

Development of ontology based on semantic web

For Sudanese's national pension fund

Chapter one

Introduction

1.1 Background

The pension system represents the basis of the to the social security system for state employees when they are exposed to the risks of old age, disability and death or loss of a source of income for any reason and their families.

The Fund has sought recently to expand the social security, include social assistance that fit the circumstances to employer's pensions, such as provision of social services such as health services - assistance in the field of education expenses - college education - social support public - special programs in national events and holidays.

- The role of the (S-NPF)
 - Management of the pension system under the law of public service pensions for the year 1992 or any superseding law
 - Management of pension systems and government parts, institutions and public sector companies established by special laws entrusted to him to manage it, the collection of the government's commitment and the collection of the pension deduction for any government employees for his pension .
 - Investment Fund resources and work on development and development.
 - Borrowing from local banks and global investment purposes and accept grants and donations and bequests and insurance funds invested

- Development Services pension and late periodic review of pensions and benefits in order to improve

- The conclusion of contracts and agreements

- Objective:

Raise the intellectual potential of the nation and the financial system to build an effective and comprehensive and unified social security.

- Revelation:

To be the leading institution of social security institutions of the Arab and African and offers a balanced level of features and benefits and has the financial capability and to continue in the long term and is managed and according to the standards of transparency and global governance, and be given its activities and services with the consent and acceptance of the beneficiaries and associates and its employees.

- Message:

1 - The Fund becomes a reference for social security institutions of Sudan.

2 - To develop methods and tools, including his knowledge can provide sophisticated insurance service

3 – To seeks to bridge the gap between the pension and lost wages (due to retirement) current retirees down sufficiently reduce

4 – To Publishes the social security system through the expansion of the coverage of the dangers of social Outs society towards it through social assistance programs and maternity care and family [1].

1.2 Problem Statement

Managing and sharing information still a problem facing (S-NPF) that need to share domain information define the concepts and relationships (also referred to as “terms”) used to describe and represent the terms for

(S-NPF) to be understand for employees and pensioner

- Managing have been a big challenge
- Lack of awareness among Pensioners
- It has been a challenge among working in managing information in due its complexity.

1.3 Research Objectives

The main objectives of this research are:

- To build (S-NPF) management ontology.
- To design a framework of self-managed (S-NPF).
- To validate the designed -Managed (S-NPF) by sending queries to ontology system

1.4 Significance of the study

An automated management (S-NPF) manager can be an effective solution, complement the works on (S-NPF) off line systems, fills a gap on existing knowledge – developing a new ontological system & contributes to ongoing research for effective management.

1.5 Research Methodology

This research shall be carried out in five phases as illustrated in the following research methodology framework by flowing steps: [2].

- Step 1 – LR and Analysis of an Ontology Design for (S-NPF) Self Management System Modeling
- Step 2 – Pre & Post-Survey of an Ontology Design for (S-NPF) Self Management System Modeling
- Step 3 – Formulate the Ontology Design for (S-NPF) Self Management System

- Step 4 – Develop the System
 - Step 4-1 – Requirements
 - Step 4-2 – Analysis
 - Step 4-3 – Design and implementation
 - Step 4-4 – Verification & Validation
- Step 5 – Evaluation

1.5.1 Step 1 – LR and Analysis of (S-NPF) Self Management System

This phase involves evaluation of existing Administrative structure of the National Pension Fund Management System and literature reviews on academic journals, books, and information search on groupware tools.

1.5.2 Step 2 -Conduct Pre-Survey on (S-NPF) Self Management System

Questionnaire survey shall be used to evaluate Self Management models used by (S-NPF). The survey shall cover collaborative activities and knowledge required within the (S-NPF) Self Management processes. Also, usage of collaborative tools and knowledge sharing tools between the system and the users shall be identified.

1.5.3 Step 3 -Formulate (S-NPF) Self Management System

The next step is to formulate new model of applying techniques for (S-NPF) collaborative and Self Management System based on earlier literature reviews and pre-survey results, areas of concerns are noted

This stage is in progress.

1.5.4 Step 4 -Develop the System

The system development shall include the following steps:

- ✓ Requirements
- ✓ Analysis
- ✓ Design and Implementation

✓ Verification and Validation

- **Step 4-1 –Requirements**

Our entire system requirements will be identified in this stage.

- **Step 4-2 –Analysis**

Our entire system requirements will be analyzed in this stage.

- **Step 4-3 - Design and Implementation**

To develop the system. The platform and development tool shall be determined at a later stage.

- **Step 4-4 - Verification and Validation**

This is an iterative process, whereby the tools are tested and reworks and enhancements are carried out. We expect several builds are required to stabilize the application.

1.5.5 Step 5 -Evaluation

In this stage, we shall identify the pilot site for implementation, conduct training and implement the proposed system.

1.6 Expected Results

- The system will support Better control measures in improving the management of business processes and achieve this through the standard work and provide good attempts.
- Improve customer service and the ease and simplification of procedures leading to a better outlook on the level of response
- Improve work processes to focus on business processes leads to the sequence and the oversimplification

Chapter Two

Theoretical Background and Literature Review

The semantic Web will provide a semantic meaning to the current Web, so it will be easier (for people and machines) to work with this data. That can be done by uses of ontology's, [3]. this chapter explains what semantic web is? , Definitions of Ontology and some ontology made be for.

2.1 Ontology and Semantic Web Overview :

The Semantic Web is a network or mesh of information connected in such a way that its easily executable by machines, on a large scale globally. We can also define it as being a proficient way of showing data on the World Wide Web or as a worldwide connected database. The current web is the collection of documents and computers are phrasing around these documents. The end users search for documents by asking the questions from search engines. The computer understand the HTML code literally word by word by reading single words and show the results regarding to it. But it cannot understand the meaning behind those documents which the users parsing around.

The Semantic Web extends the current World Wide Web by adding facilities for machine-processable descriptions of meaning. In order for semantic exchanges of information to take place, there needs to be agreement on how to model meaning. Ontologies are the mechanism for representing formal and shared domain descriptions (Geroimenko and Chen, 2002). Ontologies help both people and machines communicate more effectively by providing a common definition of a domain (van Harmelen et al., 2001). Ontologies

can be generally defined as a “specification of a conceptualization” (Gruber, 1993). Ontologies are metadata that provide a controlled vocabulary of terms, where each term is defined explicitly so that it is machine-processable. Ontologies facilitate machine processing by allowing information to be annotated with metadata so that meaning can be determined. Ontologies are expected to play a central role in the development of the Semantic Web, and will be used for many different purposes, such as querying, presentation, and navigation (Fluit et al., 2003). Ontologies are similar to taxonomies in that classification is provided. However, ontologies also include information about the relationships among terms (concepts), which provides the basis for semantic reasoning (Seeling and Becks, 2003) [4].

2.1.1 Definitions of Ontology:

- Ontology is about the exact description of things and their relationships.

For the web, ontology is about the exact description of web information and relationships between web information.

- Ontology is an organizational system designed to categorize and help explain the relationships between various concepts of science in the same area of knowledge and research.

- **In Computer Science:**

Ontology is a formal explicit specification of a shared conceptualization
Of a domain.

- **Ontologies are formal:**

- Meaning of ontology is unambiguous, avoids misunderstanding
- Specification using formal language enables reasoning
- Specification may hamper consensus

Ontologies use explicit specifications:

- To make domain assumptions explicit for reasoning
- **To support understanding of domain.**

A specification consisting of

- classes
- relations between classes
- individuals
- axioms

Ontology describes the concepts in the domain and also the relationships that hold between those concepts. Different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C). There are several definitions of ontology each and one differ from each other. Another definition can be “ontology is a formal explicit description of concepts in a domain of discourse.

Classes which are sometimes called concepts, properties of these classes are call slots or roles, while restrictions on slots are called facets. When ontology is together with the instances it creates knowledge base.

Ontology is truly based on the representation of classes. So we can say that classes gives concept to ontology.

Since the beginning of the 1990s ontology became an admired research topic investigated by quite a few AI research communities including knowledge engineering, natural language processing, and knowledge representation. More recently, the notion of ontology is also becoming extensive in the fields like intelligent information integration, cooperative information

systems, information retrieval, electronic commerce, and knowledge management.

In computer science and information science, ontology is a formal representation of the knowledge by a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to describe the Domain

2.1.2 Common components of ontology's include:

- **Individuals:** instances or objects (the basic or "ground level" objects)
- **Classes:** sets, collections, concepts, types of objects, or kinds of things.
- **Attributes:** aspects, properties, features, characteristics, or parameters that objects (and classes) can have
- **Relations:** ways in which classes and individuals can be related to one another
- **Function terms:** complex structures formed from certain relations that can be used in place of an individual term in a statement
- **Restrictions:** formally stated descriptions of what must be true in order for some assertion to be accepted as input
- **Rules:** statements in the form of an if-then (antecedent-consequent) sentence that describe the logical inferences that can be drawn from an assertion in a particular form
- **Axioms:** assertions (including rules) in a logical form that together comprise the overall theory that the ontology describes in its domain of application. This definition differs from that of "axioms" in generative grammar and formal logic. In these disciplines, axioms include only

statements asserted as a priori knowledge. As used here, "axioms" also include the theory derived from axiomatic statements.

- **Events:** the changing of attributes or relations

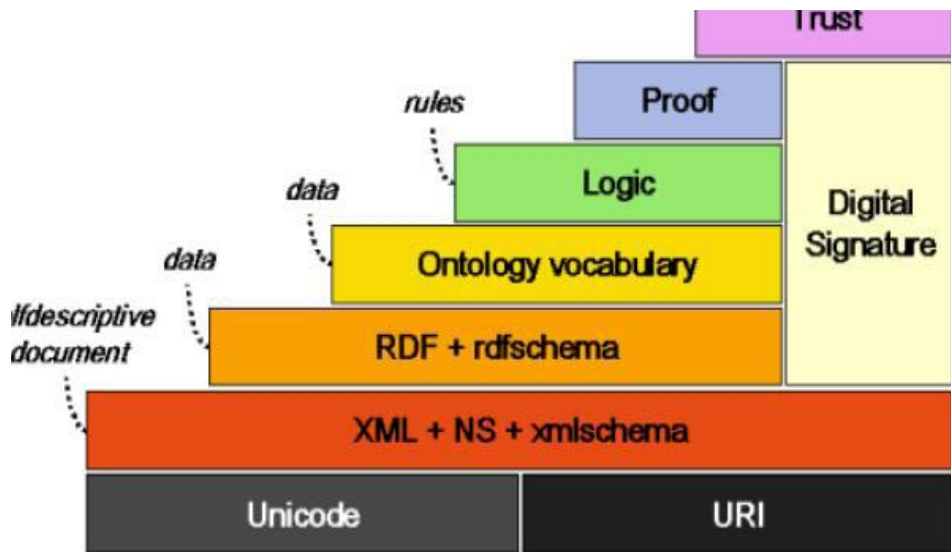
2.1.3 Functions of Ontology

- a. It provides a common background and understanding of a particular domain, or field, of study, and ensures a common ground among those who study the information.
- b. It is a way of organizing concepts, information, and ideas that is meant to be universal within the field, and allows for a common language to be spoken.
- c. It is a structural framework that allows the concepts to be laid out in a way that makes sense.
- d. It helps show the connections and relationships between concepts in a manner that is generally accepted by the field [5].

2.2 layers of ontology

Tim Berners Lee has divided the semantic web into layers which are

- Unicode and URI layer
- XML and XML schema layer
- RDF layer
- Ontology
- Logic
- Proof
- Trust



Semantic Web Layers
 Courtesy: Semantic Web Activity

Figure 2.1: layers of ontology by Tim Berners Lee

Tim Berners Lee when at the first come up with the idea of web 2.0 his vision was not only for human-human communications but also for machine interaction. The contents on web currently are majorly for humans. Let's take an example there is a lot of information on web about:

- Weather
- Airline schedule
- Sports Stats
- TV and Movies Guidelines

Theses information is easily available on the web and can be seen but it's very difficult to use these contents or make it customizable on own website or any other application. To explain it better let's take the case of online calendars it's very easy to see data but very difficult to pull out information

and utilize it on other websites or any portable device. Though Google has done a lot more work on that task to create API with the help of which one can easily pull out information and utilize it anywhere. And that's because of using OWL, RDF, SPARQL and many other new languages. With the help of these new languages the concept of web is change and it's all in the new dimensions of search contents and explaining the web contents e.g. Ontologies can define more briefly than the HTML etc. The Semantic is web is new idea and research going on currently and the purpose behind this idea is to introduce artificial intelligence to the internet where searches are not based just on the phrase match but the meaning behind that searches. Many organizations are attempting to make the Web computer-friendly via Web services [6].

2.3 Ontology language

2.3.1 XML

XML is a machine language which is abbreviated as Extensible mark-up language and it is used for documents like HTML etc.

Browsers compatibility in the past has been a great challenged. The document created by the user has a lot of issues with the other browsers for example the user created a document in Internet explorer then the document tested on opera will have different look. WWW (World Wide Web) has now extended the limits of using just few tags of HTML. W3C is also known as World Wide Web consortium is the platform which has been working to promote standards in technology for decades. With the help of XML the user can create self created new tags, new elements in no time. Most of the browsers now a day's support XML

XML is currently a very popular and effective way of exchanging information. XML languages conform to a well defined syntax that is compatible with many parsers which are widely available. XML provides a proficient solution to the syntax problem for data sharing.

XML consists of tags which the user can create and use it for structure of the program and it's different from HTML tags.

XML can identify the type of documents, elements, attributes of those elements and the connection or relationship of those elements and documents.

XML and HTML have quite a lot of differences. HTML is purely used to display or design the web pages. The function of HTML is different from XML. HTML cannot save data while XML is used to store data in the document. So XML has no concern over the design and layout of web content.

XML can also be used as a good communication tool. If there are group of users using same tags in an application to express data then these users can robustly communicate so due to that reason XML is known as easier platform of exchanging information between entities.

How XML does look:

Let's look at some examples of XML text:

```
<?xml version="1.0"?>
<Gift>
<to>Peter</to>
<from>Frank</from>
<wordings>Happy Birthday!</wordings>
<body>Have a Great day</body>
</Gift>
```

At the start of the line there is declaration of the XML and its version. It is necessary to include that part in the XML code. Also we have to note down

that there is no closing tag for the declaration line. So we have to keep in mind that declaration doesn't need closing tag.

2.3.2 RDF

Resource Description Framework, this is the basic framework that the rest of the Semantic Web is based on. On the Semantic Web, information is represented as a set of declarations called statements made up of three parts: subject, predicate, and object. Because of these three parts, statements are also sometimes referred to as triples. The three elements of a statement have meanings that are similar to their meanings in normal English grammar. The subject of a statement is the thing that statement describes, and the predicate describes a relationship between the subject and the object. Let's take an example

Andrew's surname is Perez-Lopez.

Matt knows John.

Ryan works with John.

RDF representation will be like the figure given below.

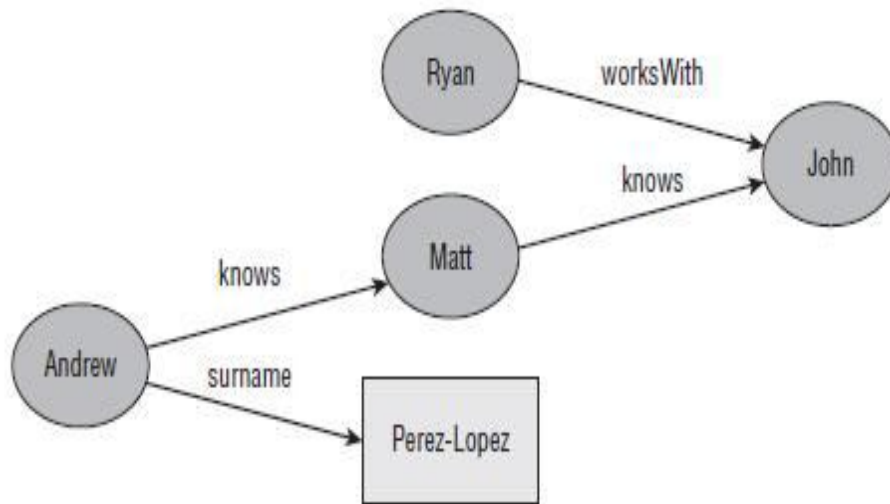


Figure 2.2: RDF representation

2.3.3 RDF schema:

RDFS is the schema language for RDF. RDF Schema extends RDF by introducing a set of distinguished resources into the language. This is related to the way in which a traditional programming language can be extended by defining new language-defined keywords. But there is an important difference: XML parsers can automatically determine whether a particular XML document conforms to a given schema. Other schema languages help us to interpret particular data. For example, a database schema provides header and key information for tables in a relational database. There is neither anything in the table itself to indicate the meaning

RDF and RDF Schema

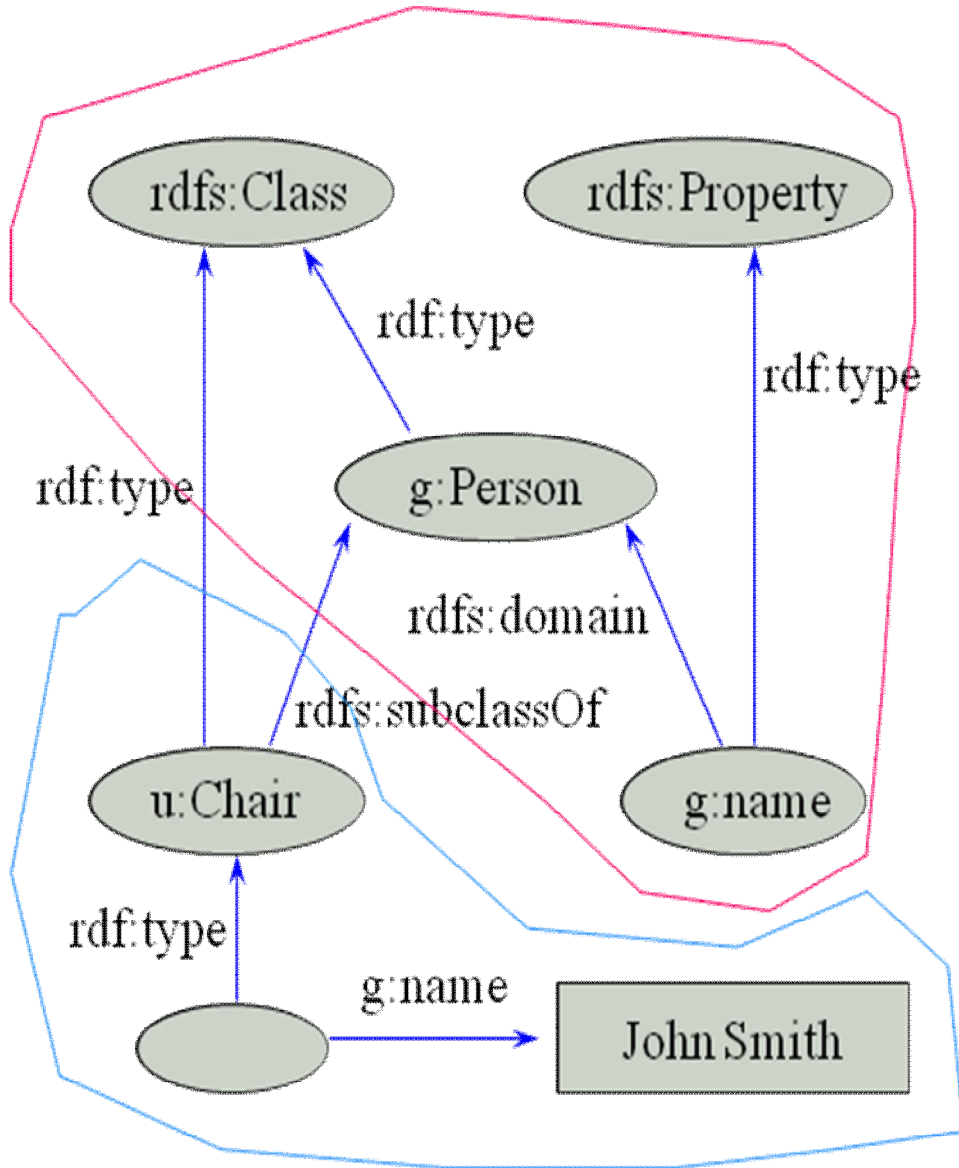


Figure 2.3: RDF and RDF Schema

e.g 1

```
<rdf:RDF
  xmlns:g="http://schema.org/gen"
  xmlns:u="http://schema.org/univ">
  <u:Chair rdf:ID="john">
    <g:name>John Smith</g:name>
  </u:Chair>
</rdf:RDF>
```

e.g 2

```
<rdfs:Property rdf:ID="name">
  <rdfs:domain rdf:resource="Person">
</rdfs:Property>

<rdfs:Class rdf:ID="Chair">
  <rdfs:subclassOf rdf:resource=
    "http://schema.org/gen#Person">
</rdfs:Class>
```

2.3.4 Problems with RDFS

- RDFS too weak to describe resources in sufficient detail, e.g.:

-No localised range and domain constraints

Can't say that the range of hasChild is person when applied to persons and elephant when applied to elephants

– No existence/cardinality constraints

Can't say that all instances of person have a mother that is also a person, or that persons have exactly 2 parents

– No transitive, inverse or symmetrical properties

Can't say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical

- We need RDF terms providing these and other features.

2.4 OWL

The Web Ontology Language (OWL) recommendation is designed to let applications process the content of information (W3C OWL, 2004).

The Web Ontology Language OWL extends RDF and RDFS. Its primary aim is to bring the expressive and reasoning power of description logic to the semantic web. Unfortunately, not everything from RDF can be expressed in DL. For example, the classes of classes are not permitted in the (chosen) DL, and some of the triple expressions would have no sense in DL. That is why OWL can be only syntactic extension of RDF/RDFS (note that RDFS is both syntactic and semantic extension of RDF). To partially overcome this problem, and also to allow layering within OWL, three species of OWL are defined.

2.4.1 The Three Species of OWL

- **OWL Full**

- contains OWL DL and OWL Lite
- only OWL variant which completely contains RDFS
- semantics problematic from a logical point of view
- undecidable
- only partial support by existing software tools

- **OWL DL**

- contains OWL Lite and is contained in OWL Full
- decidable.
- almost complete support by existing software tools

- complexity NExpTime (worst-case)
- **OWL Lite**
- contained in OWL DL and OWL Full
- decidable
- not very expressive
- Complexity Expire (worst-case) [7].

2.5 Previous study

The (S-NPF) system has many different systems to Managing data but there is no ontology made before to define the concepts and terms on semantic web.

2.5.1 Antonia Albani, Jan L.G. Dietz, The Benefit of Enterprise Ontology in Identifying Business Components:

An automated management for Enterprises need to developing an ontological system the following paper discus that .Companies are more than ever participating in so-called value networks while being confronted with an increasing need for collaboration with their business partners. In order to better perform in such value networks information systems supporting not only the intra- but also the inter-enterprise business processes are necessary in order to enable and ease collaboration between business partners. Therefore, they need to be interoperable. As the basis for building these information systems the concepts of enterprise ontology and business components are very promising. The notion of enterprise ontology, as presented in this paper, is a powerful revelation of the essence of an enterprise or an enterprise network. Reusable and self-contained business components with well-defined interaction points facilitate the accessing and execution of coherent packages of business functionality. The identification of business components, however, is still a crucial factor. The reported

research seeks to improve the identification of business components based on the ontological model of an enterprise, satisfying well-defined quality criteria.

2.5.2 Simon K. Milton, *University of Melbourne*, Understanding the Benefits of Ontology Use for Australian Industry: A Conceptual Study

In IT, rather than philosophy, an ontology makes explicit the meanings of terms used in domains, or concerning a specific reality, so that people and machines can precisely discuss the meaning of data. Specifically, ontology makes data sharing and analysis easier by making the meaning of data, and of the reality to which the database refers, explicit. Ontology has significant uptake in biomedicine but not yet in industry despite much technical development and reporting of specific successes. This research seeks to determine how and why organisations gain benefits from using ontology leading to a rigorously tested model of how business gains benefit from ontology use. This research in progress paper develops a model explaining the benefit of ontology use to firms and outlines our plans to test the model empirically. The outcome is significant for Australian industry because it will guide the efforts of organisations to use ontology effectively.

Chapter there

Programming tool or software development

3.1 backroud

When starting out on an ontology project, the first and reasonable reaction is to find a suitable ontology software editor. Our main concern must include the provided capabilities like ontology versioning (the project development often involves various ontologies – external as well as newly in-house developed), mapping and linking, comparing, merging, reconciling and validating, converting them into other forms (such as XML Schemas, database schemas, and others).

A programming tool or software development tool is a program that software developers use to create, debug, maintain, or otherwise support other programs and applications. The term usually refers to relatively simple programs, that can be combined together to accomplish a task, much as one might use multiple hand tools to fix a physical object. The ability to use a variety of tools productively is one hallmark of a skilled software engineer.

The most basic tools are a source code editor and a compiler or interpreter, which are used ubiquitously and continuously. Other tools are used more or less depending on the language, development methodology, and individual engineer, and are often used for a discrete task, like a debugger or profiler. Tools may be discrete programs, executed separately – often from the command – or may be parts of a single large program, called an integrated development environment (IDE). In many cases, particularly for simpler use, simple ad hoc techniques are used instead of a tool, such as debugging instead of using a debugger, manual timing (of overall program or section of

code) instead of a profiler, or tracking bugs in a text file or spreadsheet instead of a bug tracking system.

3.2 Modeling Tools/Ontology Editors

- Top Braid Composer
- Protege
- Pool Party (vocabulary modeler)
- Neon
- Altova Semantic Works
- Visual Ontology Modeler
- Neologism
- Hozo Ontology Editor
- Text Editor

3.3 Decision criteria for software choice tools/ editors:

Among the most relevant criteria for choosing an ontology editor are the degree to which the editor abstracts from the actual ontology representation language used for persistence and the visual navigation possibilities within the knowledge model. Next come built-in inference engines and information extraction facilities, and the support of meta-ontologies such as OWL-S, Dublin Core, etc. Another important feature is the ability to import & export foreign knowledge representation languages for ontology matching.

Ontologies are developed for a specific purpose and application. Ensure the tool has the right interface for the information science team developing it - which could be highly technical for a help desk/decision tree system and could be a non-technical librarian or business analyst for a knowledge map over an intranet. Also make sure that the end product can be used - i.e. that the tool has an API or appropriate export capability to plug the ontology into the desired application

General description of the tools includes information about developers and availability.

- **Software** architecture and tool evolution includes information about the tool architecture (standalone, client/server, n-tier application). It also explains how the tool can be extended with other functionalities/modules; furthermore, it describes how ontologies are stored (databases, text files, etc.) and if there any backup management system
- . **Interoperability** with other ontology development tools and languages includes information about the interoperability of the tool. Tools interoperability with other ontology tools can be recognized by functionalities like (merging, annotation, storage, inference, etc.), in addition to translations to and from ontology languages.
- **Knowledge representation** is related to presenting of knowledge model of the tool. It also includes the possibility of providing any language for building axioms and whether tool gives support to methodology.
- **Inference services** attached to the tool tells if the tool has a built-in inference engine or it can use other attached inference engine. It also shows if the tool performs constraint/consistency checking. It also provides the possibility of classifying concepts automatically in concept taxonomy and capabilities to manage the exceptions in taxonomies.

- **Usability shows** the existence of the graphical editors for the creation of concept taxonomies and relations, the ability to prune these graphs and the possibility to perform zooms of parts of it. It also says if the tool allows some kind of collaborative working and whether it provides libraries of ontologies.

All these reasons are lat us to use Protégé tool

3.4Protégé:

Protégé is a free, open source ontology editor and a knowledge acquisition system. Protege provides a graphic user interface to define ontologies. It also includes deductive classifiers to validate that models are consistent and to infer new information based on the analysis of ontology. Like Eclipse, Protégé is a framework for which various other projects suggest plugins. This application is written in Java and heavily uses Swing to create the rather complex user interface. Protégé recently has over 200,000 registered users [8].

3.4.1Functionality:

The Protégé platform supports two main ways of modelling ontologies:

- The Protégé-Frames editor enables users to build and populate ontologies that are frame-based, in accordance with the Open Knowledge Base Connectivity protocol (OKBC). In this model, an ontology consists of a set of classes organized in a subsumption hierarchy to represent a domain's salient concepts, a set of slots associated to classes to describe their properties and relationships, and a set of instances of those classes - individual exemplars of the concepts that hold specific values for their properties.

- The Protégé-OWL editor enables users to build ontologies for the Semantic Web, in particular in the W3C's Web Ontology Language (OWL). "An OWL ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms"[9].

3.4.2 Why protégé

Protégé's plug-in architecture can be adapted to build both simple and complex ontology-based applications. Developers can integrate the output of Protégé with rule systems or other problem solvers to construct a wide range of intelligent systems. and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications.

Protege implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protege can be customized to provide domain-friendly support for creating knowledge models and entering data.

- **W3c standards support**

Protege fully supports the latest OWL 2 Web Ontology Language and RDF specifications from the World Wide Web Consortium

- **Extensible open source environment**

Protege is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development

- **Active community**

Protege is actively supported by a strong community of users and developers that field questions, write documentation, and contribute plug-ins. [10].

Screenshot for Protégé

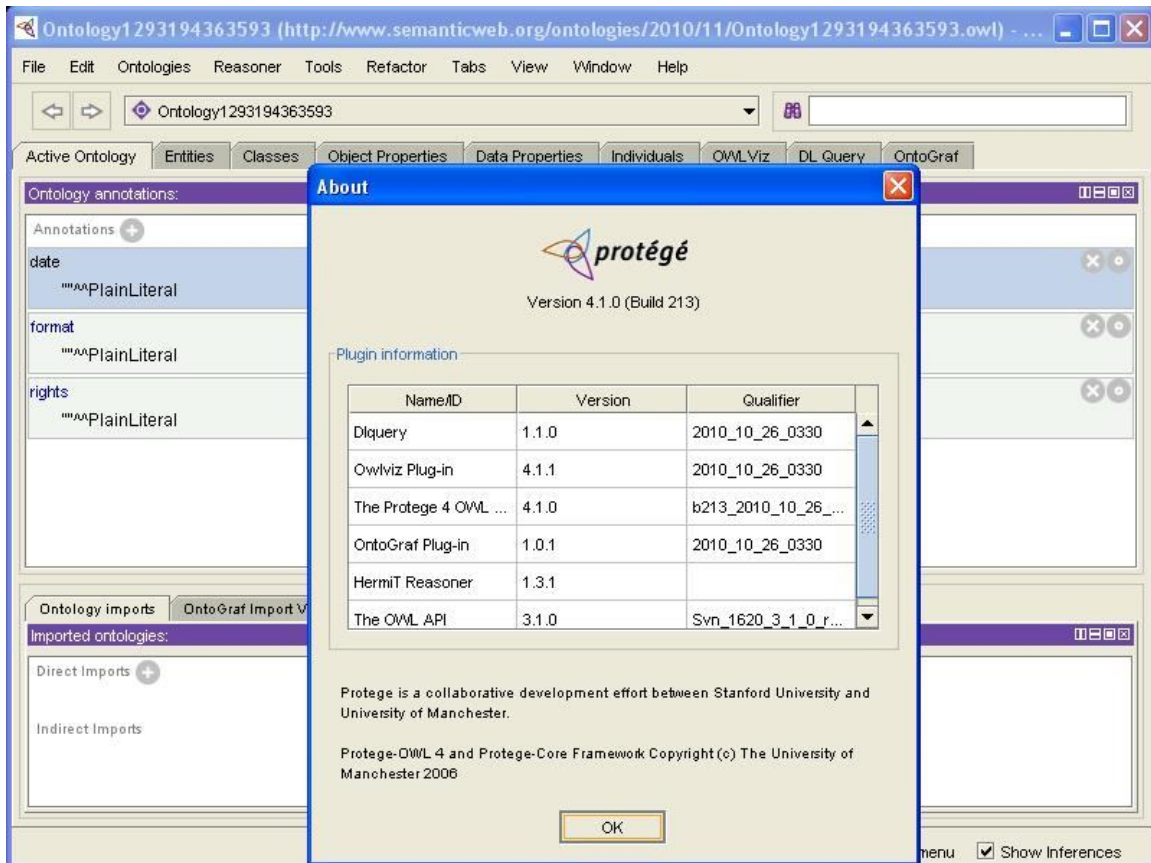


Figure 3.1: Screenshot for Protégé

Chapter Four

Ontology development

4.1 Proposed Ontology

The semantic Web will provide a semantic meaning to the current Web, so it will be easier (for people and machines) to work with this data. There are several ways to improve the Web by providing it with meaning. One is to structure all the information available in some semantic-based form, providing the data along with its meaning. These can be done with some of the current semantic web languages, like XML, OWL, DAML, etc.

The uses of ontology to support the multi-agent system (MAS) tool, development of ontology is critical in the following ways:

Agents use ontology to share common terms and to communicate to other agents (Wooldridge, 2002).

Agent must understand the environment in which they operate. When agents retrieves or store the knowledge, it needs to know how the knowledge is structured. These semantics is the ontology of the knowledge

There are various methods available to design ontology. In this study Protégé OWL methodology shall be used. Protégé is chosen due to the following considerations:

Protégé is a much easier editing tool to learn and more suitable for earlier stage of ontology

The user interface could be optimized to allow non-experts to input and amend ontology in the knowledge base

Supports the OWL/DL format, which is intended to be used in the to-be-developed MAS.

Protégé is free, available for download under the Mozilla open-source license, and has been used for thousands of projects ranging from modeling cancer-protocol guidelines to modeling nuclear power stations. Ontology models are available in the Protégé user forum for others to view [3].

During development and combination of the (S-NPF) self management system ontology, we discovered that certain steps are better performed in iterations, as follows:

1. Define the scope and purpose of ontology
2. Consider reusing existing ontology
3. Enumerate important terms in the ontology
4. Define the individual examples
5. Define the properties of classes—slots
6. Define the classes and the class hierarchy for individual groups
7. Define the facets of the slots
8. Create instances

4.1.1 Determine the domain and scope of the ontology

To determine domain and scope of ontology we can begin with answer several questions and sketch a list of questions that knowledge base based on the ontology should be answer (say it competency questions) [11].

The questions are?

- What is the domain that the ontology will cover in (S-NPF)?
- What are types of departments in (S-NPF)?
- What are types of branches in (S-NPF)?
- What is the service provided by departments?

- Who is the responsible director of administration?

Judging from this list of questions, the ontology will include the information on the domain and scope of the ontology for (S-NPF)

4.1.2 Consider reusing existing ontologies

Technologies for the efficient and effective reuse of ontological knowledge are one of the key success factors for the Semantic Web. The re-use of ontologies and of knowledge collected in the context of ontology creation

In this research, and Reusing existing ontologies may be a requirement if your system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies the ontology will be built from scratch.

4.1.3 Enumerate important terms in the ontology

It is useful to write down a list of all terms we would like either to make statements about or to explain to a user. What are the terms we would like to talk about? What properties do those terms have? What would we like to say about those terms? For example:

- Branches
- departments
- Services
- responsible director
- pensioner

4.1.4 Define the Classes and the Class Hierarchy

There are several possible approaches in developing a class hierarchy:

- A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts.
- A bottom-up development process starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts.
- A combination development process is a combination of the top-down and bottom-up approaches: We define the more salient concepts first and then generalize and specialize them appropriately [12].

- A top-down development process:

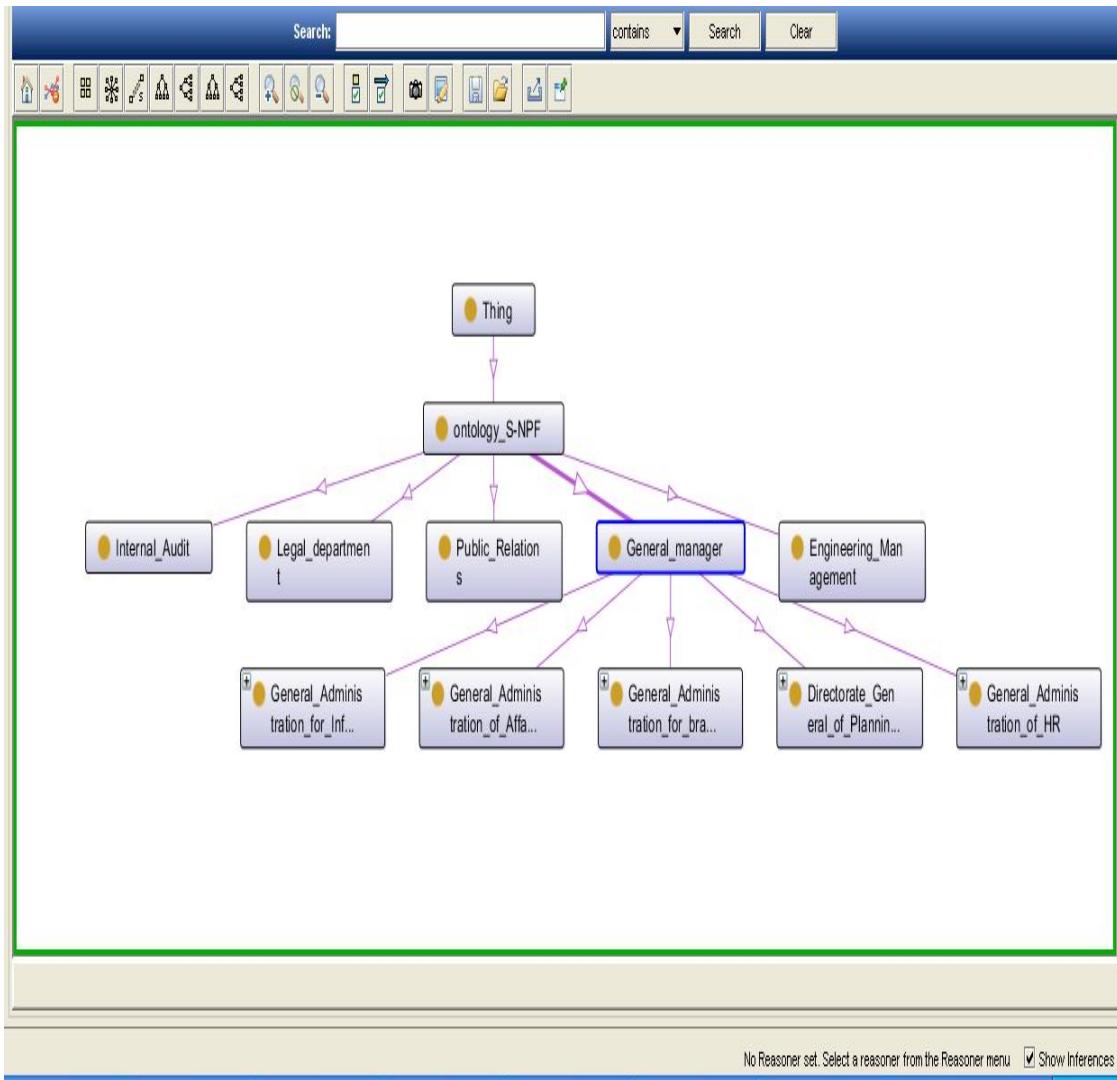


Figure 4.1: top-down development process Protégé Class browser

- A bottom-up development process:

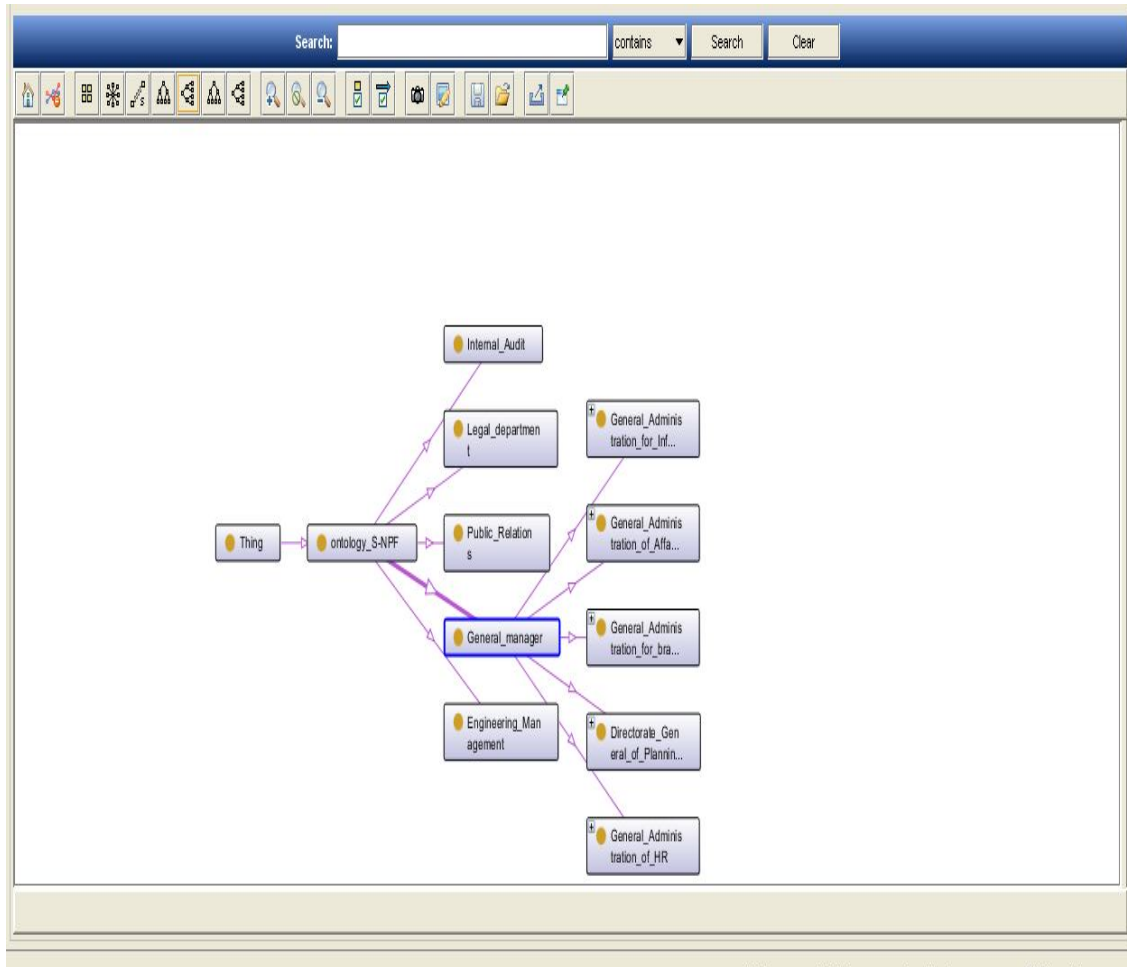


Figure 4.2: A bottom-up development process Protégé Class browser

- **A combination development process:**

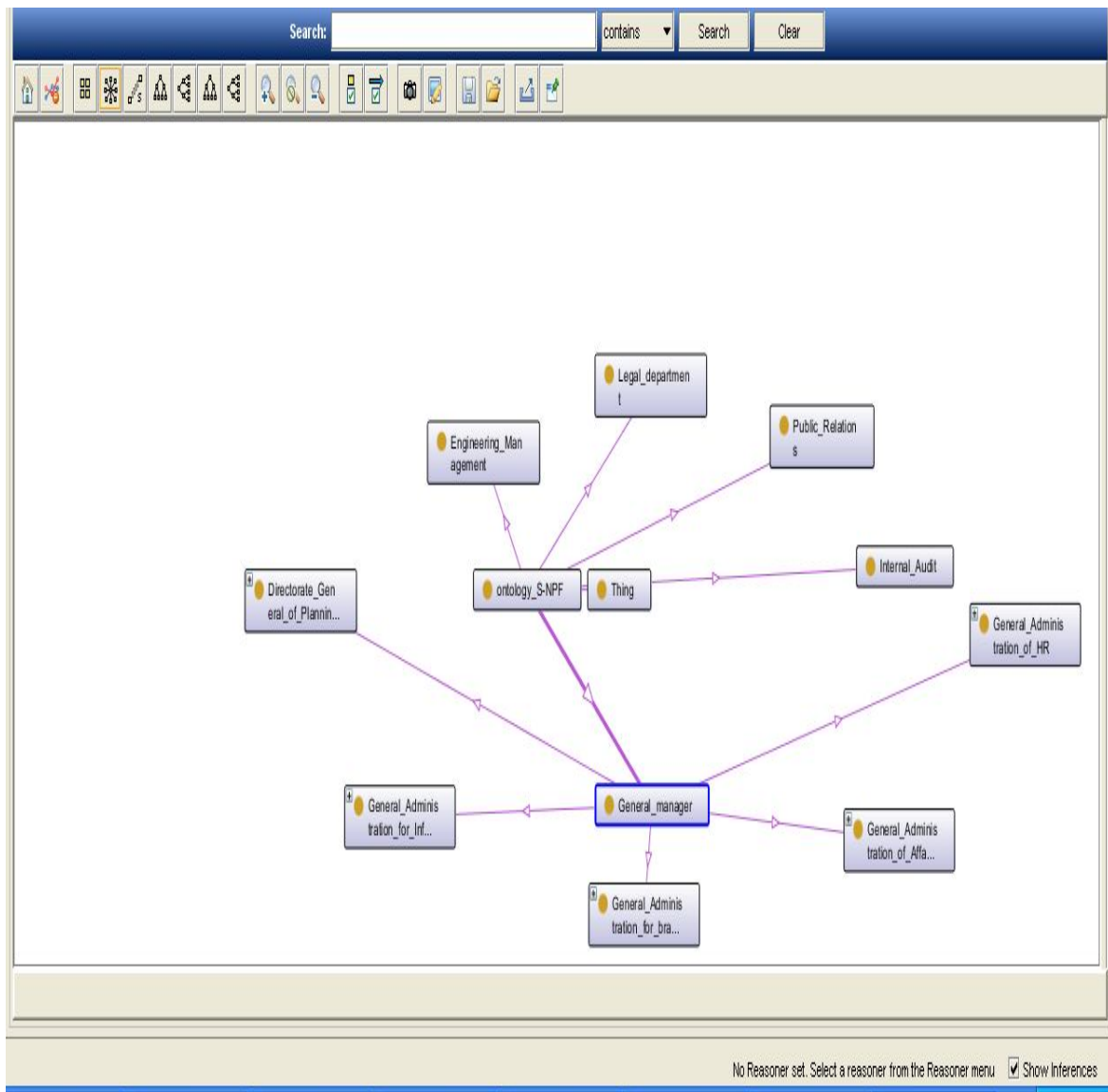


Figure 4.3: A combination development process Protégé Class browser

4.1.5 Define the properties of classes

Once we have defined some of the classes, we must describe internal structures of concepts. After selecting the classes from the list created by step-3 most of the remaining terms are likely to be properties of these classes. For each property in the list we must determine which class it describes. These properties become slots attached to classes. In general, there are several type of object properties that can become slots in an ontology. In ontology properties of classes are:

- *has Branches*
- *has departments*
- *is part of*
- *has Services*
- *has responsible manger*
- *has pensioner*

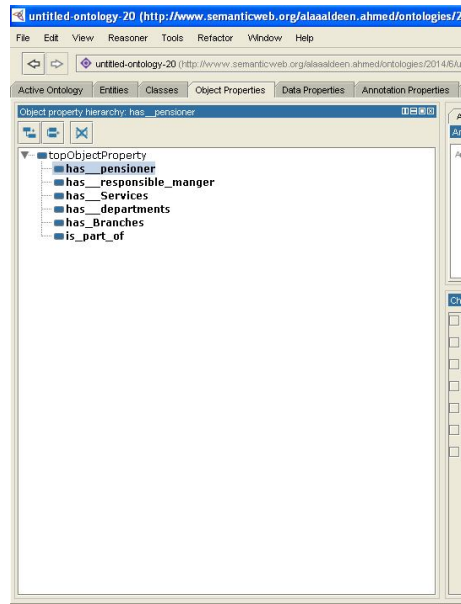


Figure 4.4: Define the properties of classes

4.1.6 Define properties of properties

This step to describing single individuals, their class memberships, and how classes can relate to each other based on their instances. But more often than not, ontology is also meant to specify how the individuals relate to other individuals. These relationships are central when describing (S-NPF) system.

4.1.7 Create instances

Individuals (instances) are the basic, "ground level" components of an ontology. The individuals in an ontology may include concrete objects such as people, animals, tables, automobiles, molecules, and planets, as well as abstract individuals such as numbers and words (although there are differences of opinion as to whether numbers and words are classes or individuals). Strictly speaking, an ontology need not include any individuals, but one of the general purposes of an ontology is to provide a means of classifying individuals, even if those individuals are not explicitly part of the ontology

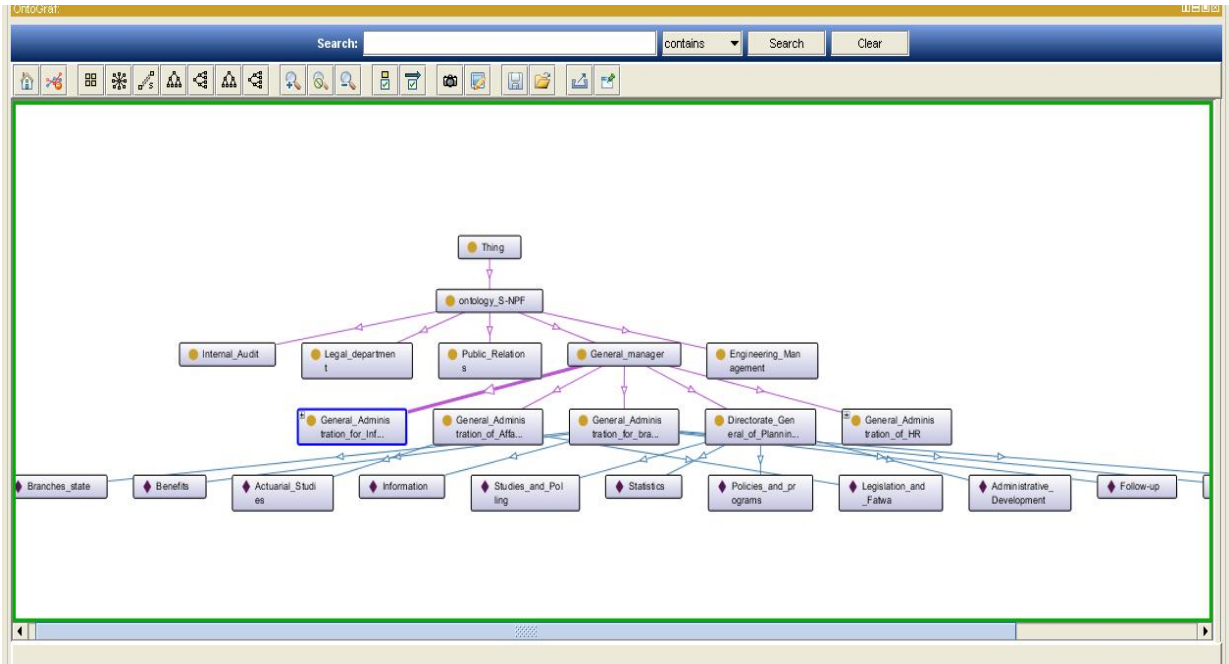


Figure 4.5: Create instances of ontology 1

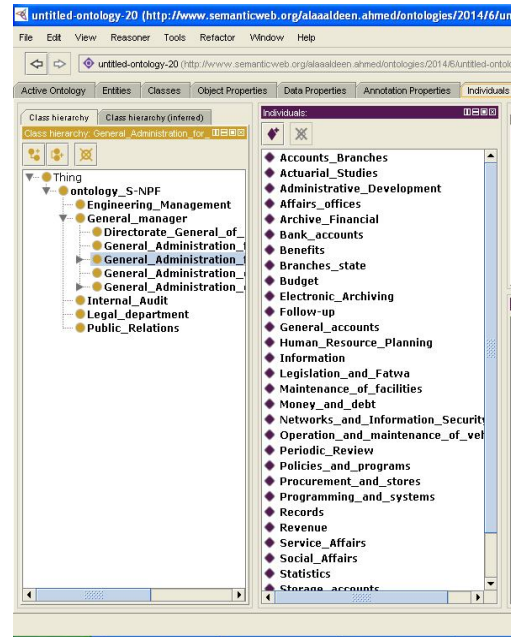


Figure 4.5: Create instances of ontology 2

4.2 Invoke the Reasoner

A reasoner checks for hierarchies (subsumption), domains, ranges, conflicting disjoint assertions, and the like. In addition to these checks, the reasoner also calculates the resulting inferred hierarchy and other properties. you must turn it on via the Reasoner main menu option in Protégé. Choosing this menu option presents a listing of all reasoners.

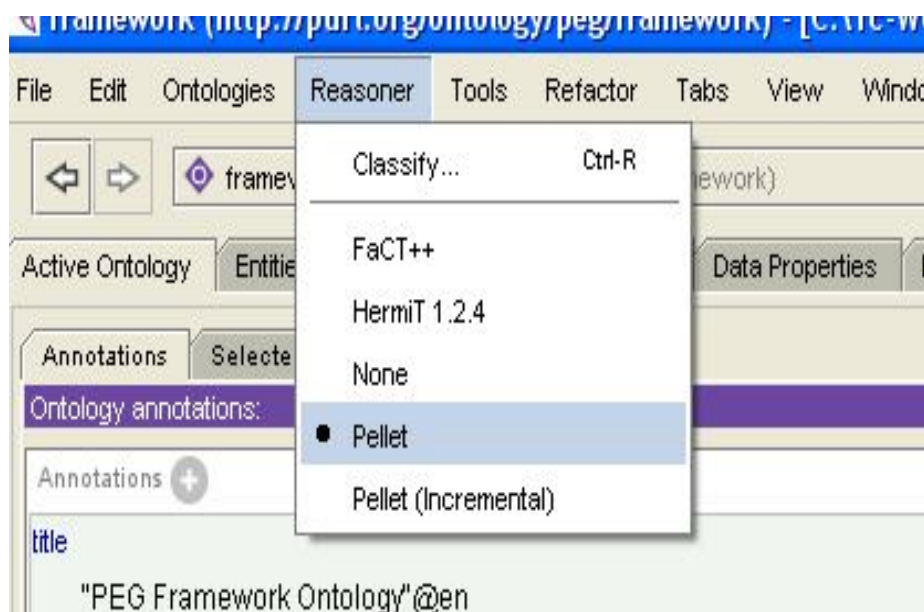


Figure 4.6: Invoke the Reasoner by Protégé

Chapter five

Results & Recommendations

5.1 Results

The ontology development has been done by using semantic web for (S-NPF) ,by providing a conceptualization of a domain into machine readable format formal explicit description of a domain consisting of classes, which are the concepts found in the domain , by using . Protégé tool -OWL editor enables users to build ontologies for the Semantic Web, in particular in the W3C's Web Ontology Language (OWL). "An OWL ontology may include descriptions of classes, properties and their instances . and checking an ontology by Invoke the Reasoner option

5.2 Recommendations

- The (S-NPF) must use the ontology to share domain information define the concepts and relationships.
- The ontology must be updated by providing and define new terms.

References

1. Eltayb kononoa, (S-NPF),(2013),hasad journal.
2. Muhammad Shahzad,(2013), A semantic web approach for dealing with university courses
3. David Vallet, (2011), An Ontology-Based Information Retrieval Model
4. Larry Reeve, (Spring 2004), Information Retrieval on the Semantic Web Using Ontology-based Visualization INFO780.
5. <http://kogs-www.informatik.uni-hamburg.de/~neumann/WMA-WS-2007/WMA-12.pdf>, 18/6/2014,time 12:30
6. https://markwatson.com/opencontent_data/book_lisp.pdf, 18/6/2014,time 12:30
7. <http://www.obitko.com/tutorials/ontologies-semantic-web/web-ontology-language-owl.html2/72014>, time 20:00
8. Emhimed Salem Alatrish, Faculty of Mathematics, University of Belgrade, Serbia (phone: 381-616169812, email: emhimed_alatrish@yahoo.com).
9. <http://www.dcc.ac.uk/resources/external/protege-ontology-editor> ,10/8/2014,time 9:40
- 10.<http://protege.stanford.edu/> 10/8/2014,time , 9:40
- 11.http://en.wikipedia.org/wiki/Ontology_editor 10/8/2014,time , 9:40