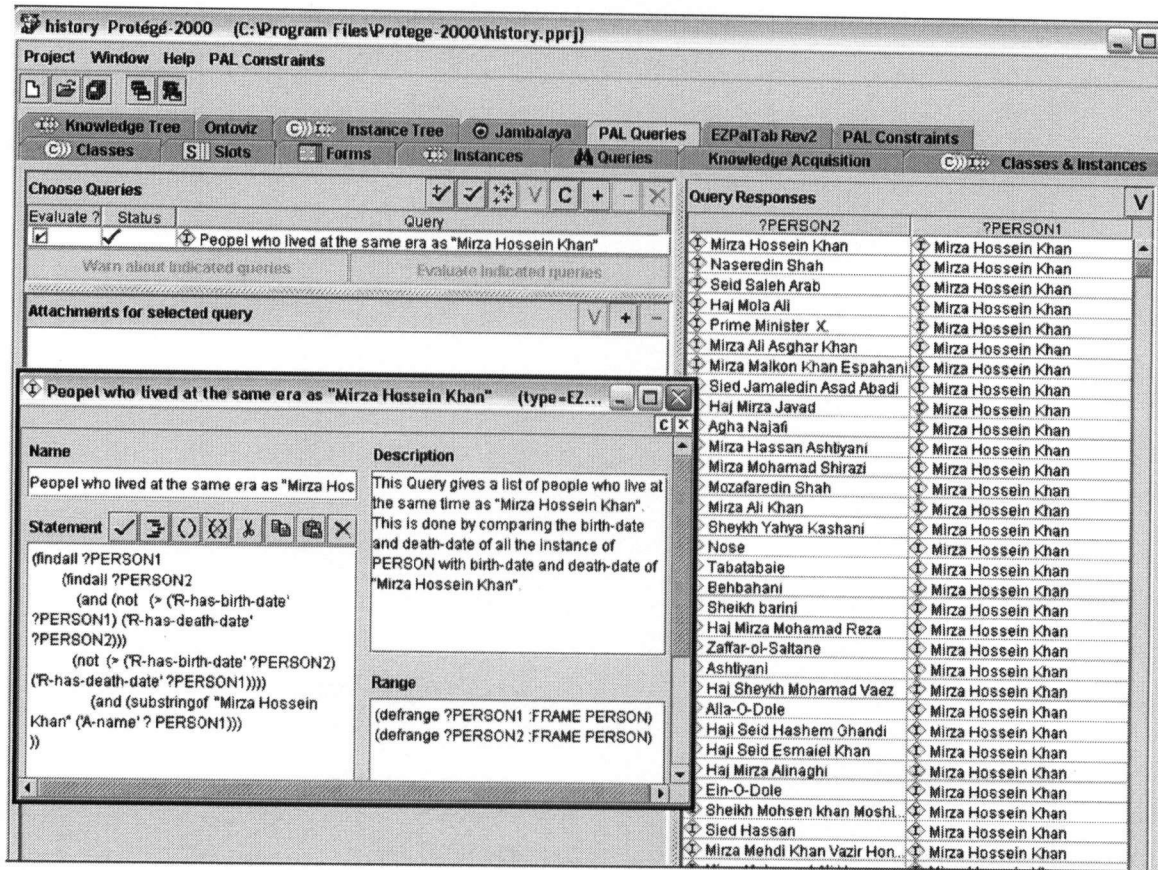


Figure 4-45 Results for Competency Question #9

Question 9: Who else lived at the same era as "Mirza Hossein Khan"?

This figure illustrates the use of the PAL QUERY engine in Protégé-2000 to search for instances of people who lived in the same era as Mirza Hossein Khan. In order to do this we compare the birth-date and death-date of this person with every other person in the domain. The outlined window in the middle shows our PAL Query statement and the right side pane demonstrates the results obtained with this search. This query could be extended and refined to only include those persons in a specific area or with a specific relation to Mirza Hossein Khan.

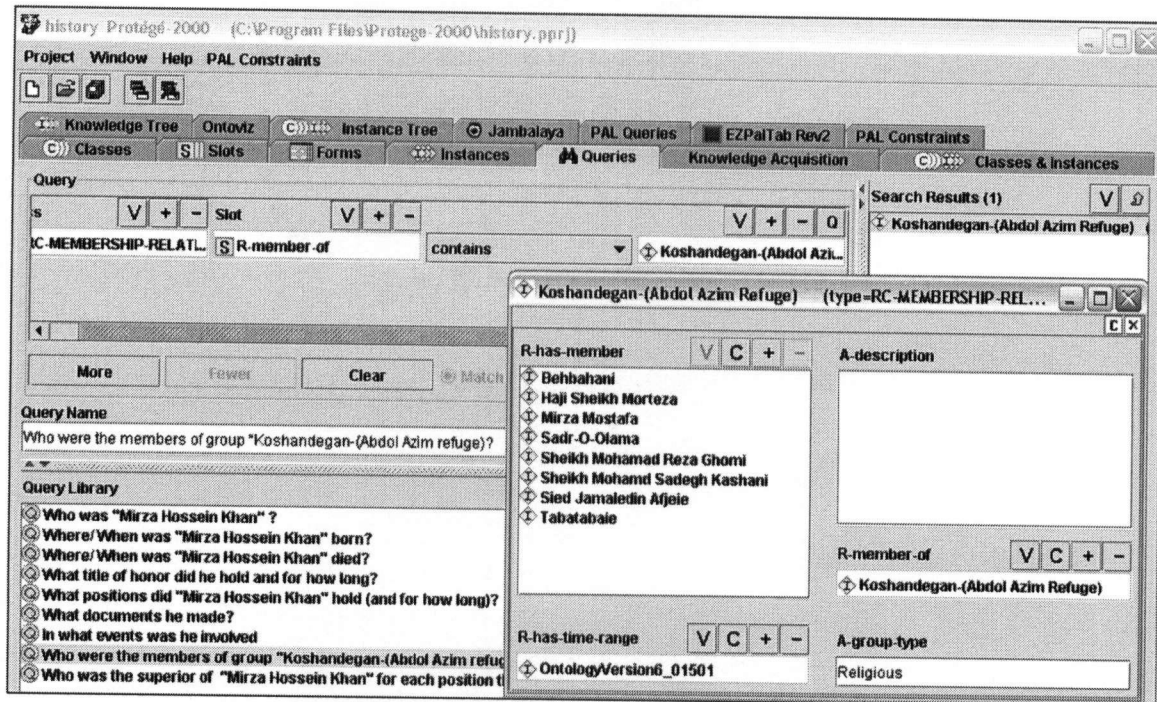


Figure 4-46 Results for Competency Question #10

Question 10: Who were the members of group “Koshandegan-(Abdol-Azim refuge)”?

This figure illustrates the results (right pane) obtained from the query stated above. By selecting any of the results, a pop up window (right side of figure) shows the members of the group, the group’s name, and the time interval during which this group exists.

Following a similar approach, each of the competency questions our model was designed to answer, were queried and tested.

5 Conclusions and Future Work

5.1 Summary of thesis and results

In this work we confronted the limitations of traditional electronic documents. In particular we were interested in capturing the semantic of a historical document in a manner that allows the user functionality beyond keyword search and allows reuse and sharing of the captured information.

We selected to use an ontological approach to represent the information found in the book: *the history of Iranian Constitution*. Ontologies are often used to represent the semantic of the information in a domain by providing a consensual agreement amongst members of a community of interest. Ontologies not only allow representing hierarchical structures but can also be used to capture the relationships amongst concepts within a domain. This facilitates representing dynamic hierarchical structures such as governmental positions and their relations as well as geopolitical place hierarchies commonly found in a history domain. Another motivation for using ontologies is that they allow easy reuse and sharing of the knowledge.

After a review of available ontology development environments, we selected Protégé-2000 to formalize and instantiate our ontology. Our selection was based on the tool's expressiveness, flexibility, customizability, scalability, extensibility, and usability. Additionally, this tool provides us with a series of facilities to test and evaluate our model. Protégé supports simple and complex queries and includes a couple of different graphical visualization plug-ins that allow visualizing and analyzing hierarchical structures contained within the ontology.

The ontology was later evaluated by utilizing a set of competency questions and motivating scenarios we had defined previous to our implementation. The competency questions are the questions that the ontology must provide answer for and the motivating scenarios are the situations that the ontology should provide support for. The created model was successful in answering the questions as well as providing support for the chosen scenarios.

5.2 Conclusions

Our main purpose with this work was to examine the possibility of representing the knowledge found in a historical document in a manner that allows further computational manipulation than that what is provided by text or HTML documents. In order to do this, we require a model where the representation of people, places, events and their relations are captured and understood at a computational level. This representation allows users to pose questions to the system that go beyond what is attainable by the currently available keyword searches. It was of utmost importance to us to allow this information to be reused and shared amongst users as well.

Using an ontology, we extracted and represented the knowledge from the book "*The History of the Iranian Constitution*." Our implementation allowed us to get an overview of the general concepts in this book, relationships amongst these concepts and provided us with different methods for visualizing dynamic hierarchical structures of both governmental positions and geopolitical interdependencies. The user can get an insight into what happens in any place, in relation to any person in the book at any time. Additionally, this model captures the changes that these relations undergo through out time (dynamicity). The temporal aspect of the knowledge we captured proved to be useful in making our representation more accurate and realistic.

5.3 Future Work

Due to time limitations and the scope of the proposed project, we were unable to develop every desirable extension to this work. In this section we present a couple of ideas for possible future work in this area.

In order to facilitate the utilization of models such as the one developed here, we require to have applications that facilitate interacting with this information. As examples of such applications we might mention:

- We would like to develop easy, intuitive interfaces to both access and query the information in the model. Developing such interfaces in a manner that allows people with different interests and backgrounds to access and use this information is a challenge left for future work.

- Implementing a web-based representation of our ontology using schemas such as RDF or OWL. This facilitates reuse and sharing of this knowledge, and allows a greater audience for our information.

Given our time constraints, we were unable to explore every possible research area related to our work. Additional areas of research we would like to explore include:

- We based our design and ideas on information available in research papers and other articles referring to the history domain. In future work, we would additionally like to consult with historians and other domain experts to consider the usability issues and to develop a set of standards for our taxonomy.
- In further work, we would like to design a guideline that establishes a clear path to follow for people interested in extracting and representing this kind of knowledge from historical text documents. As an example of this, we would like to explore the possibility of a joint research project with museum historians to facilitate capturing and representing the knowledge found in historical documents and artifacts.
- Another area we would like to explore is designing and utilizing applications that automatically extract an ontological representation from text documents. If this process could be automated, at least to some degree, a wealth of historical information could be captured and represented in an efficient manner.

Bibliography

AHDS "Arts and Humanities Data Services" <http://ahds.ac.uk/>.

Alani, H., Jones, C. and Tudhope, D. (2000). "Ontology-Driven Geographical Information Retrieval." *Proceedings 1st International Conference on Geographic Information Science (GIScience)*.

Alani, H., Kim, S., Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H. and Shadbolt, N. R. (2003). "Automatic ontology-based knowledge extraction from web documents." *IEEE Intelligent Systems*, 18(1), 14-21.

Beck, H. and Pinto, H. S. (2003). "Overview of Approach, Methodologies, Standards, and Tools for Ontologies." *The Agricultural Ontology Service (UN FAO)*.

Blazquez, M., Lopez, M. F., Perez, A. G. and Juristo, N. (1998). "Building Ontologies at the Knowledge Level Using the Ontology Design Environment." *In Proceedings of the 11th Knowledge Acquisition Workshop, KAW98*, Banff, Canada.

Borst, W. N. (1997). "Construction of Engineering Ontologies for Knowledge Sharing and Reuse. PhD Thesis." *Centre for Telematics and Information Technology*. Enschede, Netherlands, Twente University.

Burchardt, J. (2001). "Archiving the Internet - how to collect historical sources for the future." *International Conference of the Association for History and Computing*, Poznan, Poland.

Chesnutt, D. (1995). The Model Edition Partnership, historical editions in the digital age. *Digital Library Magazine*.

CYC Ontology <http://www.cyc.com/cyc/openencyc/overview>.

DAML (DARPA Agent Markup Language) Ontology Library
<http://www.daml.org/ontologies/>.

Duineveld, A. J., Stoter, R., Weiden, M. R., Kenepa, B. and Benjamins, V. R. (2000). "WonderTools? A comparative study of ontological engineering tools." *International Journal of Human-Computer Studies*, 52(6), 1111-1133.

Enterprise Ontology
<http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html#terms>.

- EuroConference (2000) *Geographical Domain and Geographical Information Systems, EuroConference on Ontology and Epistemology for Spatial Data Standards*
<http://www.geoinfo.tuwien.ac.at/events/Euresco2000/gdgis.htm> La Londe-les-Maures, France.
- Fadel, F. G., Fox, M. S. and Gruninger, M. (1994). "A Generic Enterprise Resource Ontology." *Proceedings of the third IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*: Morgantown, West Virginia.
- Fox, M. S. (1992). "The TOVE Project - Towards a Common-Sense Model of the Enterprise." *Lecture Notes in Artificial Intelligence*, 604, 25-34.
- Fox, M. S., Barbuceanu, M. and Gruninger, M. (1996). "An organization ontology for enterprise modeling: Preliminary concepts for linking structure and behavior." *Computers in Industry*, 29(1-2), 123-134.
- Fox, M. S. and Gruninger, M. (1994). "Ontologies for Enterprise Integration." *Proceedings of the 2nd Conference on Cooperative Information Systems*, Toronto, Ontario.
- Fox, M. S. and Gruninger, M. (1997). "On Ontologies and Enterprise Modelling." *International Conference on Enterprise Integration Modelling Technology 97*, Springer, Verlag.
- Fox, M. S. and Gruninger, M. (1998). "Enterprise modeling." *Ai Magazine*, 19(3), 109-121.
- Gennari, J. H., Musen, M. A., Ferguson, R. W., Grosso, W. E., Crubezy, M., Eriksson, H., Noy, N. F. and Tu, S. W. (2003). "The evolution of Protégé: an environment for knowledge-based systems development." *International Journal of Human-Computer Studies*, 58(1), 89-123.
- Grosso, W. E., Eriksson, H., Ferguson, R. W., Gennari, J. H., Tu, S. W. and Musen, M. A. (1999). "Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000)." .
- Gruber, T. R. (1993). "A Translation Approach to Portable Ontology Specifications." *Knowledge Acquisition*, 5(2), 199-220.
- Gruber, T. R. (1995). "Toward principles for the design of ontologies used for knowledge sharing." *International Journal of Human-Computer Studies*, 43(5-6), 907-928.

- Gruninger, M. and Fox, M. S. (1994). "The Design and Evaluation of Ontologies for Enterprise Modelling." *Workshop on Implemented Ontologies, European Workshop on Artificial Intelligence*: Amsterdam, NL.
- Gruninger, M. and Fox, M. S. (1994). "The Role of Competency Questions in Enterprise Engineering." *IFIP WG5.7 Workshop on Benchmarking-Theory and Practice*: Trondheim, Norway.
- Gruninger, M. and Fox, M. S. (1995). "Methodology for the Design and Evaluation of Ontologies." *Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95* (Montreal).
- Guarino, N. (1997). "Understanding, building and using ontologies." *International Journal of Human-Computer Studies*, 46(2-3), 293-310.
- Guarino, N. and Giarretta, P. (1995). "Ontologies and Knowledge Bases: Toward a Terminological Clarification." *Towards Very Large Knowledge Bases*, In N.J.I Mars (ed.): Amsterdam, 25-32.
- Guarino, N. and Pinto, J. A. (1995). "A Theory of Complex Actions for Enterprise Modelling." *Working Notes AA AI Spring Symposium Series: Extending Theories of Action: Formal Theory and Practical Applications*: Stanford.
- HDS "History Data Service" <http://hds.essex.ac.uk/>.
- Hockey, S. (1999). "Making Technology Work for Scholarship: Investing in the Data." *Technology and Scholarly Communication*, 17-36.
- Hockey, S. M. (2000). "Electronic texts in the humanities: principles and practice." Oxford; New York. *Oxford University Press*.
- Islam, A. S., Bermudez, L. and Piasecki, M. (2003) Ontology for Geographic Information-Metadata (ISO-19115) <http://loki.cae.drexel.edu/~wbs/ontology/>.
- Jarrar, M. (2000). "Methodologies for extracting ontologies from the web." *STAR Lab*. Brussels, Vrije University.
- Kim, H. M., Fox, M. S. and Gruninger, M. (1999). "An ontology for quality management - enabling quality problem identification and tracing." *BT Technology Journal*, 17(4), 131-140.
- Knublauch, H. (2003). "An AI Tool for Real World: Knowledge Modeling with Protege." *Java World*.

- Lakoff, G. (1987). "Women, fire, and dangerous things: what categories reveal about the mind." *University of Chicago Press*.
- Lopez, M. F. and Perez, A. G. (2002). "Overview and Analysis of Methodologies for Building Ontologies." *Knowledge Engineering Review*, 17(2), 129-156.
- Lopez, M. F., Perez, A. G. and Juristo, N. (1997). "METHONTOLOGY: From Ontological Art Towards Ontological Engineering." *Workshop on Ontological Engineering. Spring Symposium Series: Stanford, USA*.
- Lopez, M. F., Perez, A. G., Sierra, J. P. and Sierra, A. P. (1999). "Building a Chemical Ontology Using Methontology and the Ontology Design Environment." *IEEE Intelligent Systems & Their Applications*, 14(1), 37-46.
- Mahesh, K. (1996). "Ontology development for machine translation: Ideology and methodology." *Computer Research Laboratory, New Mexico State University*.
- Mark, D. M., Skupin, A. and Smith, B. (2001). "Features, Objects, and other Things: Ontological Distinctions in the Geographic Domain." *Conference On Spatial Information Theory (COSIT), Morro Bay, CA, USA*.
- McGuinness, D. L. (2002). "Ontologies Come of Age." *Spinning the Semantic Web: bringing the World Wide Web to Its Full Potential*, MIT Press.
- MEP (2003) Model Editions Partnership, Historical Edition in the Digital Age.
<http://mep.cla.sc.edu/mepinfo/mep-info.html>, University of South California.
- Merriam Webster Online Dictionary, <http://www.m-w.com>.
- Noy, N. F., Fergerson, R. W. and Musen, M. A. (2000). "The knowledge model of Protege-2000: Combining interoperability and flexibility." *Knowledge Engineering and Knowledge Management, Proceedings*, 1937, 17-32.
- Noy, N. F. and Hafner, C. D. (1997). "The state of the art in ontology design - A survey and comparative review." *Ai Magazine*, 18(3), 53-74.
- Noy, N. F. and McGuinness, D. L. (2001). "Ontology Development 101: a Guide to Creating Your First Ontology." Stanford, CA, *Stanford University*.
- Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R. W. and Musen, M. A. (2001). "Creating Semantic Web contents with Protege-2000." *IEEE Intelligent Systems & Their Applications*, 16(2), 60-71.
- OKBC <http://www.ksl.stanford.edu/software/OKBC/>.

- OKBC (1998) OKBC Specification Document <http://www.ai.sri.com/~okbc/spec.html>.
- OntoEdit <http://www.ontoprise.de/products/ontoedit>.
- Ontolingua <http://www.ksl.stanford.edu/software/ontolingua/>, Knowledge System Laboratory, Stanford University.
- Oxford Dictionary <http://dictionary.oed.com/>.
- PAL <http://protege.stanford.edu/plugins/pal/pal-documentation/index.html>.
- Palmer, S. B. (2001) The Semantic Web: An Introduction.
<http://infomesh.net/2001/swintro/>.
- Perez, A. G. (1994a). "From Knowledge Based Systems to Knowledge Sharing Technology: Evaluation and Assessment." Knowledge Systems Laboratory. *Stanford University*.
- Perez, A. G. (1994b). "Some Ideas and Examples to Evaluate Ontologies." Knowledge System Laboratory. *Stanford University*.
- Perez, A. G., Lopez, M. F. and De Vicente, A. (1996). "Toward a Method to Conceptualize Domain Ontologies." *Workshop on ontological Engineering. ECAI'96*, Budapest, Hungary. 41-51.
- Pinto, S., Perez, A. G. and Martins, J. P. (1999). "Some issues on ontology integration." *Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends. (IJCAI99)*. Stockholm, Sweden.
- Protege-2000/OWL (2003) Protege-2000 Plug-ins
<http://protege.stanford.edu/plugins/owl/>.
- Protege-2000/RDF (2001) Using Protege-2000 to Edit RDF
<http://protege.stanford.edu/protege-rdf/protege-rdf.html#differences>.
- Protégé <http://protege.stanford.edu/index.html>.
- Relation Tab (Protege-2000)
<http://protege.stanford.edu/plugins/relationstab/RelationsTab.htm>.
- Robertson, B. (2001). "The historical event markup and linking project." *International Conference of the Association for History and Computing*, Poznan, Poland.
- Robertson, B. (2003) The historical event markup and linking project (HEML).
<http://heml.mta.ca/heml-cocoon/>.

- Sowa, J. F. (2000). "Ontology, metadata, and semiotics." *Conceptual Structures: Logical, Linguistic, and Computational Issues, Proceedings, 1867*, 55-81.
- Sowa, J. F. (2000a). "Knowledge Representation: Logical, Philosophical, and Computational Foundation." Pacific Grove, CA. *Brooks Cole Publishing Co.*
- Spyns, P., Meersman, R. and Jarrar, M. (2002). "Data Modelling versus Ontology Engineering." *Sigmod Record*, 31(4), 12-17.
- Stader, J. (1996). "Results of the Enterprise Project." *Proceedings of Expert Systems, 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems*, Cambridge, UK.
- Studer, R., Benjamins, V. R. and Fensel, D. (1998). "Knowledge Engineering: Principles and methods." *Data & Knowledge Engineering*, 25(1-2), 161-197.
- SUMO (Suggested Upper Merged Ontology) <http://ontology.teknowledge.com/>.
- Swartout, B., Patil, R., Knight, K. and Russ, T. (1996). "Toward Distributed Use of Large-Scale Ontologies." *Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96)*, Banff, Canada.
- Swartz, A. (2002) The Semantic Web in Breadth. <http://logicerror.com/semanticWeb-long>.
- TOVE Project <http://www.eil.utoronto.ca/enterprise-modelling/tove/>.
- UMLS "Unified Medical Language System" <http://www.nlm.nih.gov/research/umls/>.
- UPM (2002). "Deliverable 1.3: A Survey on Ontology Tools." http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-3.pdf.
- Uschold, M. (1996). "Building Ontologies: Toward a Unified Methodology." *16th Annual Conf. of British Computer Society Specialist Group on Expert Systems*, Cambridge, UK.
- Uschold, M. and Gruninger, M. (1996). "Ontologies: Principles, methods and applications." *Knowledge Engineering Review*, 11(2), 93-136.
- Uschold, M. and King, M. (1995). "Towards Methodology for Building Ontologies." *Workshop on Basic Ontological Issues in Knowledge Sharing*, held in conjunction with IJCAI-95.
- Uschold, M., King, M., Moralee, S. and Zorgios, Y. (1998). "The Enterprise Ontology." *Knowledge Engineering Review*, 13(1), 31-89.

- vanHeijst, G., Schreiber, A. T. and Wielinga, B. J. (1997). "Using explicit ontologies in KBS development." *International Journal of Human-Computer Studies*, 46(2-3), 183-292.
- W3C/RDF (2000) Resource Description Framework, World-Wide Web Consortium
<http://www.w3.org/RDF/>.
- WebODE <http://delicias.dia.fi.upm.es/webODE/>.
- WebOnto (2002). "Deliverable 1.3: A Survey on Ontology Tools." http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-3.pdf.
- Webster Merriam Webster Online Dictionary, <http://www.m-w.com>.
- Weiss, S. M., Kulikowski, C. A., Amarel, S. and Safir, A. (1978). "Model-Based Method for Computer-Aided Medical Decision-Making." *Artificial Intelligence*, 11(1-2), 145-172.
- WordNet <http://www.cogsci.princeton.edu/~wn/>.
- Zhou, Q. and Fikes, R. (2002). "A Reusable Time Ontology." *Proceeding of the Ontologies for the Semantic Web Workshop, AAAI National Conference*.