An Ontology Driven Knowledge Discovery Framework for Dynamic Domains: Methodology, Tools and a Biomedical Case.

Paulo Gottgtroy

A thesis submitted to

Auckland University of Technology

in fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)

2010

School of Computing and Mathematical Sciences

Primary Supervisor: Prof. Nikola Kasabov

Co-supervisor: Prof. Stephen MacDonell

Table of Contents

List of Figures

List of Tables

"I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning."

Paulo Cesar Moreira Gottgtroy

ACKNOWLEDGEMENTS

This thesis would not be possible without the contribution and assistance of a number of individuals. I would like to acknowledge their support.

A special thanks to my thesis supervisors, Prof. Nikola Kasabov and Prof. Stephen MacDonell, for their patience and guidance in bring me to the end of this work.

To Joyce d'Mello who has not only taken care of our Institute, doctoral procedures, etc but has also been a friend who support me and my family in this long journey.

To my colleagues, Simei, Wagner, Richard, Dougal, Vishal, and all Kedri members for their constant support and incentive.

I gratefully acknowledge the financial assistance provided the Knowledge Engineering and Discovery Research Institute, the Auckland University of Technology and specially the Tertiary Education Commission for the Top Achiever Doctoral Scholarship.

I dedicate this thesis to my family, Márcia, Renata and André for their tolerance, understanding and unlimited support. I could not forget my parents and in-laws who have been source of love, persistence and dedication throughout my life.

ABSTRACT

The explosive growth in the volume of data and the growing number of disparate data sources is bringing enormous opportunities and challenges to many research communities. In the biomedical domain, the challenge of knowledge discovery from diverse and heterogeneous biomedical data sources in order to make knowledge and concepts sharable over applications/experiments and reusable for several purposes is both complex and crucial.

Opportunities arise by the simple act of connecting different facts and points of view that have been created for one purpose, but that in light of subsequent information can be reused in a quite different context, to form new concepts or hypotheses. However such interactions cannot be determined in advance - for one thing, there may be more or fewer problem dimensions involved in a process than were known when the process initially started. Modelling of such processes is a challenging task but is one with practical applications in many disciplines. Identifying these data interactions, learning about them, extracting knowledge, and building a reusable knowledge base that applies leading artificial intelligence and soft-computing methods will guide future research and practice and is at the core of this research.

This thesis bridges two fields, ontology engineering and knowledge discovery in databases (KDD) for successful data mining in dynamic domains. The novel Ontology Driven Knowledge Discovery framework (ODKD) developed here provides a means of describing and representing evolving knowledge, managing shared knowledge, integrating data mining tools and algorithms, and enabling semantically rich knowledge discovery. The ODKD framework encompasses a meta-knowledge model, a methodology and software engineering tools to semantically support knowledge discovery in databases.

As a research endeavour that crosses the boundaries of information systems, software engineering, computer science and knowledge engineering, this thesis integrates in its research design a multi-methodological constructive approach widely used in the information systems discipline with the Design Science research methodology used in the engineering disciplines.

The ODKD suite of tools implements a framework able to integrate the evolving ontology meta-knowledge model and methodology to provide a more holistic view of the KDD process than previously possible. The extensible system architecture enables the adoption of new knowledge, industry standards and tools and provides a solution architecture framework to deploy the ontology driven knowledge discovery process in different domains and software platforms. In this thesis the functional capabilities of the tools and the appropriateness of the conceptual structures are demonstrated and evaluated in the context of a biomedical application case study.

The ontology driven knowledge discovery proposal is novel in that it integrates both ontology engineering and KDD processes into one framework and a supporting methodology. It creates a new semantic structure, channel and process able to combine several sources of information and data mining tools within a shared knowledge repository. As such, it addresses the challenge of using computer models in concert with human knowledge to test hypotheses and to validate and integrate knowledge that may be created by different sources with diverse intentions but that when linked can promote the discovery of new knowledge.

Introduction

1.1. Introduction

The explosive growth in the volume of data and the growing number of disparate data sources is bringing enormous opportunities and challenges to the research community (Blagosklonny & Pardee, 2002; Foster, 2006). Terabyte and petabyte databases, once unthinkable, are now a reality in a variety of domains, including marketing, sales, finance, healthcare, molecular biology, and various government sectors.

In the biomedical domain, for example, the challenge of knowledge discovery from diverse and heterogeneous biomedical data in order to make knowledge and concepts sharable over applications/experiments and reusable for several purposes is both complex and crucial. It is central in the support of decision making in medical practice as well as enabling comprehensive knowledge acquisition by medical research communities and molecular biologists involved in advancing biomedical understanding.

Biomedical data can consist of information stored in the genetic code, identified in genomics and proteomics research by discovering sequencing patterns, gene functions, and protein-to-protein interactions. It can also consist of experimental results from several sources such as therapeutic data, biomedical literature, and clinical data, for example, information collected from clinical patient data for clinical trial design (Burgun, Botti, Fieschi, & Le Beux, 1999; D'Hollosy et al., 1996; Ramoni, Stefanelli, Magnani, & Barosi, 1992). Large amounts of information and knowledge are produced at a rapid rate in the practice of medicine (Bodenreider, Mitchell, & McCray, 2002). Making medical knowledge sharable over applications is important as it enables the efficient management of resources and government investment in the health sector, as well as supporting physicians involved in health care.

Similarly, we are faced with many complex and dynamic problems in the business world, particularly those related to decision making processes. The amounts of data produced daily from deals and transactions, including internal and external processes are growing at an exponential rate. Moreover, decision makers need to deal with information from different sources. In some business domains, such as agribusiness, they need to analyze climate, marketing, financial, and biological data among others to gain competitive advantage and improve the quality of their products.

This huge amount of data, and its diversity, has been the main driving force for the development of theories, methodologies and technologies that aim to transform these data into real assets in both the industry and research communities. On-line analysis, data warehousing, and visualization techniques, for example, when combined with simple tabular data, empower users, mainly decision makers, with their familiar world of spreadsheets and graphs resulting in the billion-dollar industry called business intelligence.

There are several aspects and levels of abstraction that must be considered when working with data. The database industry is involved, among other things, with the management, storage, and security of data. Systems theorists tend to investigate the power of data transformation into information, knowledge and wisdom as an organization-changing factor. Ontologists look at the challenges associated with knowledge integration and representation. Even among more focused areas of activity, such as data mining and knowledge engineering, we can find different approaches working with data and extracting useful and non-trivial knowledge from previous facts.

In spite of the different approaches to the transformation of data into real viable assets, all of these research areas are presented with a common challenge that is central to this thesis - how to acquire, share and maintain knowledge from large and distributed databases in the context of domains/problems in which processes are changing over time.

Biomedical discovery, for example, is an intrinsically complex and risky process. One of the aspects of the biomedical discovery process is its iterative nature in terms of analyzing existing facts or data, to validate current hypotheses or to generate new ones linked into testable chains and networks. Opportunities arise by the simple act of connecting different facts and points of view that have been created for one purpose, in light of subsequent information, which can be reused in a quite different context, to form new concepts or hypotheses. However this interaction cannot be determined in advance, for example there may be more, or fewer problem dimensions involved in the process than were known when the process initially started.

These evolving characteristics are difficult to model, because some of their dimensions may not be known *a priori*. In addition unexpected discoveries or changes may happen at certain times during development, so they are not predictable in the longer term. Thus, modelling of such processes is a challenging task but with many practical applications in life sciences and business.

Modelling data interactions, learning about them, extracting knowledge, and building a reusable knowledge base which combines leading artificial intelligence and soft-computing methods will guide future research and practice and is at the core of this research.

This thesis bridges two fields, ontology engineering and knowledge discovery in databases (KDD) for successful data mining in large databases. In particular, it analyzes how data mining can assist in efficient and effective large-volume data analysis in order to build a sharable and evolving knowledge repository, and how ontologies can leverage this repository's semantic power to improve the results of data mining tasks and the analysis of knowledge discovered.

The thesis addresses three main tasks: knowledge acquisition, knowledge maintenance and knowledge sharing, in the context of an ontology-driven knowledge discovery process. The primary outcome is a process, which integrates the activities underlying ontology engineering and knowledge discovery in databases and that facilitates the creation of ontologies able to represent the dynamic and uncertain nature of domains in order to provide ongoing support for the decision making process over time.

To that end, a novel ontology engineering framework has been developed which encompasses a methodology, a meta-knowledge model and software engineering tools to semantically support knowledge discovery in databases.

1.2. Research questions

'Knowledge Discovery in Databases' focuses on the overall process of knowledge discovery from data, including how the data are stored and accessed, how features can best be selected and identified for data mining tasks, how results can be interpreted and visualized, and how the overall human-machine interaction can usefully be modelled and supported (Mihael, 2002). In a nutshell, it is the process of transforming data into non-trivial knowledge for decision making support.

Transforming data into true assets is a challenging task which involves the discovery and sharing of knowledge from a potentially huge and diverse amount of facts/data and sources. This research addresses this challenge by investigating the broad question of how to acquire, share and maintain knowledge from large and distributed databases in the context of domains/problems in which processes are changing over time.

In order to answer this broad question, the thesis explores three main research topics: ontology engineering, knowledge discovery in databases, and the integration of the two. It addresses these topics in terms of four specific questions:

- Can ontologies ‘evolve’?
- Can we integrate ontology engineering and data mining processes?
- Can ontology support data mining tasks?
- Can data mining tasks enhance ontology knowledge?

Of course, each of these questions can be answered as a simple ‘Yes’ or ‘No’. However, it is the substance underlying each answer that forms the content and contributions of this thesis. Thus the questions above reflect the design science challenges of the research – each with specific requirements and success criteria that are addressed in turn.

Ontology evolution is addressed by the creation of a meta-knowledge able to capture information which supports ontology change, taking into consideration several inputs such as those informed by experts (expressed in rules or annotations) and those acquired from artificial models (data mining models and simulations).

Ontology engineering and data mining process integration is addressed by a novel methodology and process which takes into consideration both processes requirements and suggests a new ontology driven methodology and framework.

Ontology support for data mining is addressed by a two-fold approach: by the selection of data samples using a multidimensional and semantically rich ontology model at the initial phase of a knowledge discovery process; as well as at the end of the process when enhancing the results acquired from a data mining model by linking the results into the generic ontology model. This approach then closes the loop of an ontology driven knowledge discovery process.

Ontology enhancement from data mining is addressed as a mechanism which acquires knowledge from data mining and integrates it into the ontology. It allows sharing, annotation, enhancement, and evaluation of the knowledge by human experts who can then validate hypotheses or create new ones in the light of the newly acquired knowledge.

The rest of this chapter describes the research methodology adopted in this thesis. It covers a multi-methodological approach which has as a core component the development of tools that support the framework proposed. The research design defines the research process, conceptual framework, system architecture; its requirements and assumptions, delimitation of the study and its limitations.

1.3. Research Methodology

As a multi-disciplinary topic involving engineering processes and software development, this research falls more readily into the category of applied research than that of basic research. Applied research has two main goals, one theoretical and one practical (Adams & Courtney, 2004):

- “To increase (theoretical) knowledge: to understand why things happen in a particular context, and
- “To improve practice by conducting research that will ultimately yield usefulness and improve processes in a particular field.”

There is substantial support in the literature for the use of systems development and prototyping as a valid and valuable computer and information sciences research method (Hartmanis, 1994; Newell & Simon, 1976; Nunamaker, Chen, & Titus, 1990). Newell and Simon (1976) argue that computer programs are not black boxes. *“We can open them up and look inside. We can relate their structure to their behaviour and draw many lessons from a single experiment”*. Hartmanis (1995) argues that the changes in research paradigms in computer science are driven largely by technology and therefore demonstrations via the likes of simulation and functional prototyping can play the role of experiments in this domain. Empirical study of the development process and the use of prototype systems provide valuable feedback for more effectively designing a system and consolidating an iterative research process.

This research applies an adapted version of Nunamaker's methodology (Nunamaker, Chen, & Titus, 1991) that considers Design Science (Hevner, March, Park, & Ram, 2004) as a theory-building methodology. It is an iterative research process (Figure 0-1), integrating ontology engineering and knowledge discovery in databases.

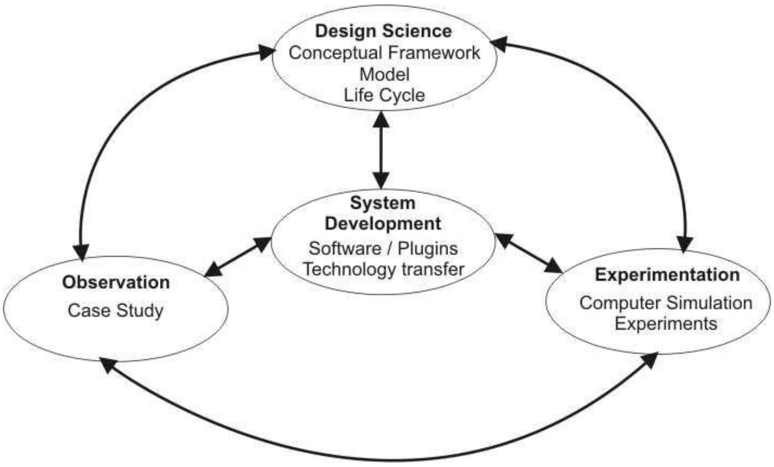


Figure 0-1– Multi-methodological approach applied in this research (adapted from Nunamaker et al. (1991) and Adams et al (2004))

The adapted multi-methodological approach consists of four major phases: *design science*, *system development*, *experimentation* and *observation.* This research methodology then forms the base for the evaluation of the thesis as described below.

"*When the proposed solution of the research problem cannot be proven mathematically, or if it proposes a new way of doing things, researchers have to develop a system to demonstrate the validity of the solution, based on the suggested new methods, techniques, or design*" (Nunamaker, Chen, & Titus, 1991).

Design Science

As defined by the Buckminster Fuller Institute (2006), design science's function *"is to solve problems by introducing into the environment new artefacts"*. Design science as used in this work is directed at the building and evaluation of IT artefacts. It is especially valuable and appropriate in this work as it facilitates the iterative process of developing tools and evaluating them in order to validate the conceptual framework.

Design science involves the development of constructs, models, methods and implementations which can be compared with the theory-building notion of concepts, construction of conceptual frameworks, new methods or models (Nunamaker, Chen, & Titus, 1990). These design science outcomes are manifested in the model, conceptual framework and life cycle developed in this research.

Constructs as **conceptualizations** (conceptual frameworks) can be used to describe problems within a domain, define methods and processes and specify their solutions. In the context of this research, a conceptualization is embedded in the process and methodology.

The third outcome is the hybrid **life cycle** integrating ontology engineering and an industry standard process for data mining delineated by the Ontology Driven Knowledge Discovery Process.

Conceptualization or the establishment of the theoretical grounding of the system requirements is posited in Adams (2004) as "*the focal point of the research* *effort”*. The conceptual basis is followed by the development of the system that acts as proof of the concept of the proposed life cycle and conceptual framework.

System Development

The **System development** phase of this research consists of five stages that result in the design of the system, system architecture, prototype, product development, and technology transfer respectively. This phase is concerned with theory testing and allows for a realistic technological evaluation of the product developed and its potential for acceptance. This phase also guides the design of associated experiments and forms the basis for conducting systematic observation and the development of the case studies.

Experimentation

The **Experimentation** phase concentrates on the validation of the underlying theories, systems, and technology transfer. It provides the main impetus for the refinement of the conceptual framework, models and tools. The results produced enable the refinement of theories and a comparison against the research requirements defined in the earlier stages of the work.

Observation

Once the prototype is built, the **observation** phase is enacted to test the prototype and validate it against the specified requirements as well as to consider its impact on the research problem. This contributes to the provision of holistic insights into the domain.

The results produced through the use of the prototype are interpreted and evaluated based on the conceptual framework requirements and the requirements of the system. The results guide the improvement of the tools to better support the conceptual framework. Further observations are then made in the context of the main biomedical case study and these observations form the basis for the contribution, conclusions and future research sections of this thesis.

The multi-methodological approach is an evolutionary process. In every phase of the research process, experiences accumulated guide improvements in design decisions made in previous phases and/or leverage new discoveries which improve the conceptual contribution of this research.

The next section describes the research process in detail, linking the methodology with the specific outcomes of each chapter.

1.3.1. Detailed Research Process

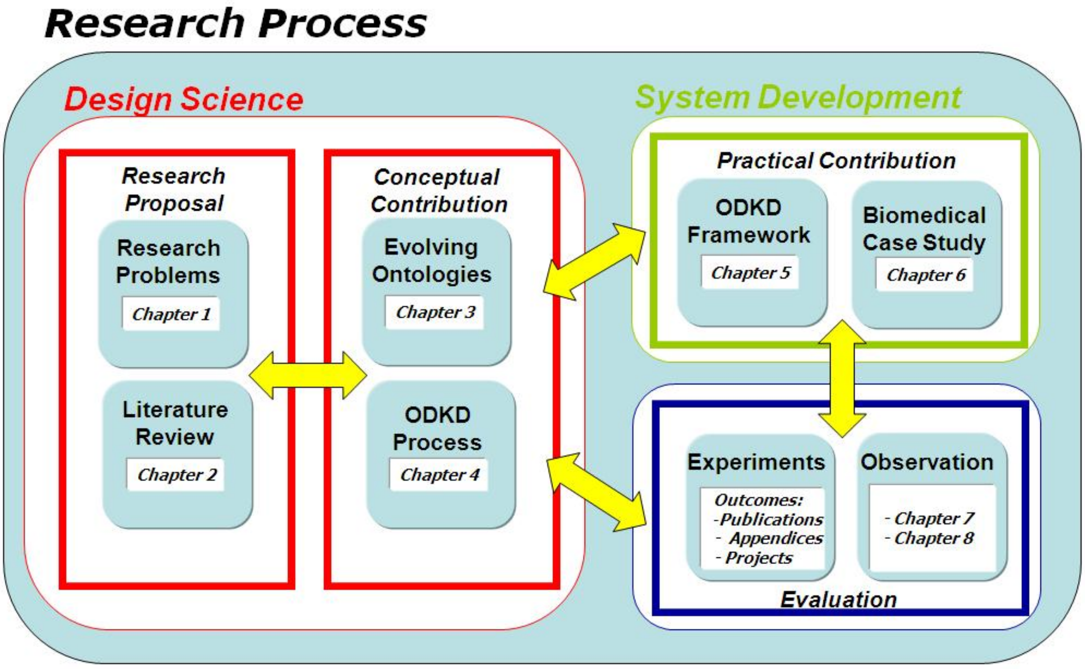


Figure 0-2– A Research Process diagram integrating the methodology and the thesis' structure.

The research process is divided into four main blocks: Research Proposal, Conceptual Contribution, Practical Contribution and Evaluation. The first two blocks are covered by a Design Science research methodology and are responsible for the theoretical contribution of this research.

The Practical Contribution encompasses the set of ontology engineering tools developed to support the conceptual framework and its application in a biomedical knowledge discovery tool. Nunamaker's systems development research methodology, described briefly in the next section, is followed in this Practical Contribution stage.

The Evaluation stage is concerned with the validation of the framework and associated tools. This phase covers the iterative process of design, implementation, validation and refinement. In each phase of the process, the outcomes produced are compared with the requirements and then both the conceptual framework and tools are adjusted when and where necessary.

Although the chapters represent the final results of this iterative process, each iterative step can be followed through the reporting of the experiments that are explicitly described as sections within the chapters or through the list of published outcomes which shows the evolutionary design process of both the conceptual and practical contributions of this research.

System Development Research Process

Software Systems Development can be considered a research domain and a research methodology. As a methodology, systems development is not new and has been widely used to study nature and to create new things (Ives, Hamilton, & Davis, 1980; Joline & Joey, 1995; Pechenizkiy, Puuronen, & Tsymbal, 2006). The use of engineering tools such as integrated development environments (IDEs) and computer-aided design and computer-aided manufacturing (CAD/CAM) to support systems development has also been found very useful in amplifying human intelligence and in transferring knowledge for wider use.

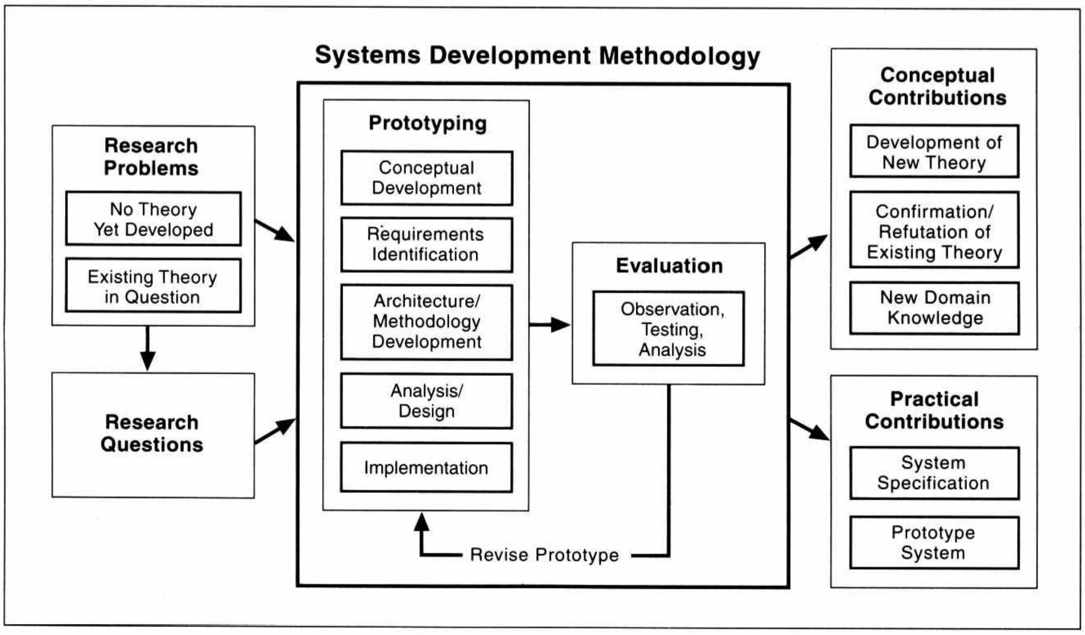


Figure 0-3 – A Research Process of Systems Development Research Methodology (Source: Nunamaker 1996).

This research employs Nunamaker's systems development process (Figure 0-3), taking into consideration the requirements posited by Pechenizkiy (2006) and Dhar's Intelligence Density measurement (1997) (section 2.2.5) as criteria for the evaluation of the usefulness and 'interestingness' of the ontology driven knowledge discovery process.

Nunamaker et al. (1991) describe systems development as belonging to engineering, developmental and formative types of research. Its basic principles are summarized as follows:

- Design is the most important part of a system development process. Design involves the understanding of the studied domain, the application of relevant scientific and technical knowledge, the creation of various alternatives, and the synthesis and evaluation of proposed alternative solutions.

- Building a prototype system always helps to study and understand a research domain.

- A good system architecture provides a road map for the system building process. It puts the system components into the correct perspective, specifies the system functionalities, and defines the structural relationships and dynamic interactions among system components.

- Researchers must identify the constraints imposed by the environment, state the objectives of the development efforts (i.e., the focus of the research), and define the functionalities of the resulting system to achieve the stated objectives.

- The process of implementing a working system can provide researchers with insights into the advantages and disadvantages of the concepts, the frameworks, and the chosen design alternatives.

- Depending on the focus of the research, one might emphasize the new functionalities or innovative user interface features of the proposed new system rather than the throughput or the response time of the system.

- Implementation is used to demonstrate the feasibility of the design and the usability of the functionalities of a system development research project.

- Once the system is built, researchers can test its performance and usability as stated in the requirement definition phase, as well as observe its impact on individuals, groups, or organizations.

- The test results should be interpreted and evaluated based on the conceptual framework and the requirements of the system defined during the earlier stages.

The software engineering nature of this research determined the adoption of a systems development research methodology as its core component. However the integration of two ‘evolving’ fields, namely ontology engineering and KDD, and the proposal of a novel conceptual framework brought further requirements to the development of the conceptual framework and its tools. In the next section we present the scope, design framework and requirements of this research.

1.4. Scope of the research

Although ontology building has been gaining increasing attention since 1990, the mid-90s can be considered the beginning point of the ontology engineering field.

The first workshop on ontology engineering, for example, was held in conjunction with the 12th European Conference on Artificial Intelligence in 1996, where defined steps aiming to reduce the effort to build ontologies were identified (Gómez-Pérez, Fernández-López, & Corcho, 2004). The mid-90s is also considered a Knowledge Discovery in Databases milestone. In 1996, Fayyad et al proposed the idea of the second generation of KDD tools where the process characteristics were emphasized rather than the development of data mining (modelling) techniques. The idea for a continuum process for making sense of data was established.

Since the mid-90s significant research effort has been expended in both ontology development and KDD processes. However, somewhat less common is the investigation of the role of ontologies in incremental and cyclic approaches to knowledge discovery – the focus of this work.

There exist examples of the application of machine learning techniques for building ontologies. For instance, ontology learning from text in the semantic web is already being recognized as a worthwhile technique to reduce the work of building hierarchical ontologies from scratch or refining existing ones (Honavar et al., 2001; Reinberger, Spyns, Pretorius, & Daelemans, 2004). The role of ontologies in knowledge discovery, however, has received less attention (KDO-2004, 2004).

The integration of ontologies and knowledge discovery is a novel and challenging field of research. *"The use of prior knowledge may significantly enhance knowledge discovery from large datasets or text collections. Currently, in most KDD projects, prior knowledge is only present implicitly (in the head of the human analyst)* *or in the form of textual documentation. Even in knowledge-intensive approaches, the background knowledge is often not organized around a well-formed conceptual model. This practice seems to ignore latest developments in knowledge engineering, where domain knowledge is typically defined by formal ontologies."*(Text extracted from the motivations of the first Workshop in Knowledge Discovery and Ontologies (KDO-2004))

This thesis is concerned with knowledge, represented as ontologies, and its integration in a knowledge discovery process. The use of prior knowledge may help (the user or a system) in selecting suitable data for a data mining task, to prune the space of hypotheses and to represent the output in the most comprehensible way in a knowledge discovery process. Instead of looking specifically at the application of ontology in KDD, such as the use of prior knowledge to improve the effectiveness of clustering algorithms, or the application of data mining techniques in ontologies, such as ontology learning from text, the focus of this research is on the closed-loop integration of ontologies in knowledge discovery in databases (Figure 0-4).

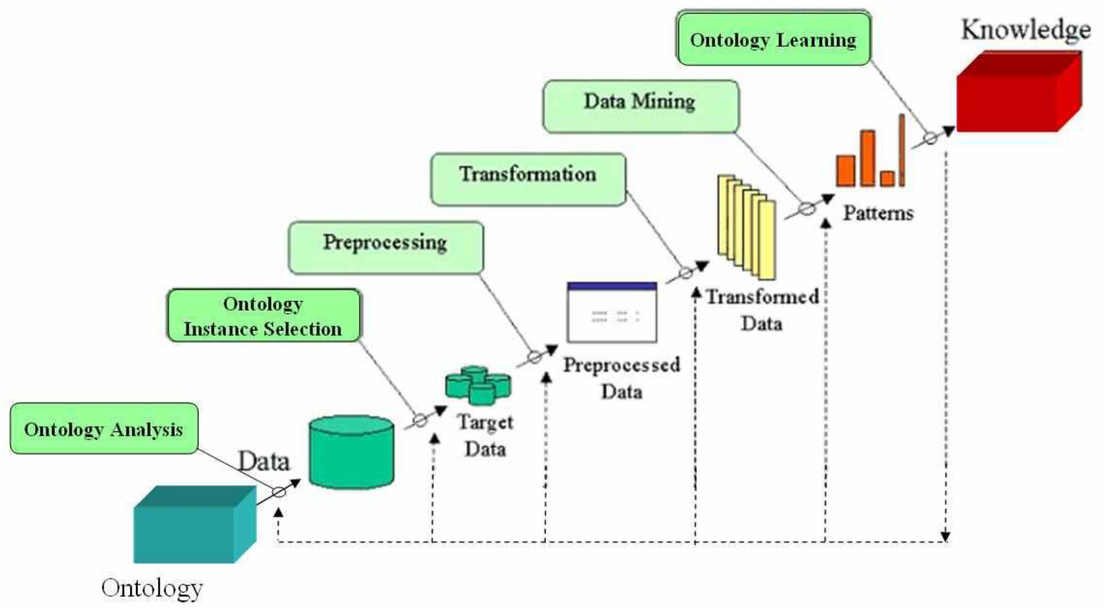


Figure 0-4 - Ontology and KDD integration (adapted from (Fayyad, 1996)).

1.4.1. Implementation Environment

There are three main aspects of the implementation environment to be discussed: the knowledge discovery process being enacted, the domain in which the implementation is centred, and the ontology engineering tool extended in this thesis.

Knowledge discovery tasks

There are various knowledge discovery processes in the literature (Fayyad, Piatetsky-Shapiro, & Uthurusamy, 2003; Gregory, 2000; Shearer, 2000). There are also various methodologies and methods for building an ontology (Gandon, 2006; Gómez-Perez, 1999; A. Gómez-Pérez, Fernández-López, & Corcho, 2004; Mizoguchi & Ikeda, 2006). Most of these ontology building methods are influenced by the problem domain on which they are based, such as TOVE (Toronto Virtual Enterprise) in the domain of enterprise modelling. Others are influenced by reference disciplines. The METHONTOLOGY methodology (Fernandez, Gomez-Perez, &

Juristo, 1997), for example, is based on the adoption of the IEEE standard for software development - 1996.

Considering the positioning of this thesis in KDD, two of the most widely used KDD processes in both academia and industry have been selected: Fayyad (1996) and CRISP- DM (Shearer, 2000). Although the research is also influenced by a reference discipline (KDD), the implementation phase considers proposed ontology engineering activities such as ontology integration, merging and learning in the context of the ontology driven knowledge discovery process.

It is important to mention that, although the tools developed in this research are general, that is, they can be used in different knowledge discovery tasks and to support different techniques such as text mining, the main target of this thesis is mining knowledge from data and enhancing ontologies from instances within the context of biomedical knowledge discovery.

The techniques and life cycle adopted in this research are discussed in detail in Chapters 4 and 5.

Implementation Domain

The biomedical domain has been selected as a representative application domain for this thesis for the following reasons:

– **Biological knowledge discovery is a challenging task**. The tremendous amount of DNA sequence information that is now available provides the foundation for the study of how the genome of an organism functions (Bodenreider, Mitchell, & McCray, 2003). At the same time, millions of easily retrievable facts on biological behaviours are being accumulated from a variety of sources in seemingly diverse fields, and from thousands of journals. Biological "*knowledge is evolving so rapidly that it is difficult for most scientists to assimilate and integrate the new information within their existing knowledge"* (Barnes, 2002).

- The use of **ontology is key in structuring biomedical data** (Bray, 2001) in a way that helps scientists to understand the relationships that exist between terms in a specialized area of interest, as well as to help them understand the nomenclature in areas with which they are unfamiliar.

- In biological systems, **everything is interconnected**, and ostensibly unrelated fields are related - the separation of biology into different disciplines, while useful, is artificial (Blagosklonny & Pardee, 2002).

- As an **evolving field**, biomedical informatics brings all the requirements needed to evaluate the proposed "Evolving Ontology" model.

- The complexity involved in biomedical discovery is significant (Barnes, 2002) so there is a need to find tools to acquire, maintain and represent knowledge in a more understandable way.

Implementation Platform

When this research began in 2003, the Ontoweb Deliverable 1.3 (Gómez-Pérez, 2002) was the main work attempting to compile a survey of existing ontology tools. The ontology development tool section reported in that survey included tools, environments, and suites that could be used for building a new ontology from scratch or for reusing existing ontologies. The survey was also complemented with extra features such as tools for ontology documentation, ontology exportation and importation from different formats, graphical views of the ontologies built, ontology libraries, attached inference engines, and so on.

Among the 11 listed tools, the public domain tool Protégé (Noy, Fergerson, & Musen, 2000), developed at Stanford, was reported as being one of the most used and powerful ontology tools, offering a range of important functionalities (described in Chapter 5) for the development of the meta-model which supports the concept of Evolving Ontologies. Therefore Protégé was the tool selected in which to implement our conceptual framework and develop the ontological model.

In spite of Protégé being able to meet some of the ontology engineering requirements of this research, it has limited capability with respect to integration with knowledge discovery processes. This limitation has led to the development of new plug-ins able to test our conceptual framework and form the foundation for full implementation and refinement of this research proposal. Existing plug-ins have also been used or enhanced to support the ontology driven knowledge discovery framework.

Protégé, as an open source code application, also provides an opportunity to test and to expose the thesis' tools to a wider audience as well as to transfer the developed technology.

Protégé was also favoured due to its industry partnership with Daimler Chrysler, the main partner for the industry life cycle adopted in this research – Cross Industry Standard Process for Data Mining (CRISP-DM). Additional technical reasons for selecting Protégé as the preferred ontology tool are discussed in Chapter 5

1.4.2. Requirements Identification

This section defines the requirements established to evaluate the main outcomes of this research (Knowledge representation, Conceptual Framework, Ontology Driven Methodology and Ontology Engineering tool). These requirements were defined taking into consideration Nunamaker's systems development research methodology, and the needs of the biomedical ontology engineering community, and were also informed by requirements derived from literature and from users and experts conducting research on brain diseases.

Knowledge Representation

A meta-knowledge model able to cope with the evolving characteristics of a domain should be built to support several knowledge representation requirements. The user should be able to:

- Define different sources of information (Glass & Karopka, 2002; D'Hollosy, De Vries Robbe, Mars, Witjes, Debruyne & Wijkstra, 1996; Barnes, 2002; Ashburner & Ball, 2000) and maintain a traceable link to original sources (Marchiori, 2002; Kauppinen & Hyvönen, 2006). This requirement is essential for the acquisition of different information sources in the context of dynamic domains.

- Represent uncertainty ((Heflin, Hendler & Luke, 1999; Flouris & Plexousakis, 2005; Avery & Yearwood, 2004). This requirement reflects the need to include metadata about the quality of the information acquired by both experts and modelling techniques in order to define uncertainty.

- Acquire new knowledge while establishing degrees of knowledge acceptance based on external measures e.g. evidentiary strength, statistical significance (Stojanovic, Maedche, Stojanovic & Studer, 2003; Peter, Olga & Sven, 2007; Bairoch, Boeckmann, Ferro & Gasteiger, 2004).

- Annotate any concept with domain specific and general meta-data (Burgun, Botti, Fieschi & Le Beux, 1999; Cheah, & Abidi, 2001; DCMI. 2007; Bodenreider, 2001). This capability is essential to ensure that the provenance of change can be noted and held persistent, and so that decisions can be made (and their rationale recorded) regarding inclusion or exclusion of concepts in any given analysis.

Conceptual Framework

An extensible conceptual framework able to integrate both ontology engineering and knowledge discovery in databases should be developed. The user/framework should be able to:

- Reuse knowledge across multiple applications and multiple forms of analysis (Spyns, Meersman & Jarrar, 2002; Noy, Fergerson & Musen, 2000; Gruber, 1993a).

- Support different (and presently unknown) knowledge discovery tasks (Yoon, Henschen, Park & Makki, 1999; Garcia, Ferraz & Pinto, 2006). This requirement is essential for the development of a closed loop approach in the context of ontology engineering and KDD integration.

- Acquire knowledge using data mining techniques (Jinze, Wei & Jiong, 2004; Honavar, Andorf, Caragea, Silvescu, Reinoso-Castillo & Dobbs, 2001; Glass & Karopka, 2002). This capability is needed for the creation of a hybrid solution that can integrate expert knowledge and knowledge acquired using machine learning techniques.

- Define a hybrid life cycle integration of both otology engineering and KDD processes (Gregory, 2000; Fayyad & Uthurusamy, 2002; Giarratano & Riley, 2004).

- Use navigation, visualisation and a query language to support the hybrid life cycle (Gandon, 2006; Gómez-Perez, 1999; Gómez-Pérez, 2002; Wang & Gottgtroy, 2006) in a way that supports the user in their interactions with the knowledge.

Ontology Driven Methodology

It is important that this research establish reusable processes, methods and tools in the form of an ontology engineering methodology which should also respect the requirements identified in the previous sections. All requirements defined in this and the following subsection are related to the research goal to create a methodology and develop tools able to use computer models in concert with human knowledge to test hypotheses and to validate and integrate knowledge that may be created by different sources with diverse intentions but that when linked can promote the discovery of new knowledge. The user/methodology should then be able to:

- Integrate different databases to suit specific and dynamic needs requiring different knowledge content.

- Use manual, automatic and semi-automatic knowledge acquisition tools (Falconer, Noy & Storey, 2006; Gómez-Pérez, 2003).

- Follow a hybrid ontology driven KDD Life Cycle so that the knowledge base continues to evolve and so that the closed loop is achieved.

- Suggest a conceptual modelling technique so that a guideline on how to acquire expert and ontological knowledge can be applied to the methodology in real-world problems (Gómez-Pérez, Fernández-López & Corcho, 2004; Gottgtroy, 2001).

Ontology Engineering Tool

In order for the framework to be used in practice a set of tools able to support the proposed conceptual framework must be integrated and developed in order to validate the research. These tools should:

- Provide support for all stages of the entire ODKD hybrid life cycle.

- Include navigation, visualization and query support to support the user in their interactions with the knowledge.

- Integrate in a flexible manner with different data mining workbenches so that knowledge can be acquired from multiple sources.

- Be extensible in order to support the addition of new methods and tools to the proposed ontology driven framework.

- Be reusable across multiple application areas.

- Be published in the public domain to maximise use.

- Integrate and reuse the best tools and practices in the ontology engineering field.

The system's criteria defined for the ontology engineering tool are briefly described here in order to support the research design evaluation. For example, the methodology must be supported by novel developed, extended or adopted tools.

Further and specific system requirements related to plug-ins developed in this research are defined in Chapter 5 through the requirement specification of the ontology driven knowledge discovery framework. They are then used to validate the contribution of this thesis in Chapter 8 – Contribution/ Future research.

1.5. Summary

This chapter began with a characterization of the thesis domain, problems and research questions. It then presents a multi-methodological and iterative research approach able to cope with the plurality of research fields and with the requirements for the development of a new methodology to integrate ontologies in the context of knowledge discovery in databases. It then argues from the literature that System Development is a valid and viable research methodology for the development of the proposed ontology driven knowledge discovery conceptual framework and its supportive tools. This is followed by descriptions of the research process, scope of the work, design framework and implementation requirements.

The remaining chapters present the development of the thesis and are structured as follows:

Chapter 2 introduces the main elements of this research: ontology, ontology engineering, knowledge discovery in database, and ontology and knowledge discovery integration. These elements form the basis for the more specific research topics addressed in each of the following chapters.

Chapter 3 concentrates on the development of an ontological representation able to cope with the evolving characteristics of dynamic domains. The core question investigated in this chapter is: can ontologies ‘evolve’? The “Evolving Ontology” definition is presented followed by the description of the conceptual model. A case study for a Leukaemia ontology is then explored.

Chapter 4 presents the integration of ontology engineering and the knowledge-discovery-in-databases process. It shows how this thesis addresses the second research question through the development of an ontology-driven knowledge discovery process. We describe each step of the process, its integration with the CRISP-DM life cycle and the methodology used.

Chapter 5 describes the implementation of the framework. The selected ontology environment and its characteristics are presented. The set of tools developed and adapted to support the conceptual framework are presented and a comparison of the requirements for knowledge discovery in databases workbenches is conducted and reported.

Chapter 6 demonstrates the utility of the proposed methods and tools in a biomedical case study.

The evaluation of the research process and outcomes are presented in chapter 7. The framework and tools are assessed against the criteria and requirements defined in the research design along with a comparison with the requirements for next generation KDD tools. Recommendations for practice are drawn for further development of ontologies and their integration with knowledge discovery workbenches using the ontology-driven knowledge discovery process and tools.

Finally, Chapter 8 presents conclusions based on the findings of the thesis, analyzing the impact of the study in the ontology engineering field, stressing its strengths and its limitations. Implications and future directions for research are presented.

Chapter 2

Research literature review

2.1. Introduction

Ontology is a multidisciplinary notion related to various fields including philosophy, artificial intelligence, cognitive science, and conceptual modelling. However, when considered as an engineering discipline, it is directly related to software engineering and applied information systems.

Although knowledge discovery in databases has inherited relationships with the data mining domain comprising pattern recognition, statistics, database management and artificial intelligence, KDD is also directly related to information systems and software engineering when it is considered as a process whose main goal is to develop tools to make sense of data and support decision making. KDD as an information systems research topic is clearly referenced in the Association for Computing

Machinery (ACM) classification system for the computing field where data mining is a subject of database applications that, in turn, is related to database management and to the information systems field (ACM, 1998).

In order to cope with this diversity/richness of research domains, while having information systems as a common discipline, this chapter describes the main research elements of the thesis which span through ontologies, knowledge discovery and the relationship between these two research fields. Although most elements are reviewed in this Chapter, some specific related topics are briefly described within subsequent chapters in order to improve the reading of the thesis. The next sections describe the most important and general elements of the research.

2.2. Elements of the research

2.2.1. Ontology

Although ontology, as a computer science topic, has just a short history, philosophers have been studying these ideas for centuries. Human beings have always been regarded as having a natural desire to know. Questions such as; what is knowledge? What is the relationship between external perception of the world and personal models? These are only some of the basic questions that have been investigated by philosophers and psychologists.

Technological advances brought new questions to this scenario: can we represent knowledge on a computer? Can we share knowledge? Attached to these questions is our wish to intelligently use knowledge to take competitive advantage in businesses, to improve society for individuals and groups, to search for information on the web, to extract knowledge from data and so forth, and this results in pressure to find new ways of building, maintaining and sharing knowledge among people, machines and processes.

The notion of reusable pieces of knowledge had its starting point in 1991 when the Defence Advanced Research Projects Agency (DARPA) (Gómez-Pérez, Fernández-López, & Corcho, 2004) proposed a new way of building knowledge based systems. Their proposal encompassed the idea of assembling reusable components and the interoperability of different reasoning. That way *"declarative knowledge, problem-solving techniques and reasoning services would all be shared among systems"*. This movement was followed by other initiatives within the knowledge engineering community (Gómez-Perez, 1999) which built the conceptual bases on which was developed the idea of ontology within computer science.

Ontology is often defined in artificial intelligence as a specification of a conceptualization (Gruber, 1993b). Ontology specifies at a high level what classes of concepts are introduced to a domain and what classes of relations exist between these concept classes. Ontology captures the intrinsic conceptual structure of a domain. *"Given a domain, its ontology forms the heart of any system of knowledge representation for that domain"* (Chandrasekaran, Josephson, & Benjamins, 1999). Ontology is said to be a unifying framework for different viewpoints (Uschold & Gruinger, 1996). It is an in-principle agreement which enables communication between people, people and systems, and between systems.

Ontologies (Gruber, 1993a; Sowa, 2002) have been used to provide a common conceptual framework for several systems, notably in bioinformatics (Glass & Karopka, 2002; Köhler & Schulze-Kremer, 2002; Schulze-Kremer, 2002), medical decision support systems (Burgun, Botti, Fieschi, & Le Beux, 1999; Cheah & Abidi, 2001), and knowledge management (M. Peter & Hans, 2004; Sure, Staab, & Studer, 2002).

Ontologies have been widely investigated by the knowledge engineering, artificial intelligence, philosophy and computer science communities, emerging as an important research topic at the end of 1990's and beginning of this century. The incredible advance of the internet, the huge amount of data produced and stored in enterprise databases, the genome project among others are producing new requirements for the evolution of ontologies and the creation of an ontology engineering field.

2.2.2. Ontology Engineering

Ontology engineering relates to the systematisation of the activities involved in the process of building ontologies. It is concerned with the development process, the life cycle, methodologies and tools to support the development and use of ontologies.

Mizoguchi and Ikeda (2006) argue that the most important challenge of the ontology engineering field is to make implicit knowledge explicit. "*Ontology in philosophy contributes to understanding of the existence. While it is acceptable as science, its contribution to engineering is not enough, it is not for ontology engineering which has to demonstrate the practical utility of ontology".*

Software and databases, for example, can be considered and/or perceived as having ontologies within their models. However, their models are implicit conceptualizations (Gottgtroy, Kasabov, & MacDonell, 2003a). The knowledge management field, for example, is calling for technologies which will enable organizations to build their corporate memory, that is, an explicit and consistent representation of knowledge and information within an organization, which facilitates access, sharing and reuse by members of the organization and preserves knowledge within an organization (Fensel et al., 2000).

An explicit representation of an ontology is critical to its purpose of making machine readable sharable knowledge. In fact, by making at least some aspects of the conceptualizations explicit to the system, it can improve them through inferences, exploiting these explicit partial conceptualizations of our reality. Guarino and Welty state: *"The philosophical discipline of Ontology is evolving towards an engineering discipline and in this evolution the need for a principled methodology has clearly arisen"* (Guarino & Welty, 2000).

Ontology, as an engineering discipline, searches for methods and tools to enable users to build ontological representations of a domain and its problems. As a young field, instead of searching for the best possible theory for all problems, ontology engineering is currently investigating theory that can be applied to specific problems. Although none of the available research has ever developed a general theory that can solve all problems, ontology engineering has been successful in finding adequate methods that can deal with specific but complex problems (Sowa, 2005). Ontology engineering bridges the gap between the real world and models and theories. Figure 2-1 shows this engineering process.

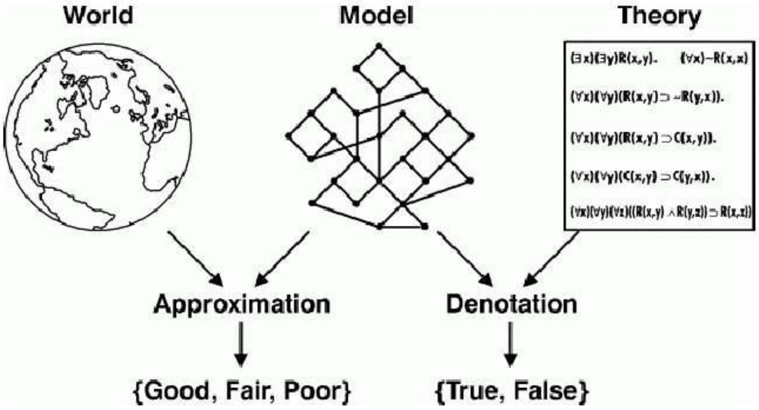

World
Model
Theory
$(\exists x)(\exists y)R(x,y).$
$(\forall x)\sim R(x,x)$
$(\forall x)(\forall y)(R(x,y) \supset \sim R(y,x)).$
$(\forall x)(\forall y)(R(x,y) \supset C(x,y)).$
$(\forall x)(\forall y)(C(x,y) \supset C(y,x)).$
$(\forall x)(\forall y)(\forall z)((R(x,y) \wedge R(y,z)) \supset R(x,z))$
Approximation
Denotation
{Good, Fair, Poor}
{True, False}

Figure 2-1 -World, Model and Theory (Sowa, 2005)

A large number of ontologies have been developed by different groups, with different approaches, and with different methods and techniques (Gandon, 2006; Gómez-Perez, 1999). Sowa argues that "*if the world had a unique decomposition into discrete objects and relations, the world itself would be a universal model, of which all accurate models would be subsets".* This plurality of ontology engineering approaches is in part a consequence of different problem domains, intentions and different limitations within ontology engineering tasks. Sowa's argument is based on the fact of that even the best models are approximations of a limited aspect of the world for a specific purpose (Sowa, 2005).

In a nutshell, ontology engineering is responsible for the investigation of frameworks able to translate the real world into useful ontological representations, that is, building a body of knowledge taking into consideration the engineering saying: *"All models are wrong, but some are useful"* (Sowa, 2002). Such an understanding is adopted here in that a generic framework is developed but is assessed using models from specific cases.

2.2.3. Knowledge discovery in databases

Knowledge Discovery in Databases (KDD) is an iterative process based on the analysis of current facts or data, pre-processing to clean and transform that data, application of mining algorithms, and deployment using the mining results on new data.

The KDD field is concerned with the development of methods and techniques for making sense of data (Fayyad, 1996). KDD addresses the problem of mapping low-level data, mainly from large databases, which can be too voluminous or have a high dimensionality, into other forms that might be more understandable and/or manageable, for example, more abstract, or more useful in terms of a data mining task. KDD’s main goal is to apply methods and tools to transform data into knowledge.

KDD is an interdisciplinary field which finds intersection with diverse research fields such as databases, machine learning, artificial intelligence, pattern recognition, statistics, knowledge acquisition, data visualization, and so forth.

The name *knowledge discovery in databases* was introduced at the first Knowledge Discovery in Databases workshop in 1989 by Gregory Piatetsky-Shapiro. His intention was to emphasize that knowledge is the end product of a data-driven discovery process. Although KDD has been widely used in the artificial intelligence and machine-learning fields, data mining is still used as its synonym by industry and commercial applications.

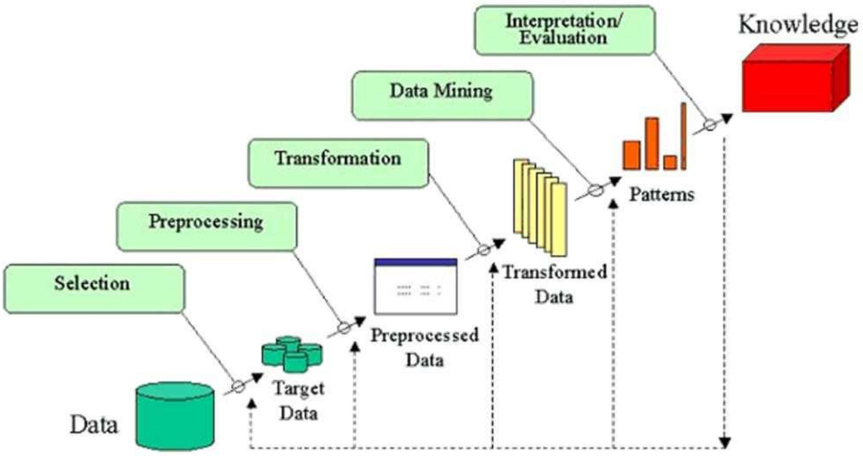


Figure 2-2- Knowledge Discovery in Databases (Fayyad, 1996).

This thesis follows the broader notion of KDD as a process as presented by Fayyad et al: *"KDD refers to the process of discovering useful knowledge from data, and data mining refers to a particular step in this process. Data mining is concerned with the application of specific algorithms for extracting patterns from data. The additional steps in the KDD process* [Figure 2-2], *such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, are essential to ensure that useful knowledge is derived from the data"* (Fayyad, 1996).

KDD generations

The beginning of the century has been marked by constant KDD evaluations (Fayyad, Piatetsky-Shapiro, & Uthurusamy, 2003; Gregory, 2000; Mihael, 2002). In

2003, for example, at the KDD Workshop 2003, a panel composed of representatives from academia and industry discussed where the field stood, what its current challenges were? What was coming next? And so forth. This analysis was motivated by almost fifteen years of both commercial and academic development.

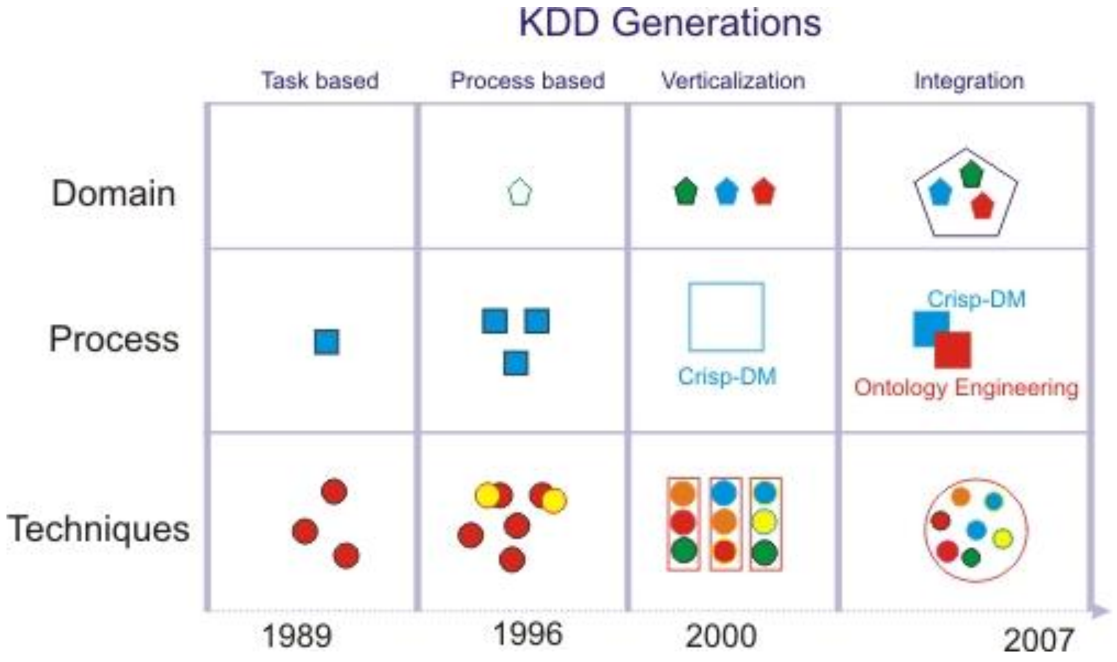


Figure 2-3- KDD evolution.

One product of this analysis is a historical timeline of KDD generations. It is argued that KDD can be classified as having four main stages/generations (Fayyad, Piatetsky-Shapiro, & Uthurusamy, 2003) as depicted in Figure 2-3.

The first generation (1989) was focused on single task data mining tools and techniques such as the C4.5 decision tree algorithm (Quinlan, 1993). They needed a high level of expertise to be used effectively and considered KDD as a one shot process based on the requirements of the algorithm used (Mannila, 1996).

The second-generation (1995) can be associated with the idea of process and its supporting tools. KDD was defined as *"the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"* (Fayyad, Piatetsky-Shapiro, Smyth, & Uthurusamy, 1996). KDD as a systematic approach was pursued by both academia and industry in the development of products such as IBM Intelligent Miner and SPSS Clementine.

The third generation is associated with specialized applications to solve domain specific problems. Around 200 tools or workbenches (http://www.kdnuggets.com/) that could perform several tasks were available at the beginning of the century. These represent solutions to business problems which increasingly embed data mining technology, often in a hidden fashion, into the application.

The increasing number of "vertical solutions" for different problem domains and the possibility of integrating and reusing their findings in different knowledge discovery tasks led to the call for a more holistic view of the KDD process. This new generation of KDD tools should be able to discover knowledge by selecting and combining several sources of information, such as information generated by machine learning algorithms and their integration with human-knowledge, and applying the most suitable techniques for domain-specific problems in order to generate new hypotheses and build links across domains (Fayyad, Piatetsky-Shapiro, & Uthurusamy, 2003) motivating the research presented here.

2.2.4. Knowledge Discovery Life Cycle

There are several life cycles for knowledge discovery proposed in the literature. Most of them reflect the background of their proponents, originating as they did in the database community (Fayyad, 1996), in the artificial intelligence community (Schreiber et al., 1999), in the decision support community (Sprague & Carlson, 1982), and in the information systems community.

This research takes into consideration different aspects of these life cycles in order to develop a hybrid life cycle that incorporates the best practices developed in the ontology engineering field as well as the best industry practice in the knowledge discovery process. To this end this research adopts the Cross Industry Standard Process for Data Mining (CRISP-DM) as the basic KDD life cycle and considers some of the requirements for the next generation of KDD processes and tools in order to create the skeleton of the Ontology Driven Knowledge Discovery process. CRISP-DM is briefly described next.

CRISP-DM

The Cross Industry Standard Process for Data Mining - CRISP-DM (Chapman et al., 2000) is a comprehensive methodology and process model that defines at different levels a complete mapping for a knowledge discovery in database task. Although called a process for data mining, CRISP-DM has been adopted by different vendors as industry standard for knowledge discovery tasks. It breaks down the life cycle of a KDD process into six phases: *business understanding*, *data understanding*, *data preparation*, *modelling*, *evaluation*, and *deployment*.

The process was born from the demand for a standard process by the industry in the 1990s and became both an industry standard adopted by most of the main vendors (SAS, SPSS, Oracle and Microsoft) as well as being adopted in different academic research, especially by innovative research that was investigating the adoption of previous knowledge in a knowledge discovery process (Cespivova, 2004).

The current process model provides an overview of the life cycle of a KDD project. It contains the phases of a project, their respective tasks, and relationships between these tasks. The life cycle of a knowledge engineering task, or data mining task as referenced by the CRISP-DM methodology, consists of six phases as shown in Figure 2-4. The sequence of the phases is not rigid. Moving back and forth between different phases is always necessary. The phase or particular task of a phase which has to be performed next is dependent on the outcome of the preceding phase. The smaller arrows indicate the most important and frequent dependencies between phases.

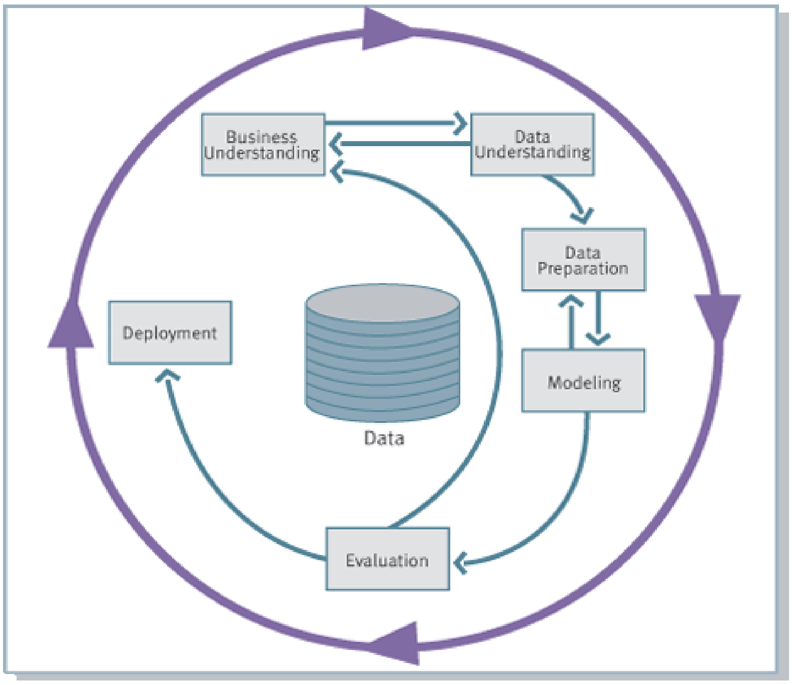


Figure 2-4 - Phases of the CRISP-DM reference model (Chapman et al., 2000).

The outer circle in Figure 2-4 symbolizes the cyclical nature of the data mining process itself. Data mining is not finished once a solution is deployed. The lessons learned during the process and from the solution deployed can trigger new, often more focused questions. Subsequent data mining processes will benefit from the experience of previous ones. Given its central role in the work presented here, each phase of the CRISP-DM methodology is briefly outlined:

Business understanding

This initial business centred phase focuses on understanding the project objectives and requirements from a business perspective. The business understanding phase involves several key steps, including determining business objectives, gaining domain knowledge, assessing the situation, determining the data mining goals, and producing the project plan.

Data understanding

The data understanding phase starts with an initial data collection and proceeds with activities that will enable familiarization with the data, identification of data quality problems, first insights into the data or detection of interesting subsets to form hypotheses for hidden information.

Data preparation

The data preparation phase covers all necessary activities for constructing the final dataset (data that will be fed into the modelling tool(s)) from the initial raw data. Data preparation tasks are likely to be performed multiple times and not in any prescribed order. Tasks include table, record and attribute selection as well as transformation and cleaning of data for modelling tool input.

Modelling

In this phase, various modelling techniques are selected and applied and their parameters are calibrated to optimum values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements of the form of the data. Therefore, stepping back to the data preparation phase is often necessary at this point.

Evaluation

At this stage of the project users have built a model (or models) that shows high quality extracted knowledge from a data analysis perspective. Therefore, before proceeding to the final deployment of the model, it is important to more thoroughly evaluate the model and review the steps executed to construct it to be certain it properly achieves the KDD exercise objectives. A key objective in this phase is to determine if there is some important issue that has not been sufficiently addressed. At the end of this phase, an evaluation of the quality of the data mining results should be undertaken in order to decide on the use of the model or its reconstruction.

Deployment

The creation of a model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained needs to be organized and presented in a way that the user can effectively utilise and act on. It often involves applying ‘live’ models within an organization’s decision-making processes, as, for example, repeated scoring of the evidence found in a knowledge discovery process. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise.

2.2.5. Ontology and Knowledge Discovery

From a philosophical point of view, discovery can be defined as “*the act of becoming aware of something previously existing but unknown*” (Noe, 2002). This broad definition includes both kinds of scientific discovery: factual and conceptual. The former typically occurs during the investigation of current ‘known’ facts or data. The latter emerges from different points of view of ‘unknown’ facts or data.

Ontologies might be used to facilitate both forms of scientific discovery in providing a common framework of understanding for several systems and problem solving methods, and, at the same time, connecting different facts and points of view within a peer-review approach.

The integration of ontologies and knowledge discovery in databases in this research is aimed at the extraction of knowledge from data and knowledge sharing in a hybrid system which integrates machine and human knowledge. Thus it is important to define and note the differences between data, information and knowledge. The thesis follows Devlin’s definition (Devlin, 2001):

- Data = Recorded facts

- Information = Data + meaning

- Knowledge = Internalized information + ability to utilize the information

Considering Devlin, we extend the previous discovery definition to one that is more strongly related to the discovery *of knowledge – "the act of adding meaning to data in order to share knowledge and become aware of the "unknown""*.

Although this definition helps to give meaning to a process that is able to leverage the advantages of ontology engineering and knowledge discovery, it cannot help in the evaluation of the usefulness, "interestingness" (Fayyad & Uthurusamy, 2002) and intelligent support that the integration of ontologies and knowledge discovery in databases (KDD) brings to the decision making process. Deciding what should be considered knowledge is a fairly difficult (and perhaps domain-sensitive) task which may involve a combination of technical measurements and human-intelligence under some philosophical constraints. The outcome of this exploration guides the iterative characteristics of the KDD process. It might result in changes to any of the preceding steps and a restarting of the entire process.

This thesis considers both technological and philosophical aspects of this evaluation. From a usefulness perspective we use Dhar's intelligence density framework (Vasant & Roger, 1997) as a guideline to allow us to measure the level of intelligent support provided by the integration of ontology and knowledge discovery processes. The methodology and framework developed in this research then support a closed loop life cycle which in turn supports the decision making process. Different technical measurements such as statistical measures, visualization, evidence and so forth, and which are covered further in Chapters 3 and 5, are also combined in the framework.

Adding value to data enables organizations, decision makers, decision systems, or even a software agent to ‘know more’ about a domain – its environment, circumstances under which decisions are made, and so on. Increased intelligence in a KDD process means improved KDD results in terms of transforming data into knowledge.

Intelligence density is defined by Dhar as a measure of productivity. It is the *“amount of useful decision support information that a decision maker gets from using the output of an analytic system”*. Intelligence density supports a decision through the transformation of data into knowledge through the following steps:

- Access;
- Scrub;
- Integrate;
- Transform;
- Discover;
- Learn. (Vasant & Roger, 1997)

By adding value to data from the Access step up to the Learning step we can transform data into knowledge. Thus to evaluate a system we might measure the level of support that it can give to such a decision making process.

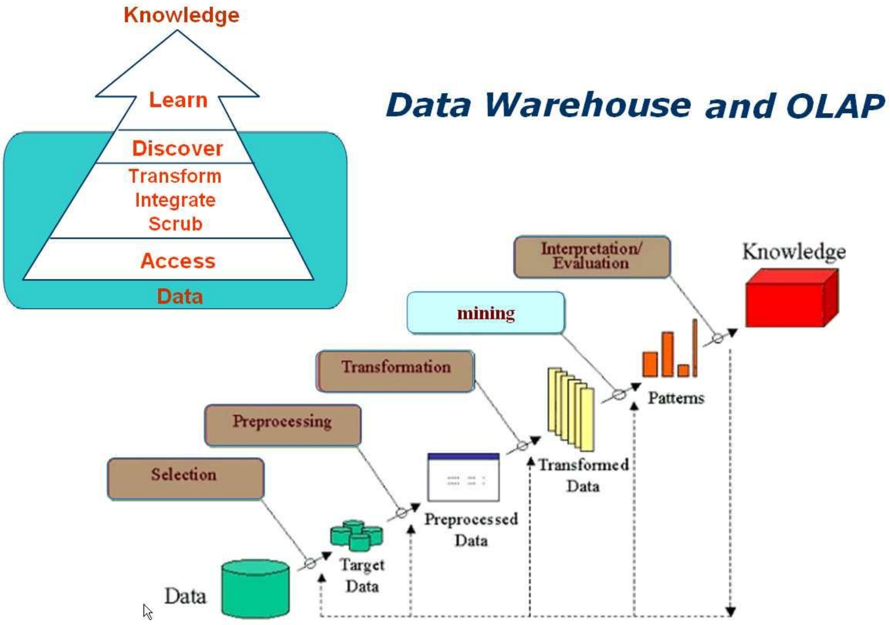


Figure 2-5 – An intelligence density diagram and a KDD process, masked with shadow boxes, representing all data warehouse and OLAP steps

A data warehouse application, for instance, is able to provide Access, Scrub, Integrate, and Transform functionalities, but it cannot in itself discover. To increase intelligence and discover useful information we must use additional tools and techniques in conjunction with data warehouses. For example, when using on-line analytical processing (OLAP) tools to analyse multi-dimensional models represented by a data warehouse (Figure 2-5), the system can identify unknown information; however it cannot learn or acquire knowledge because it is not mining or storing the meaning of the discovered facts in any repository, such as an ontology or knowledge base to be further used or reused in a new knowledge discovery scenario.

This thesis' definition of knowledge discovery – "*the act of adding meaning to data in order to share knowledge and become aware of the "unknown""* – aims to encompass both the intelligence density framework and Devlin's definition in order to evaluate the degree of intelligent support delivered by the ontology-driven knowledge discovery process.

2.2.6. Domain Knowledge and KDD

Domain knowledge has been a topic of strong and enduring research interest since the advent of expert systems in the early 1980s. In spite of some very successful applications, the initial excitement and high expectations raised within the artificial intelligence community by expert systems was gradually undermined by the so called knowledge acquisition bottleneck (Giarratano & Riley, 2004).

In response to this bottleneck, the machine-learning field gained attention as a means of overcoming the high and costly dependence on experts' knowledge. Langley and Simon cited in (Pat & Herbert, 1995), for example, instead of taking a more integrative approach as suggested by Brachman and Anand (Brachman & Anand, 1996), proposed the replacement of the *"time-consuming human activity by automatic techniques that could improve accuracy or efficiency by discovering and exploiting regularities in stored data"*. Machine learning techniques such as pattern recognition and especially those which allow rule extraction represent then a vast improvement over traditional methods of knowledge acquisition. However those techniques alone also failed to provide a one-shot solution, because they tended to substitute effectiveness for efficiency, and lost much of the ability to explain outcomes or combine expert knowledge with knowledge extracted from models such as those delivered by hybrid methods (Giarratano & Riley, 2004).

The idea of discovering ‘knowledge’ in large amounts of data is very appealing when the huge amount of data produced and stored nowadays is considered. However, finding non-trivial, useful knowledge has proved to be very challenging. Selecting the right data features that enable the extraction of high-value relationships and patterns is, for example, very dependent on an understanding of the domain and the problem.

The amount of operational data stored in different databases, the need for secure and highly scalable mechanisms along with the huge amount of unstructured data available have brought different fields of expertise to the knowledge engineering scenario. In contrast to the old knowledge-based systems approaches (Giarratano & Riley, 2004) where the key roles were those of the domain expert and the knowledge engineer, today more disciplines are playing key roles (e.g. data base experts, data warehouse developers). In consequence a broader integration is necessary in order to maximize the value obtainable from knowledge engineering.

Research has suggested different roles for and use of domain knowledge in the process of knowledge engineering relevant to this work, for instance:

- Domingos (Domingos, 1999) suggests the use of domain knowledge as the most promising approach to ensure that knowledge discovery is appropriately bounded and to avoid the common problem of data over fitting by discovered models.

- Yoon et al. (Yoon., Henschen, Park, & Makki, 1999) suggest the use of domain knowledge in various contexts and propose the following domain knowledge usage classification: *inter-field knowledge*, which describes relationships among attributes, *category domain knowledge*, which presents useful categories for the domains of the attributes, and *correlation domain knowledge* that suggests correlations among attributes.

- Sarabjot et al (Sarabjot, David, & John, 1995) also identify the use of domain knowledge in KDD tasks: for a description of attribute relationship rules, for hierarchical generalization trees and constraints. An example of the latter is a specification of degrees of confidence in the different sources of evidence.

- Jinze et al (Jinze, Wei, & Jiong, 2004) present domain knowledge as a hierarchical ontology used to guide a clustering task. They argue that while domain knowledge is always the best way to justify clustering, few clustering algorithms take domain knowledge into consideration. They present a framework that directly incorporates domain knowledge into the clustering process, yielding a set of clusters with a strong ontological implication. On the other hand, domain knowledge can also undermine the power of unsupervised learning which can help in the acquisition of new and perhaps unexpected knowledge.

- Phillips and Buchanan (Phillips & Buchanan, 2001) propose an ontology-guided methodology to gradually accumulate knowledge from databases in order to gain domain knowledge in the iterative process of a KDD task.

In spite of the increase in investigation of the integration of ontologies and KDD, most approaches concentrate only on the data mining phase of the knowledge discovery process while the role of ontologies in other phases of the knowledge discovery process (as addressed here) is not considered.

2.2.7. Ontologies, Data Models and Data Warehouses

Ontologies, as employed currently in computer science, are computer-based resources that represent agreed domain semantics. Unlike data models, the fundamental asset of ontologies is their relative independence of particular applications, i.e. an ontology consists of relatively generic knowledge that can be reused by different kinds of applications/tasks (Spyns, Meersman, & Jarrar, 2002).

A data model represents the structure and integrity of the data elements of what is in principle a ‘single’ (limited set of), specific enterprise application(s) in which the data model will be used. Therefore, the conceptualisation and the vocabulary of a data model are not intended, *a priori*, to be shared by other applications (Gottgtroy, Kasabov, & MacDonell, 2003b). Moreover, the massive amount of new data and facts produced daily can increase the structural complexity of data models, which, in consequence, demands constant data model updating which is an expensive and time consuming process.

Data warehousing technology has been one of the database community responses to this problem. A data warehouse model aims to organize the data to suit prospective investigation and add ‘strategic meaning’ to the relational data models. The resulting multi-dimensional model is intended to enable data modellers to represent multiple perspectives of the same operational data. However, despite the improvements achieved by the data warehousing technology in adding meaning to data models, the problem of dependence of a specific application is still present in the multi-dimensional model.

On the other hand, both ontology models and data models have similarities in terms of scope and task. They are context-dependent knowledge representations, that is, there isn’t a strict line between generic and specific knowledge when building the models (Gottgtroy, Kasabov, & MacDonell, 2003b). Furthermore, modelling techniques for both data and ontology building are knowledge acquisition intensive tasks and the resultant models represent a partial account of conceptualisations.

Opportunities arise from the similarities in and advantages of both data modelling and ontology engineering. Data warehouse technology, for example, can contribute the well-developed on-line analytical processing and visualization techniques that enable strategic analysis of ontologies. Ontologies can add an application-free perspective to a model, adding different dimensions to an otherwise narrowly-modelled problem.

2.2.8. Ontology and KDD integration

There are currently three approaches being investigated in the ontology and KDD integration emergent research: Onto4KDD, KDD4Onto, and one approach that integrates both of the previous approaches, named here Onto4KDD4Onto.

Onto4KDD is defined as the application of ontologies in order to improve the KDD process. For example, domain ontologies can be used to improve the understanding of a problem and support hypothesis-driven analysis and discovery approaches. Onto4KDD has been employed extensively in biomedical informatics research (Cespivova, 2004) where a huge amount of DNA sequence information is made available daily and thousands of projects need to be annotated. At the same time, millions of potentially relevant facts are being accumulated from a variety of sources in seemingly unrelated fields and are being made available either immediately on the internet or published in hundreds of journals. Biological knowledge is evolving so rapidly that it is difficult for most scientists to assimilate all information. Gene Ontology (GO) (Ashburner & Ball, 2000; GO, 2006), for example, has been used to "produce a controlled vocabulary that can be applied to all organisms even if knowledge of genes and proteins is changing". GO is the basis for classification systems that address the problem of linking biology knowledge and evolving literature, such as GO-KDS (TWO, 2006) and DiscoveryInsight (Biowisdom, 2003).

In contrast, KDD4Onto approaches are focused on the application of mining techniques in order to automatically or semi-automatically acquire knowledge from data. Ontologies tend to be built from text (Gómez-Pérez, 2003) by an integration of lexicons, taxonomies, and other ontologies, in the natural language processing and computational linguistics areas. Another motivation for the development of text based learning techniques comes from the Semantic Web research where ontology acquisition from text is a significant issue and has been well documented in recent years.

Another research focus for knowledge discovery is Evolving Connectionist Systems (ECoS) (Watts, 2009). ECoS is a class of Artificial Neural Network (ANN) architectures with a learning algorithm that modifies the structure of the network as training examples are presented. The knowledge discovery feature of ECoS extracts a set of symbolic rules that mimic the behaviour of the ANN. This feature is important for systems where explanation of rules is important in safety-critical domains such as the identification of cancer tissues from gene expression in microarrays (Futschik, Reeve, & Kasabov, 2003). ECoS along with the Computational Neural Genetic Model presented in the case study (Chapter 6) were investigated as candidates for the biomedical case study developed in this thesis.

Although some researchers are addressing Onto4KDD or KDD4Onto, rare is the research that encompasses both perspectives.

The work presented here is an attempt to integrate both approaches. It bridges the gap between ontological engineering and knowledge discovery in databases in order to improve both processes. In particular, it analyses how data mining can assist in efficient and effective large-volume data analysis in order to build a sharable and evolving knowledge repository, while at the same time, leveraging the semantic content of ontologies to give intelligent support and improve knowledge discovery in complex and dynamic domains.

2.2.9. Understanding Change

Evolving Ontology research relates mainly to the need for methods, models and tools to support change/evolution in an ontological representation – Ontology Change. Ontology Change is a broad research area that involves several aspects, including the reasons for change and the representation of uncertain knowledge (Avery & Yearwood, 2004; Flouris & Plexousakis, 2005; Peter, Olga, & Sven, 2007; Sanchez, 2006).

This section splits the ontology change related research into three main approaches in order to better identify research linkages. The first subsection concentrates on *understanding change* and the reasons for it. The second presents the *ontology engineering topics* related to the ontology change problem covered in this research. The final part explores the *representation formalism-oriented research* and describes the approach followed in this work.

An ontological model may need to change for several reasons, such as the modification of user requirements or a change in the underlying domain. The list below compiles several reasons for change found in the ontology engineering field that have influenced this research:

- The domain of interest has changed (Stojanovic, Maedche, Stojanovic, & Studer, 2003);

- The perspective under which the domain is viewed needed to change (Noy & Klein, 2004);

- A problem is found in the original conceptualization of the domain (Haase & Stojanovic, 2005);

- Additional functionality is required according to a change in users' needs (Haase & Stojanovic, 2005);

- New information, which was previously unknown, classified or otherwise unavailable, may become available or different features of the domain may become important (Heflin, Hendler, & Luke, 1999);

- A collaborative and parallelized process of building ontologies, whose sub products (parts of the ontology) need to be combined to produce the final ontology, requires that all changes in individual 'sub-ontologies' be replicated to reach a consistent 'final' ontology (Klein & Noy, 2003);

- The 'final' state is rarely final since building an ontology is usually an ongoing process (Klein & Noy, 2003);

- Dependency that arise among distributed ontologies over which the knowledge engineer may have no control (Heflin, Hendler, & Luke, 1999) as in the case of the Semantic Web where ontologies are dependent on other ontologies;

- It may be necessary to make changes to the knowledge representation of an ontology whose terminology or representation is different from the one being used (Euzenat et al., 2004);

- It may be necessary to merge or integrate information from two or more ontologies in order to produce a more appropriate one (Sofia Pinto, Gómez-Perez, & Martins, 1999);

- The ontology may have acquired knowledge using semi-automatic or automatic learning algorithms that need to be included in the ontological model (Gómez-Pérez, 2003).

The implication is that ‘Ontology Change’ is not trivial. It can arise for a variety of reasons. It involves both technological and philosophical challenges. Even answering the research question – Can an ontology ‘evolve’? - is controversial, as the answer is highly dependent on the researcher’s background and problem domain. Adopting a purely philosophical stance, some belief systems (Gärdenfors, 1990) treat ontology as fundamental and unchanging – it is only the representation and interpretation that changes. From a process perspective, in the context of ontology engineering, the approach tends to be more pragmatic, the steps of change defined as presented in Figure 2-6.

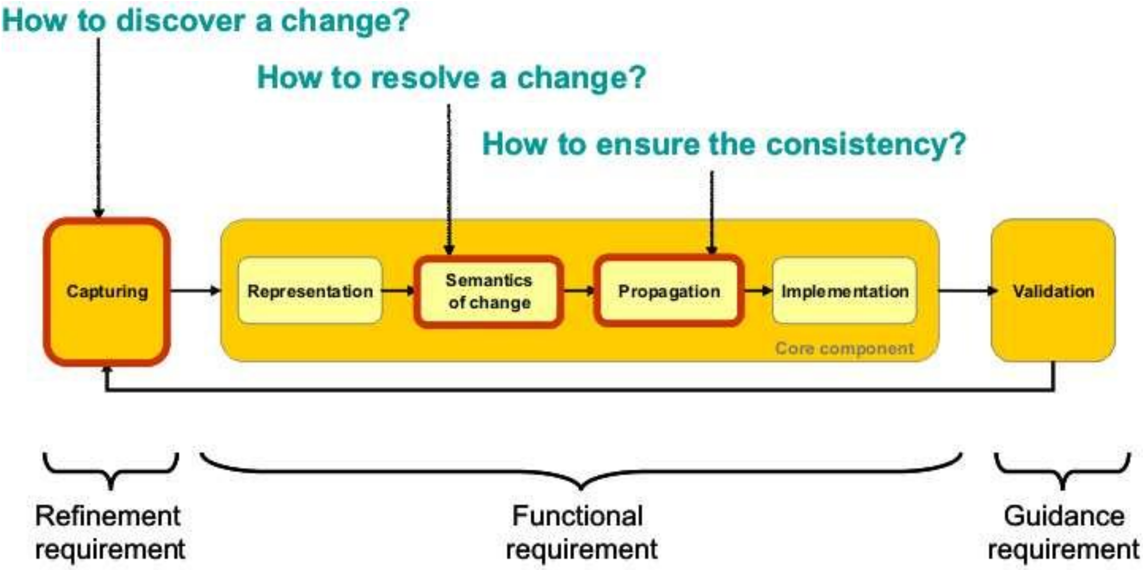


Figure 2-6- Ontology change process (Source: presentation (Sure, 2004))

Although a general answer that addresses all of the philosophical and technological requirements of the challenge of evolution might be not possible, the ontology engineering field has been developing methods and tools to solve technological elements of the problem while considering philosophical issues. This consideration comes through the development of, for example, ontology merging and alignment tools and methods which support the process of integrating different sources of information (Natalya & Mark, 2003).

The next section explores *ontology change* from an ontology engineering perspective covering topics such as ontology alignment, learning, and merging.

2.2.10. Ontology Engineering and Ontology Change

There are nine closely related ontology engineering research topics dealing with the *ontology change* problem in the literature: ontology alignment, mapping, morphism, articulation, translation, evolution, merging, integration and versioning.

“Each of them covers a small part of the complex problem of ontology change from a different view or perspective” (Flouris & Plexousakis, 2005).

According to Kauppinen (2006) ontologies will continually evolve through the help of or negotiating with other ontologies (merging and alignment), by learning from mining and/or by re-organization (evolution) with or without human support (e.g., change by ontology engineers in light of new expert knowledge).

Although this research, as later described in Chapter 5, supports ontology alignment (establishment of different kinds of mapping or links between two ontologies, thereby preserving the original ontologies) and ontology merging (generating a unique ontology from more than one original ontology) by the development or adoption of already developed ontology engineering tools, Evolving Ontology deals with changing requirements from both the perspectives of *ontology learning from instances* (the ability to acquire knowledge using KDD techniques) and *evolution* (integrating different sources of information with or without human support).

From a KDD perspective, in terms of Sure’s ontology evolution classification (Sure, 2004), this research may be classified as a “data-driven” change discovery mechanism, since it detects changes based on the analysis of the ontology instances. It addresses the knowledge acquisition problem through specific learning techniques inherited from the data mining discipline and its integration with ontology engineering.

2.2.11. Flexibility vs. formalism

Gruber, cited in (Lytras, 2004), identifies three types of ontologies: informal, semiformal and formal. Formal ontologies represent knowledge based on various forms of formal logic, such as Description logic for semantic Web (Nardi & Brachman, 2002), Conceptual Graphs (Sowa, 1984) and Simple Common Logic (Hayes & Menzel, 2004). Semi-informal and informal ontologies are based on flexible representation formalisms such as Frames (Minsky, 1974).

Rather than seeing one as superior or preferred, Gruber argues the value and roles of both formal and informal ontologies:

"I would say that all practical ontologies are semiformal, and the "sweet spot" is an ontology that specifies clearly how you can commit to it. Both the formal and informal parts should be designed to make it easy to play by the rules: the formal by automated testing and the informal by well-written documentation"

As a semantic technology (i.e. an application of techniques that support and exploit semantics of information) (Sheth & Ramakrishnan, 2000), ontologies may integrate different degrees of formality (see Figure 2-7). Depending on the scope of the problem such as in a domain industry specification, where the specification is shared and agreed between a group of applications and users, a semi-informal ontology is more suitable (as suggested in the figure below). On the other hand, formal ontologies may be more suitable for open representation where a formal specification becomes more important to define a wider shared agreement.

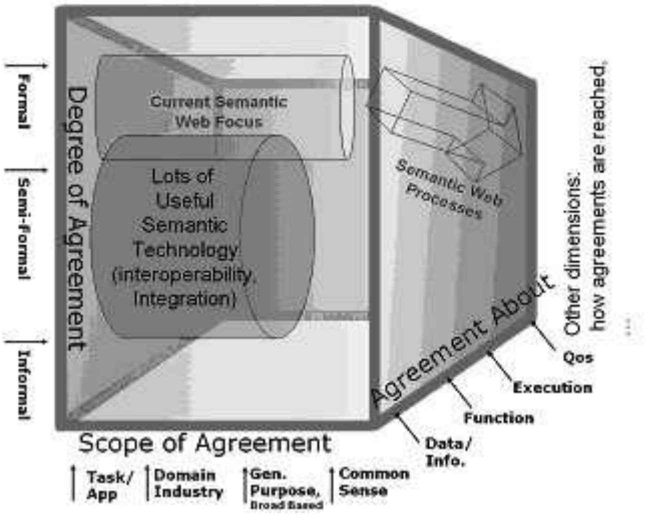

Formal
Semi-formal
Informal
Degree of Agreement
Current Semantic Web Focus
Lots of Useful Semantic Technology (interoperability, Integration)
Semantic Web Processes
Other dimensions: how agreements are reached, ...
Agreement About
Qos
Execution
Function
Data/ Info.
Scope of Agreement
Task/ App
Domain Industry
Gen. Purpose, Broad Based
Common Sense

Figure 2-7- Dimensions along which ontologies vary (source:(Sheth & Ramakrishnan, 2000))

In spite of recent interest in formal ontology research, especially due to the advances in the semantic web field, semi-formal ontologies have demonstrated very high practical value (Lytras, 2004). This is partially due to the amount of development required which can be significantly smaller for semi-formal ontologies when compared to that required for developing formal ontologies.

It is important to note that this research does not make any explicit distinction between formal and informal ontologies in the sense that the two are exclusive; rather they are complementary (Gottgtroy, Kasabov, & MacDonell, 2006). The following comments should help to clarify the research position adopted with respect to this issue: The informal parts of a specification serve to explain the knowledge in a human-accessible fashion, which makes ontologies easier to understand and powerful in terms of expressivity. The formal parts make ontologies unambiguous, enabling some degree of automated analysis and reasoning. This research has used both formal and informal languages, such as Frames and Description Logics, in its experiments depending on the particular goal of each one. As a result, it is contended here that the most effective representation is dependent on the methods used to build the ontology and on the knowledge being represented.

The research therefore adopts the notion of semantic technologies followed by industry, such as Oracle[1], IBM[2], and academia (Gruber, 1993a; Sheth & Ramakrishnan, 2000) in the sense of development of semantic-aware methods, models and tools.

In the context of knowledge discovery in databases, neither formal nor informal languages have proved dominant in supporting the ontology building process through soft computing techniques. This investigation, then, adopts Frames (Minsky, 1974) for semi-informal knowledge representation and uses the *Evolving Ontology* meta-knowledge to create an independent representational formalism layer to acquire knowledge resultant from the learning process.

This approach naturally follows the development of knowledge representation languages where definitional languages, such as Frames, were separated from process knowledge, mainly represented as rules and/or first order logics, such as those brought by formal languages, for example description logics.

[1] http://www.oracle.com/technology/tech/semantic_technologies/index.html

[2] http://www.alphaworks.ibm.com/topics/semantics

Considering that the intention of this research was to represent knowledge by means of ontology and use external inferences such as machine learning algorithms to reason, Frames was then considered as very suitable to describe concepts, especially when using defaults values to represent state or objects in a typical situation Giarratano & Riley, 2004) .Default values are useful for simulating commonsense which allow the definition of an initial knowledge when knowledge has not been acquired from machine learning techniques or experts.

Frames were also selected because of the application domain investigated and the ontology engineering tool selected. Frames have been widely used in the area of biomedical informatics knowledge discovery (Rosse & Mejino, 2003) as well as in the original formalism adopted by Protégé (Noy, Fergerson, & Musen, 2000). This fact, again, is part of the natural evolution of ontology languages and researchers' backgrounds where frame based languages were used to represent concepts loosely coupled with logics to infer knowledge while semantic net based languages such as resource description framework (RDF) and ontology web language (OWL) were developed towards more formal and closed integration with several logic flavours, such as description logics.

The main disadvantage of using this semi-informal approach is the loss of the powerful automatic classification reasoning mechanism available in logic-based representations. However as stated in (Rosse & Mejino, 2003) this loss is compensated for by the ability to build flexible knowledge structures and with the flexibility of using different methods of inference.

2.2.12. The Ontology Environment - Protégé Editor

There are several ontology engineering tools now available (Denny, 2004; Gandon, 2006). Denny (2004), for example, lists and compares more than 50 ontology editor tools available to both industry and academia. The Ontoweb Group (Gómez-Pérez, 2002) has also developed a survey on ontology tools and their capability to support different ontology engineering tasks. Other studies have focused their surveys in one area of application, such as in Lambrix (2003), where four of the most-used tools for bioinformatics projects were evaluated.

In spite of the large number of tools listed in the literature, few tools are able to fully support an ontology engineering exercise cycle as well as cover the functional requirements of a specific area of application. Therefore just a few ontology tools, including Protégé, have become industry and academic standard.

Protégé is a free, open source ontology editor and knowledge-base framework developed at Stanford Medical Informatics - Stanford University, which helps users construct domain-specific knowledge acquisition systems that knowledge experts can use to access and browse the content of knowledge bases (Knublauch, 2003; Protege, 2006a).

Protégé is built as an extensible architecture based on the development of plug-ins for the accomplishment of specific tasks. These plug-ins are modular pieces of program code that add new functionalities to the environment in well circumscribed ways (Noy, Fergerson, & Musen, 2000). Developers can contribute with new Protégé plug-ins to a library maintained on the Internet, and can freely download new plug-ins to augment the behaviour of their own knowledge-acquisition systems constructed using Protégé.

Protégé also has a number of visualization plug-ins that support knowledge sharing. This research has extended some of the available visualization plug-ins in order to integrate them in the ontology driven knowledge discovery framework. Visualization itself is an active area of research in the context of ontological knowledge navigation (Falconer, Noy, & Storey, 2006; Ontoviz, 2006),being used mainly for knowledge navigation, knowledge search, and sharing of complex knowledge networks.

Protégé is supported by a strong community of developers and academic, government and corporate users, who use it to produce knowledge base solutions in areas as diverse as biomedicine, intelligence gathering, and corporate modelling. It has currently more than 60000 registered users.

The Protégé-2000 architecture currently supports a wide range of plug-ins. There are several classes of components that developers can add to the system to expand its capabilities. User-interface widgets handle the display and input of data of particular types in domain- or task-specific ways. Alternate back ends for archival storage enable users to store knowledge bases in the formats that fit best with their environment. Utility programs for knowledge-acquisition tasks provide support for more elaborate knowledge-acquisition approaches such as accessing and importing knowledge from on-line resources and building new knowledge bases by integrating existing ones. Entire end-user applications that operate on Protégé knowledge bases can be plugged into the system as special ‘tabs’, such as those developed in this thesis to support the Ontology Driven Knowledge Discovery process.

Alongside its extensible architecture, Protégé also has a powerful knowledge model that supports different knowledge representation formalisms. As stated above, this thesis adopted the frame-based formalism. Frames (Minsky, 1974) are the principal building blocks of a frame-based knowledge base. A Protégé frame-based ontology consists of classes, slots, facets, and axioms. **Classes** are concepts in the domain of discourse. **Slots** describe properties or attributes of classes. **Facets** describe properties of slots. **Axioms** specify additional constraints. A Protégé knowledge base includes the ontology and individual instances of classes with specific values for the slots.

Protégé also has a meta-class architecture that enables extension of the knowledge-model itself. It uses the meta-class mechanism to implement its own internal class structure, which can then extend a knowledge-model in accordance with a specific problem domain, for example the novel meta-knowledge model developed in this research to support the Evolving Ontology.

Meta-classes define the representation of all frames in the system—classes, slots, facets, and individuals. A meta-class is a class whose instances are themselves classes. Every frame in Protégé is an instance of a class. Since classes are also frames, every Protégé class is an instance of another class. Therefore, every class has a dual identity: It is a subclass of a class in the class hierarchy—its super class—and it is an instance of another class—its meta-class.

This extensible architecture based on the development of plug-ins, along with the flexible and configurable knowledge model, enable the development of a set of specific tools and meta-models able to support specific tasks such as those required by the Ontology Driven Knowledge Discovery process. As such, the Protégé environment is a suitable candidate for use in this research. (It is noted, however, that other tools might also be suitable – this issue is discussed further in the final chapter of this thesis.)

2.3. Summary

This chapter presented a literature review of the core topics of this research. It focuses on the integration of ontology engineering and knowledge discovery. The next chapter describes the conceptual model aimed to build an extensible meta-knowledge able to cope with the notion of ontology change in the context of knowledge discovery.

Chapter 3

A novel meta-knowledge model for evolving ontology

This chapter presents a novel evolving ontology conceptual model. It discusses the need for flexible ontological representations for real world problems and presents the challenges of ontology change. The primary outcome of this discussion is a meta-knowledge model able to represent the dynamic and uncertain nature of domains and thereby provide constant and ongoing support for the decision making process in those domains over time. The chapter begins by defining Evolving Ontology in the context of this thesis. This is followed by the presentation of the evolving ontology conceptual model, which is composed of a meta-class, a concept meta-data, and ontology building features that support ontology change requirements. This chapter then presents a case study, summarizing research adopting the Evolving Ontology, and presents a list of associated research outcomes.

3.1. Introduction

Since their introduction into computer science, ontologies have been playing various roles in solving problems in domains such as information retrieval, e-commerce, and the semantic web. Ontologies may bring, for instance, high level data abstraction to the process of knowledge discovery in databases, enabling researchers to look less at raw data and more at the semantics of data. Although the insertion of this high level abstraction brings advantages such as the possibility for integration of different databases (Ashburner & Ball, 2000), unambiguity of term usage and so forth, it also brings some challenges.

There is a risk that building and using ontologies that are not able to represent 'unknowns', are not able to 'evolve', may lead to the "paradigm trap" (Catton & Shotton, 2004) where the ontology's representation is so strongly related to a specific paradigm in a particular field of knowledge, or its semantic representational power is so constrained by its representation formalism, that only information that fits the paradigm and/or the technology can be represented in the ontological model. This trap, from a Kuhn point of view (Kuhn, 1996), would not allow scientific progress, or from a philosophical point of view (Noe, 2002) (section 2.2.5), would constrain conceptual scientific discovery.

In order to avoid such a trap and to build a conceptual model able to integrate the advantages and best practices of ontology engineering and knowledge discovery in databases, this chapter describes an ontology meta-knowledge model able to cope with "evolving" requirements in the context of ontology learning from instances in KDD. A case study developed in the course of the research which has informed the iterative development of the meta-knowledge model is also presented at the end.

3.2. Definition of Evolving Ontology

Humans have been concerned with the description of things since the time of the ancient Greeks. Essence, existence, universals among other concepts, underlies the work on ontology in the context of philosophy. As Kant states in Vienna Logic (Sowa, 2000) "*Socrates said he was the midwife to his listeners, i.e., he made them reflect better concerning that which they already knew and become better conscious of it*". Ontology is about making things explicit; it is concerned with the description of concepts and relationships among concepts in a domain.

Although we have inherited the principles of ontological knowledge from philosophy, the artificial intelligence community has been working for many years on bringing knowledge forth and making it accessible in a machine-readable and possibly machine-understandable format. There are therefore several different definitions of ontology available in the literature. Most of these definitions reflect either the research background of their creators or are influenced by the domain in which the ontology was first used.

Gruber, among the most cited authors in the ontology field, states that, in the context of knowledge sharing, ontology is a description (like a formal specification of a program) of concepts and relationships that can exist for an agent or a community of agents (Gruber, 1993a). The formal ontology community defines ontology as an explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships held between them.

In 1998 Studer and colleagues attempted to merge the two previous definitions as follows:

"An ontology is a formal, explicit specification of a shared conceptualization." *Conceptualization* refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. *Explicit* means that the type of concepts used, and the constraints on their use are explicitly defined. *Formal* refers to the fact that the ontology should be machine-readable. *Shared* reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

There exists in the literature another group of definitions influenced by different factors such as the methods and techniques utilized to build ontology (Gottgtroy, Kasabov, & MacDonell, 2006) and the ontology's intended use. The evolving ontology definition developed in this research is also influenced by factors such as the knowledge discovery in databases field and the goals of knowledge acquisition, maintenance and sharing.

The definition of Evolving Ontology is mainly concerned with two characteristics: completeness and purpose. It takes into account the views of Guarino and colleagues (Guarino, 1998) in relation to the notion of the "partial account of a conceptualisation" and Gruber's notion of "treat among people" (Gruber, 1993a), agents, and users with some "common motive in sharing". Ontology in this research is an agreement that represents a partial shared conceptualization of the world. *Partial* means incomplete knowledge, being constructed, dynamic, changing, and evolving. *Shared* means consensual knowledge, maintained through knowledge management.

In the context of ontology driven knowledge discovery *evolving ontology* is defined as follows:

Evolving ontology is a partial shared conceptualization that supports knowledge discovery.

Evolving ontologies describe a process rather than a static model. As a consensual and partial conceptualization, ontology is an object of social pressure, different perspectives, negotiation, and mental model interpretation and so on. These requirements make imperative the ability of an ontology to change its structure and content over time as new knowledge is extracted from data mining and other learning tasks.

The Evolving Ontology model approaches integration, modification, adaptation and polymorphism by means of a meta-knowledge representation able to cope with different perspectives, uncertainty and changes in a domain. Uncertain knowledge, for example, is represented through ontology meta-knowledge and can be programmatically acquired by different modelling techniques.

The ability to express meta-knowledge of concepts is required for (Catton & Shotton, 2004):

- *Data syndication* - Systems need to keep track of provenance information, *source, author, date* and provenance chains *A said that B said C*.

- Creating annotations – Well documented concepts are better understood by users and systems.

- Restricting information usage - Information providers might want to attach information about intellectual property rights or their privacy preferences to ontologies in order to restrict the usage of published information (Marchiori, 2002).

- Expressing propositional attitudes - such as modalities and beliefs.

- Scoping assertions and logic - where logical relationships have to be captured.

The use of an evolving ontology, and in consequence its meta-knowledge and model, then enables the creation of a shared knowledge repository which are constantly enhanced by a peer-reviewed process that integrates knowledge acquired by, for example, machine learning techniques, knowledge inserted by experts, knowledge acquired by the integration and/or merging of other ontologies and so forth.

Even though this thesis focuses particularly on the requirements of the biomedical domain, the model and principles developed in this research are valid for application in others fields where the body of knowledge is jointly built and disseminated to support informed decisions. In Intelligence for example, intelligence officers grade the information acquired in the field, from databases, from environment scanning, and other sources in order to review the knowledge and create intelligence products for dissemination among their peers and for subsequent enhancement of the body of knowledge.

In the biomedical domain and especially in bioinformatics, the challenge of reusing the huge amount of data produced has been the main motivation for the development and sharing of ontologies. This sharing then brings further requirements for grading the information, merging, enhancement and so on. This thesis explores and addresses these challenges using a conceptual model which supports the evolution and enhancement requirements.

The next section presents the novel proposed Evolving Ontology Meta-Knowledge Model discussing the four main design requirements: traceability, model dependence, reified relations and meta-data, and their implementation.

3.3. Evolving Ontology Meta-Knowledge

Meta-knowledge can be defined as knowledge about knowledge. It is the equivalent of meta-data in databases. Evolving Ontology meta-knowledge is the conceptual model developed to cope with *ontology change* requirements related to knowledge discovery. [3]

The Evolving Ontology Meta-knowledge model is composed of meta-data responsible for describing concepts and relationships, a set of meta-classes developed to describe, for example, knowledge acquired from modelling techniques, an ontology environment feature responsible for integrating external ontology sources and knowledge maps responsible for storing knowledge discovered through data mining techniques. Figure 3-1 shows how the parts are related.

The meta-class is composed of several foundation schemes such as EO-Type Scheme which are used to annotate concepts within the ontology. The foundation schemes are small, simple and very stable therefore are included as metadata to better express semantics.

[3] In Chapters 5 and 6 and some figures of this chapter meta-knowledge is referred to as meta-data or meta-class as these are the terminologies employed in the ontology environment adopted

The EO-meta-class defines a set of minimal properties which is applied to every other concept or scheme in the ontology meta-model. This acts as a minimal pattern to support provenance and traceability.

The knowledge map is a set of reified relationships and concepts created to store instances acquired by machine learning techniques as well as specific relationships maps such as gene regulatory networks. This meta-class is not stable and is very customizable in order to evolve with the integration of new sources and data mining techniques.

The model integration feature enables the addition of external knowledge to the ontologies.

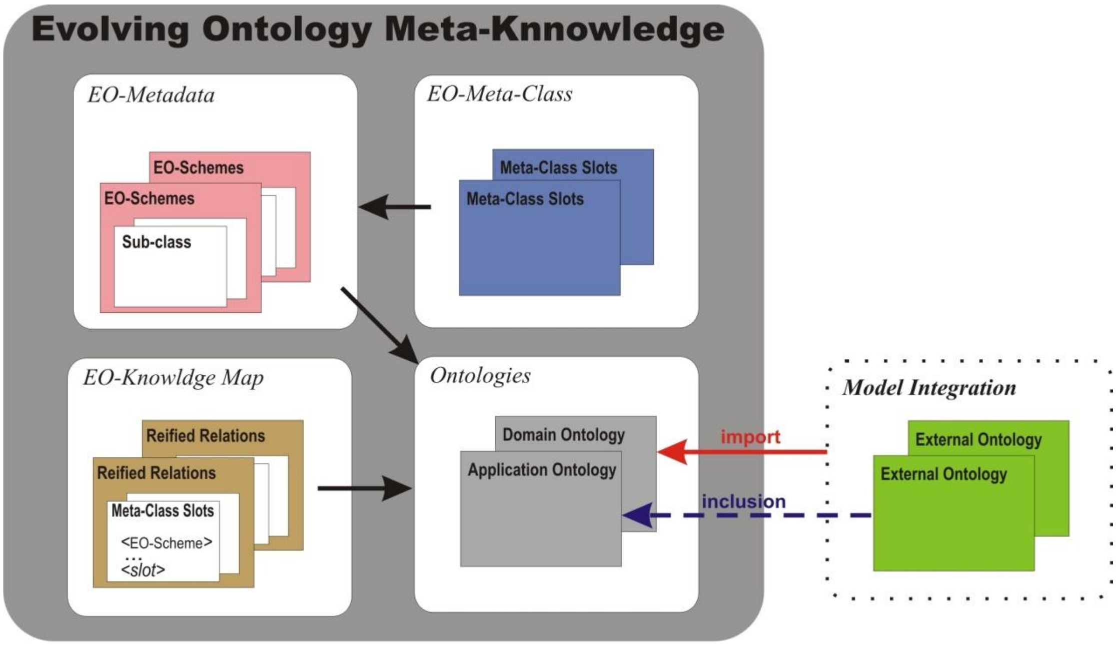


Figure 3-1 – Evolving Ontology Meta-knowledge diagram.

The next section describes the parts shown in the figure above.

3.3.1. EO-Concept_Meta-data

The Evolving Ontology concept meta-data is a set of documentation or structured information capable of describing, unambiguously, concepts and relationships within an ontological model. It maintains provenance of different aspects such as source and creation, while being invisible to the ontology user.

The extensible meta-data is represented as schemas that should aid in the interpretation of any concept and relationship built or defined within an ontology.

The goals of the EO-Concept_Metadata are:

- Simplicity of creation and maintenance
- Commonly understood semantics
- Conformance to existing and emerging standards
- Extensibility
- Interoperability among collections and indexing systems

The EO-Metadata is composed of a set of schemes responsible for representing different concepts and relationship dimensions as well as their annotation. As foundation metadata these first level schemes are not expected to change often. However the schemes' subclasses are very easy to extend, change and conform.

Some of the schemes are also more stable than others, for example an ISO standard is more stable than evidences adopted to measure a relationship such as gene expressions, literature support or any other calculated measure adopted to support evidence.

The next paragraphs describe the first level schemes adopted in the current metadata version. Figure 3-2 presents a more detailed version of the metadata. Appendix B also includes some screenshots of the metadata as reference.

The following schemes are used to represent knowledge:

Table 3-1 – Evolving Ontologies Schemes.

Schemes

Concept Name:	EO-Annotation_Scheme
Label:	Annotation
Documentation:	Defines external references to an instance of a concept.
Comments:	It might include publications, images, datasets related to the instance.
Direct Subclasses:	EO-External_Database, EO-GO_Future_Annotation, PubMed
Concept Name:	EO-Creator_Scheme
Label:	Creator
Documentation:	Defines the creator of any concept in the ontology.
Comments:	It might include persons, agents, algorithms, and systems responsible for the creation of a concept or relationship in the ontology. Typically, the name of a creator should be used to indicate the entity.

Direct Subclasses:	Person (ontology engineer, user), System, and Service
Concept Name:	EO-Evidence
Label:	Evidence
Documentation:	Defines measures which show the evidence of a relationship.
Comments:	It may consider different measures such as number of publications, reliability of the source, statistical measures, gene expression ,etc.
Direct Subclasses:	
Concept Name:	EO-Identification_Scheme
Label:	Identification
Documentation:	An unambiguous reference to the resource within a given context.
Comments:	Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system.
Direct Subclasses:	Unique_Identifier, Name
Concept Name:	EO-Language_Scheme
Label:	Language
Documentation:	Language of the content of the concept.
Comments:	Recommended best practice is to use RFC 3066 [RFC3066], which, in conjunction with ISO 639 [ISO639], defines two- and three-letter primary language tags with optional sub tags. Examples include ‘en’ or ‘eng’ for English.
Direct Subclasses:	

Concept Name:	EO-Management_Scheme
Label:	Management Rights
Documentation:	Entity responsible for store management rights upon concepts.
Comments:	May be used to control access to specific knowledge. It is well used by reporting systems, intelligence based knowledge bases, etc.
Direct Subclasses:	Access_rights, Change_rights
Concept Name:	EO-Relationship_Scheme
Label:	Relationships
Documentation:	Defines any type of relationship found by a user, system, algorithm, etc.
Comments:	It represents known relationships, such as *part_of* and *responsible_for*, and also permits the creation of new relationships acquired from experts or through data mining activities.
Direct Subclasses:	Association_Rules, Reified Relationships
Concept Name:	EO-Source_Scheme
Label:	Sources
Documentation:	Defines the source of information.
Comments:	It can be used to annotate a concept or to integrate different sources of information. It might be an information source, a domain expert, a publication etc. It allows for the acquisition of information from different sources, such as UMLS, internet sites, or clinical data, and maintains its independence of the original source.

Direct Subclasses:	UMLS_Semantic_Network, UMLS_Methasaurus, EO-Domain_Expert, EO-External_Database and EO-Experiments
Concept Name:	EO-Spatial_Scheme
Label:	Spatial
Documentation:	Spatial characteristics of the concept (a place name or geographic coordinates).
Comments:	It might be country code, city code, airport codes, mesh block, etc.
Direct Subclasses:	ISO3166
Concept Name:	EO-Subject_Scheme
Label:	Subject
Documentation:	The topic or keywords which describe a subject.
Comments:	Typically, a Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.
Direct Subclasses:	DDC - Dewey Decimal Classification
Concept Name:	EO-Temporal_Scheme
Label:	Temporal
Documentation:	Temporal characteristics of the concept (a period label, date, or date range).
Comments:	
Direct Subclasses:	ISO 8601 [W3CDTF], EO-Date_Scheme, etc
Concept Name:	EO-Type-Scheme
Label:	Type

Documentation:	The nature or genre of the content of the resource, such as image or text.
Comments:	The user can define a new type of information and constraints related to the insertion of instances.
Direct Subclasses:	Data_set, Images, Software,Sound, Stillimage

Although the schemes utilized cover most requirements to correctly annotate concepts and relationships, the meta-data can easily be extended to accommodate any further annotation required.

The EO-Concept_Metadata developed in this research extends the widely adopted Dublin Core meta-data developed by the Dublin Core Metadata Initiative (DCMI, 2007) and the Resource Description Framework (RDF) recommendations for the description and annotation of concepts (Hayes, 2004). It does so by following a similar approach to that adopted in prior research (Carroll, Bizer, Hayes, & Stickler, 2005; Catton & Shotton, 2004; Kauppinen & Hyvönen, 2006) in the sense of defining concepts through the use of other concepts and relations (meta-knowledge). In keeping with the expectations of this work, this novel approach provides for flexible knowledge representation and formalism independence which, in consequence, allows for the adoption of different representation formalisms and easy migration between them.

Unlike previous research (Carroll, Bizer, Hayes, & Stickler, 2005; Catton & Shotton, 2004), the approach adopted here does not use named graphs to give semantics to RDF or any other representation language. Instead, it has language independent meta-knowledge which enables the expression of, for example, propositional attitudes such as belief or evidence, at the same time that it expresses uncertainty-simulating named graphs.

Figure 3-2 shows a graphical representation of the current concept_metadata. A number of classes, subclasses and relationships modelled in the evolving meta-knowledge are represented. The hierarchical knowledge as well as multiple inheritance is also shown through relationships such as IS-A and Has-A.

EO-Concept_Metadata

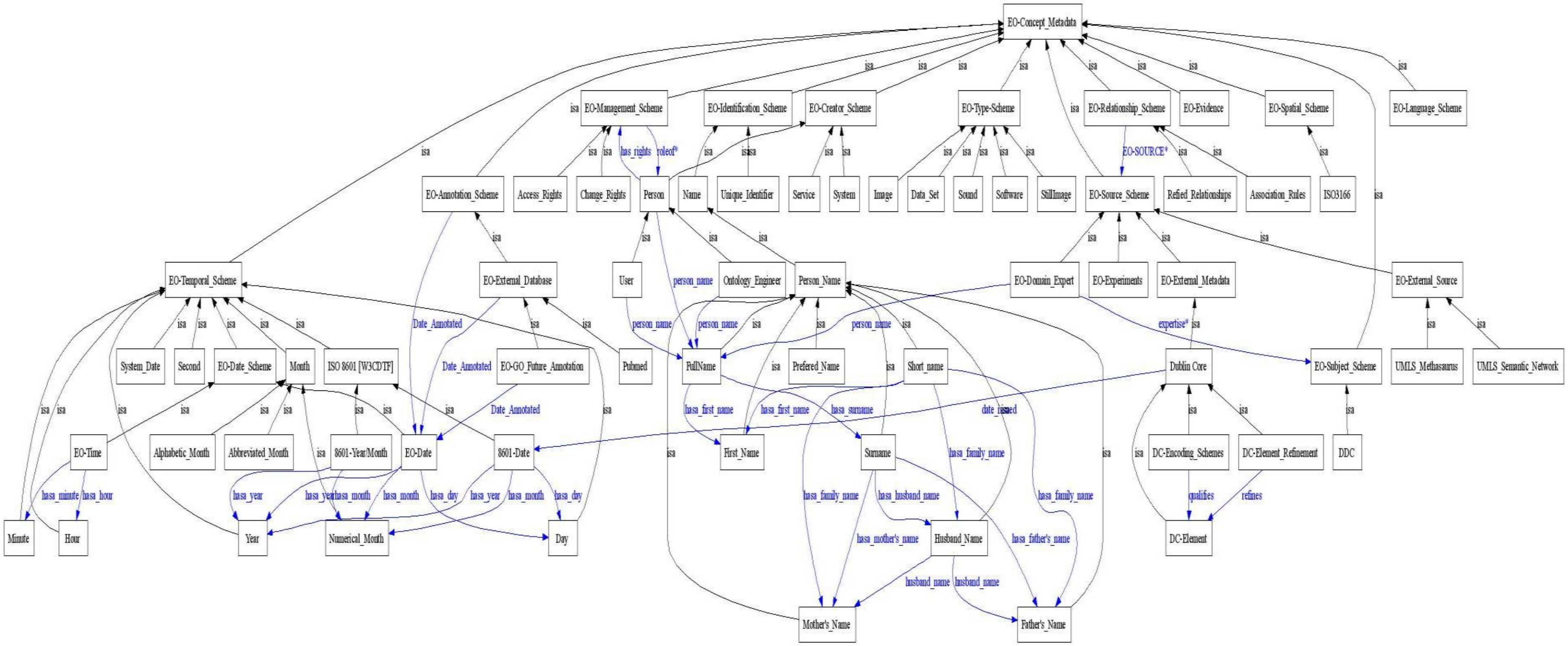


Figure 3-2– Evolving Ontology Metadata model

3.3.2. Evolving_Ontology_Meta-Class

The Evolving Ontology meta-knowledge defines a meta-class that specifies a minimal set of attributes to enable concept/relationship traceability. Traceability is related to the need to acquire and incorporate information from different sources while keeping track of the source of the information, creation or any other change in an ontological representation. Every concept, and in consequence any scheme, in the evolving ontology meta-knowledge is annotated with the meta-class which is itself annotated with the EO-metadata.

This recursive mechanism allows the integration of different sources of information as well as creating an annotation tool that maintains a record of the creation and date of any concept in an ontological representation. It allows among other things the tracking of source information when reusing ontologies instances in different scenarios. For example, in the context of intelligence, it allows the use of external sources as evidence to build intelligence products while keep track of creation and source so as to be compliant in regard to privacy and security level requirements. In the biomedical domain, it may highlight the quality of a prior analysis of data by considering the level of trust of a source, for example, Swiss-Prot data is manually annotated by experts, and therefore is very well accepted in the bioinformatics domain, while at the same time some automatically generated annotations are not considered as strong from a biological evidence point of view.

Each EO-term is specified with the following minimal set of attributes:

Table 3-2. – Evolving Ontology Meta-Class

Name:	The unique token (ID) assigned to the concept.
Internal Name	Unique identifier of a concept.
Source:	Reference to an information source from which the present concept is acquired, originated or automatically learnt.
Creator:	An entity primarily responsible for creating the concept.
Date Created:	A date associated with the creation of the concept.
Documentation:	A statement that represents the concept and essential nature of the term.
Constraints:	A system property able to represent the conceptual constraints of a concept.

Although the *Internal Name* attribute is not explicitly represented in the Evolving Ontology Meta-class, as shown in Figure 3-3, it uses Protégé Frame's internal name (Noy, Fergerson, & Musen, 2000) to uniquely identify any term within the ontological representation.

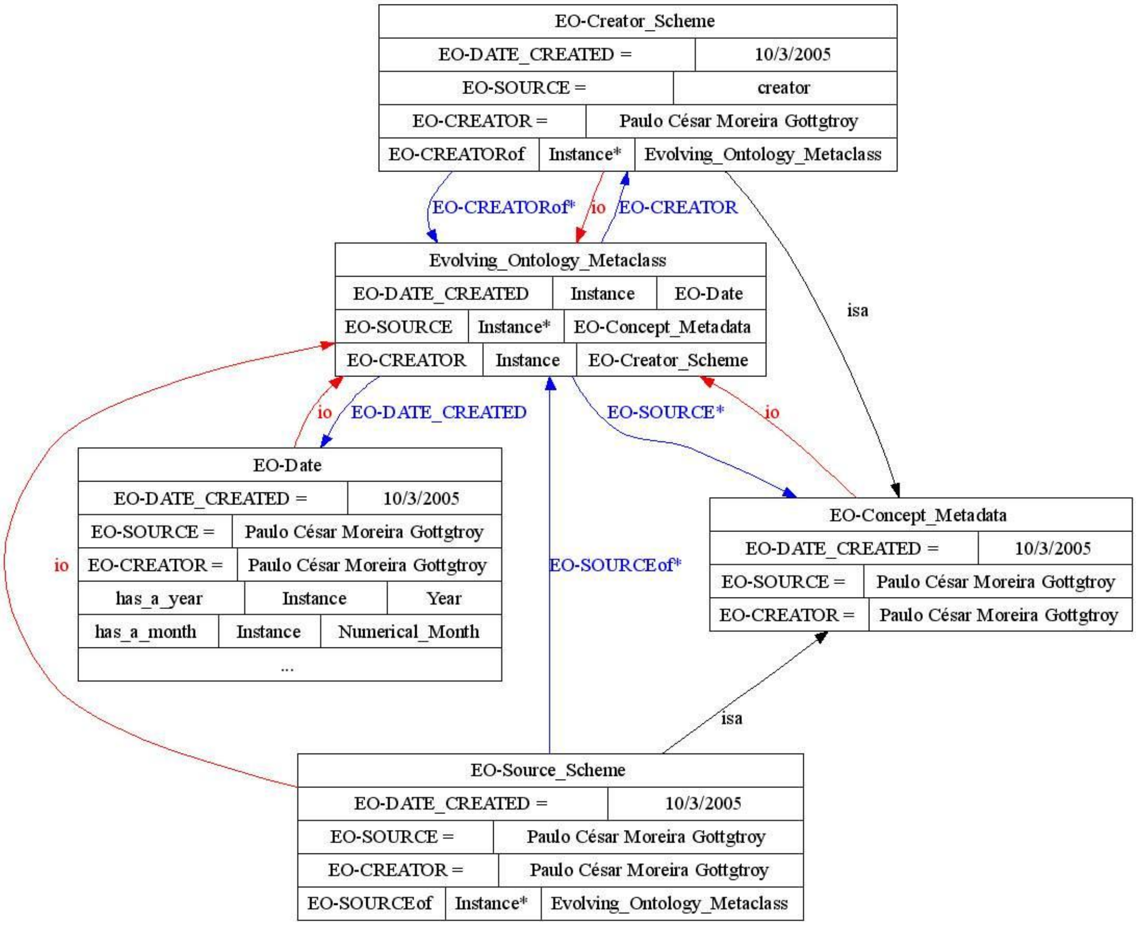
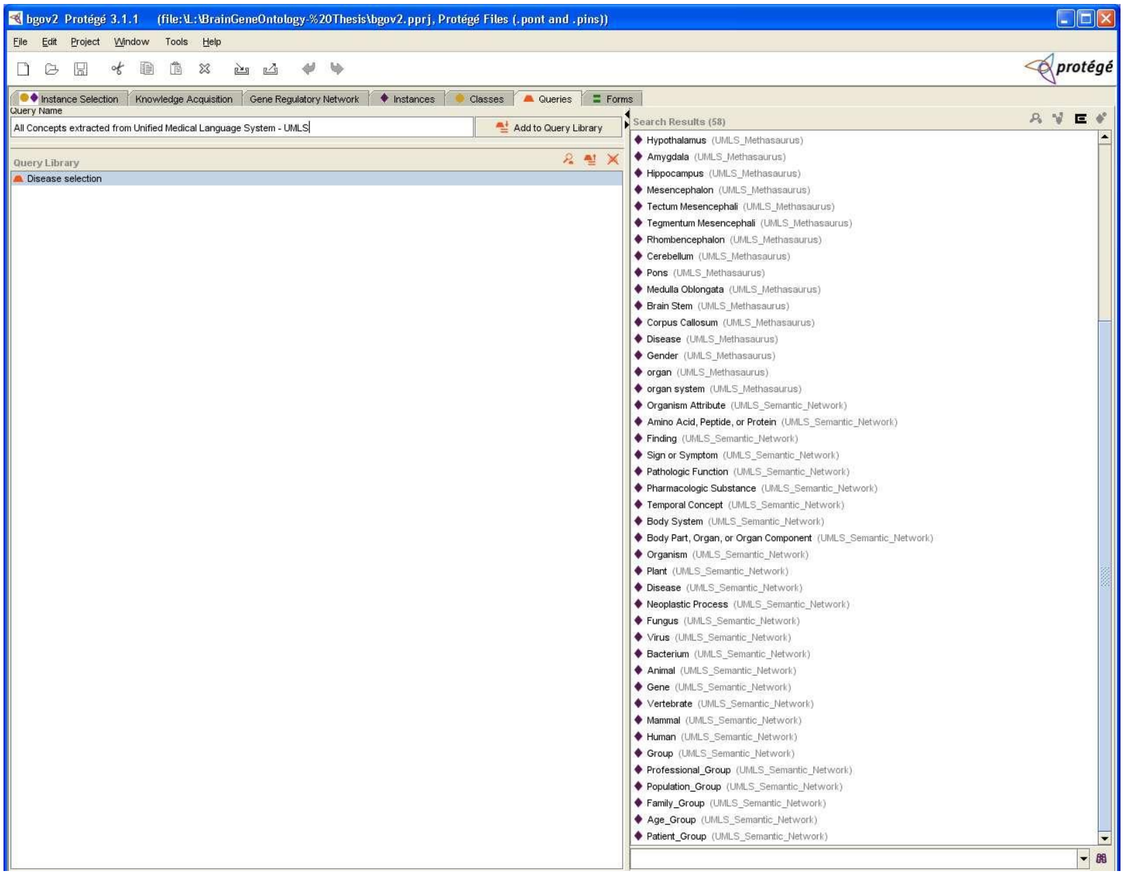

EO-Creator_Scheme
EO-DATE_CREATED = | 10/3/2005
EO-SOURCE = | creator
EO-CREATOR = | Paulo César Moreira Gottgtroy
EO-CREATORof | Instance* | Evolving_Ontology_Metaclass
EO-CREATORof*
io
EO-CREATOR
isa
Evolving_Ontology_Metaclass
EO-DATE_CREATED | Instance | EO-Date
EO-SOURCE | Instance* | EO-Concept_Metadata
EO-CREATOR | Instance | EO-Creator_Scheme
io
EO-DATE_CREATED
EO-SOURCE*
io
EO-Date
EO-DATE_CREATED = | 10/3/2005
EO-SOURCE = | Paulo César Moreira Gottgtroy
EO-CREATOR = | Paulo César Moreira Gottgtroy
has_a_year | Instance | Year
has_a_month | Instance | Numerical_Month
...
io
EO-SOURCEof*
EO-Concept_Metadata
EO-DATE_CREATED = | 10/3/2005
EO-SOURCE = | Paulo César Moreira Gottgtroy
EO-CREATOR = | Paulo César Moreira Gottgtroy
isa
EO-Source_Scheme
EO-DATE_CREATED = | 10/3/2005
EO-SOURCE = | Paulo César Moreira Gottgtroy
EO-CREATOR = | Paulo César Moreira Gottgtroy
EO-SOURCEof | Instance* | Evolving_Ontology_Metaclass

Figure 3-3 – Evolving Ontology Meta-class graphical representation.

Further attributes (properties) may also be necessary to uniquely define concepts in representation language formalisms other than Frames. The following additional properties may, for example, be used:

Table 3-3 – Additional Evolving Ontology Meta-class attributes

Unique Identifier:	The Uniform Resource Identifier used to uniquely identify a term.
Label:	The human-readable label assigned to the term.



Query Name
All Concepts extracted from Unified Medical Language System - UMLS

Query Library
- Disease selection

Search Results (58)
- Hypothalamus (UMLS_Methasaurus)
- Amygdala (UMLS_Methasaurus)
- Hippocampus (UMLS_Methasaurus)
- Mesencephalon (UMLS_Methasaurus)
- Tectum Mesencephali (UMLS_Methasaurus)
- Tegmentum Mesencephali (UMLS_Methasaurus)
- Rhombencephalon (UMLS_Methasaurus)
- Cerebellum (UMLS_Methasaurus)
- Pons (UMLS_Methasaurus)
- Medulla Oblongata (UMLS_Methasaurus)
- Brain Stem (UMLS_Methasaurus)
- Corpus Callosum (UMLS_Methasaurus)
- Disease (UMLS_Methasaurus)
- Gender (UMLS_Methasaurus)
- organ (UMLS_Methasaurus)
- organ system (UMLS_Methasaurus)
- Organism Attribute (UMLS_Semantic_Network)
- Amino Acid, Peptide, or Protein (UMLS_Semantic_Network)
- Finding (UMLS_Semantic_Network)
- Sign or Symptom (UMLS_Semantic_Network)
- Pathologic Function (UMLS_Semantic_Network)
- Pharmacologic Substance (UMLS_Semantic_Network)
- Temporal Concept (UMLS_Semantic_Network)
- Body System (UMLS_Semantic_Network)
- Body Part, Organ, or Organ Component (UMLS_Semantic_Network)
- Organism (UMLS_Semantic_Network)
- Plant (UMLS_Semantic_Network)
- Disease (UMLS_Semantic_Network)
- Neoplastic Process (UMLS_Semantic_Network)
- Fungus (UMLS_Semantic_Network)
- Virus (UMLS_Semantic_Network)
- Bacterium (UMLS_Semantic_Network)
- Animal (UMLS_Semantic_Network)
- Gene (UMLS_Semantic_Network)
- Vertebrate (UMLS_Semantic_Network)
- Mammal (UMLS_Semantic_Network)
- Human (UMLS_Semantic_Network)
- Group (UMLS_Semantic_Network)
- Professional_Group (UMLS_Semantic_Network)
- Population_Group (UMLS_Semantic_Network)
- Family_Group (UMLS_Semantic_Network)
- Age_Group (UMLS_Semantic_Network)
- Patient_Group (UMLS_Semantic_Network)

Figure 3-4 – Tracking provenance of source by a query language.

The provenance of source, of date and of creation can be programmatically accessed through an application program interface as well as by a query language which allows the construction of complex queries of the knowledge base – Figure 3-4. In the figure above (left top panel – query panel) a query is selected to search for all knowledge extracted from UMLS (Unified Medical Language System) and returns (right side panel – search results) 58 concepts current available in the ontology which were extracted from different parts of the UMLS.

These features enable the management of the meta-data in different scenarios such as in the evaluation of the quality of knowledge based on its source, and they can be used to trigger further actions when used in conjunction with rules for ontology maintenance, alerts, risk analysis and so forth.

3.3.3. Model Integration

Model integration is related to the acquisition and maintenance of external ontological models in the process of developing an ontology. There are two different mechanisms to acquire, incorporate and reuse ontologies in the EO model: importation and inclusion. The former refers to the act of permanently acquiring external knowledge. The latter embeds the source ontology in the target ontology but retains independence in both. The selection of the most suitable method is based on the updating requirement of a problem and/or domain.

In the form of meta-knowledge, ontologies are often quite stable. In well defined domains, such as the anatomy domain, stable ontological representation is acceptable, for example, the Foundational Model of Anatomy (FMA) (Rosse & Mejino, 2003). In these cases, the use of both *Source* attribute and the permanent incorporation of knowledge by the importation method are sufficient to allow a flexible representation, which might be updated long term, without impacting on the quality of the knowledge currently represented.

On the other hand, there exist specific situations where regular knowledge source updating is mandatory. This is particularly true in domains where knowledge is highly complex or consensus is not yet achieved, such as the bioinformatics domain. The inclusion method is then appropriated to embed external ontologies into a target ontology model.

The inclusion method enables the use of the knowledge acquired from an external source but doesn't allow any change in its structure. The included ontology is maintained as a separate ontology and any change in its structure is reflected in the final ontology when it is reloaded. The implementation of the inclusion method brings a huge degree of flexibility to the model, since the frequency of updating can vary according to the ontology engineer's discrimination.

This research uses the importation method along with the evolving ontology source schema to acquire concepts from stable knowledge sources such as the Unified Medical Language System (UMLS) terminological source. The inclusion method enables the acquisition of 'unstable' knowledge such as gene knowledge from unknown or new areas, such as brain-gene disease and nutrigenomics, where constant checking for new genes, molecular function, and so on plays a major role in the knowledge discovery process.

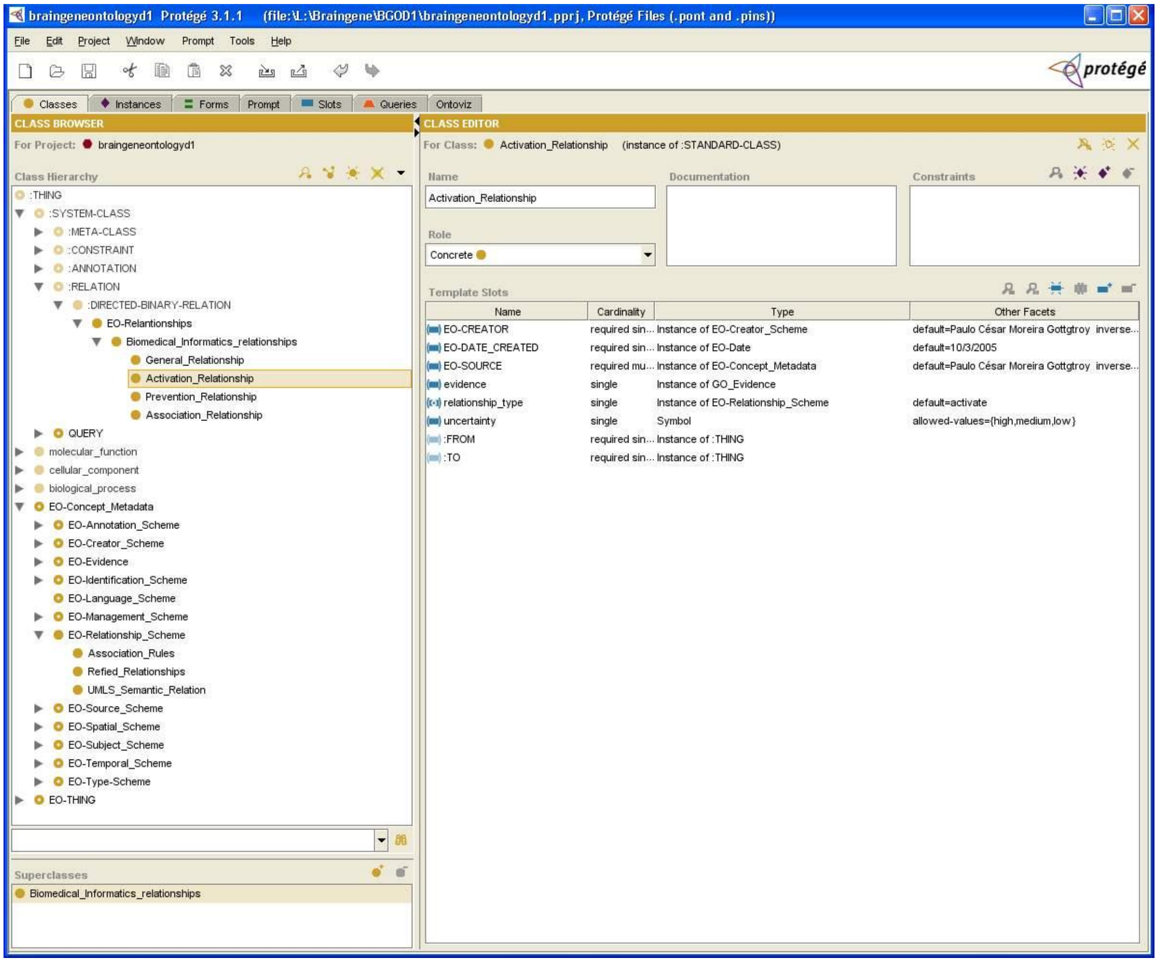
3.3.4. Knowledge Map

Although both Evolving Ontology Metadata and Meta-classes are able to record provenance of source, creation and date, and the Model Integration mechanisms enable the acquisition and maintenance of knowledge by means of external ontologies, there still exist other central questions in relation to ontology change in the context of knowledge discovery in databases: when should a discovery be considered *"strong enough"* to change an ontological representation? What evidence supports the change? Is the new knowledge valid? To mention a few examples.

Differing from the manual acquisition of knowledge during the ontology building process, the semi-automatic and automatic acquisition of knowledge by learning algorithms requires a temporary ontological structure to store the knowledge gained by the data mining algorithm before its validation. This data-driven change (Sure, 2004) adds the challenge of knowledge validation, beliefs and the treatment of uncertainty.

There are some approaches that focus on knowledge representation formalism to deal with concept and relationship validation, such as those that deal with the dynamic and uncertain nature of domains through the use of non-classical logic (Zadeh, 2006), the integration of fuzzy concepts in description logic (Straccia, 2006), and fuzzy formal concept analysis (Quan, Hui, & Cao, 2004). Contrasting with these approaches, the evolving ontology approach focuses on a generic solution based on the meta-knowledge representation composed of knowledge maps and reified relations to cope with knowledge validation along with expert review of the data and rule based approaches.

The knowledge map stores the knowledge discovered by an ontology acquisition tool that is then used by the knowledge engineer or experts, or by inference engines, to validate, annotate, and generate new hypotheses according to the criteria defined in the domain problem. Each map stores the concepts related to a problem and the relationships discovered by the learning technique as reified relations (as depicted in Figure 3-5).


braingeneontologyd1 Protégé 3.1.1 (file:\L:\Braingene\BGOD1\braingeneontologyd1.pprj, Protégé Files (.pont and .pins))
File
Edit
Project
Window
Prompt
Tools
Help
protégé
Classes
Instances
Forms
Prompt
Slots
Queries
Ontoviz
CLASS BROWSER
CLASS EDITOR

For Project: braingeneontologyd1

Class Hierarchy

```
:THING
▼ :SYSTEM-CLASS
    ▶ :META-CLASS
    ▶ :CONSTRAINT
    ▶ :ANNOTATION
    ▼ :RELATION
        ▼ :DIRECTED-BINARY-RELATION
            ▼ EO-Relantionships
                ▼ Biomedical_Informatics_relationships
                    General_Relationship
                    Activation_Relationship
                    Prevention_Relationship
                    Association_Relationship
    ▶ QUERY
▶ molecular_function
▶ cellular_component
▶ biological_process
▼ EO-Concept_Metadata
    ▶ EO-Annotation_Scheme
    ▶ EO-Creator_Scheme
    ▶ EO-Evidence
    ▶ EO-Identification_Scheme
      EO-Language_Scheme
    ▶ EO-Management_Scheme
    ▼ EO-Relationship_Scheme
        Association_Rules
        Refied_Relationships
        UMLS_Semantic_Relation
    ▶ EO-Source_Scheme
    ▶ EO-Spatial_Scheme
    ▶ EO-Subject_Scheme
    ▶ EO-Temporal_Scheme
    ▶ EO-Type-Scheme
▶ EO-THING
```

Superclasses

Biomedical_Informatics_relationships

For Class: Activation_Relationship (instance of :STANDARD-CLASS)

Name: Activation_Relationship

Documentation

Constraints

Role: Concrete

Template Slots

Name	Cardinality	Type	Other Facets
EO-CREATOR	required sin...	Instance of EO-Creator_Scheme	default=Paulo César Moreira Gottgtroy inverse...
EO-DATE_CREATED	required sin...	Instance of EO-Date	default=10/3/2005
EO-SOURCE	required mu...	Instance of EO-Concept_Metadata	default=Paulo César Moreira Gottgtroy inverse...
evidence	single	Instance of GO_Evidence	
relationship_type	single	Instance of EO-Relationship_Scheme	default=activate
uncertainty	single	Symbol	allowed-values={high,medium,low}
:FROM	required sin...	Instance of :THING	
:TO	required sin...	Instance of :THING	

Figure 3-5 – A Protégé representation of a biomedical reified relation.

The left panel shows an *activation relationship* selected from a set of biomedical-informatics relationship classes. The right panel shows (in blue) the concepts captured in that relationship, such as evidence and source. For example, the slots could have as values either a number value for the evidence and the algorithm which was used to generate that value as source, or an expression value and the paper reference for the source which was used as a basis for the annotation of the expression value.

The reified relations are represented as Evolving_Ontology_Relationships that have a minimum of six fixed set properties and other properties dependent on the problem domain, data mining algorithm, and the ontology engineer specification. Figure 3-6 exemplifies a set of relationships used in the biomedical ontology case study. The table below represents the fixed attributes and extra attributes in yellow boxes:

Table 3-4 – Evolving Ontology Relationship Attributes

From:	Concept source of the relationship.
To	Concept target of the relationship.
EO-Source:	A reference to an information source from which the present relationship is acquired, originated or automatically learnt.
EO-Creator:	The relationship creator, such as a simulation, a machine learning algorithm, an expert, etc.
EO –Date_Created	A date associated with the creation of the relationship.
Relationship_type:	The relationship type based on the evolving ontology relationship schema.
Evidence:[2]	Evidence that supports the concept of that relationship such as

	number of publications found in the literature.
Gene_Expression:[4]	The number representing the expression of a gene in a Gene Regulatory Network.

Extra attributes may also include properties that validate or qualify knowledge, such as experimental evidence, triangulation evidence, and uncertainty. Attributes can also fire rules built in accordance with some criteria established by the ontology engineer or based on the quality of annotations, quantity of annotations and so on.

The four components of the Evolving Ontology Meta-knowledge model presented in this section (Figure 3-1) – meta-data, meta-class, model integration and knowledge maps – enable the in-principle construction of flexible and dynamic knowledge repositories able to evolve and keep track of change while giving a 360 degree view of a problem domain for decision makers in a knowledge discovery process.

[4] The yellow attributes represented in the table are often utilized in bioinformatics case studies, as is the case in this thesis as described later in this chapter and in several examples in the following chapters.

Reified Relations

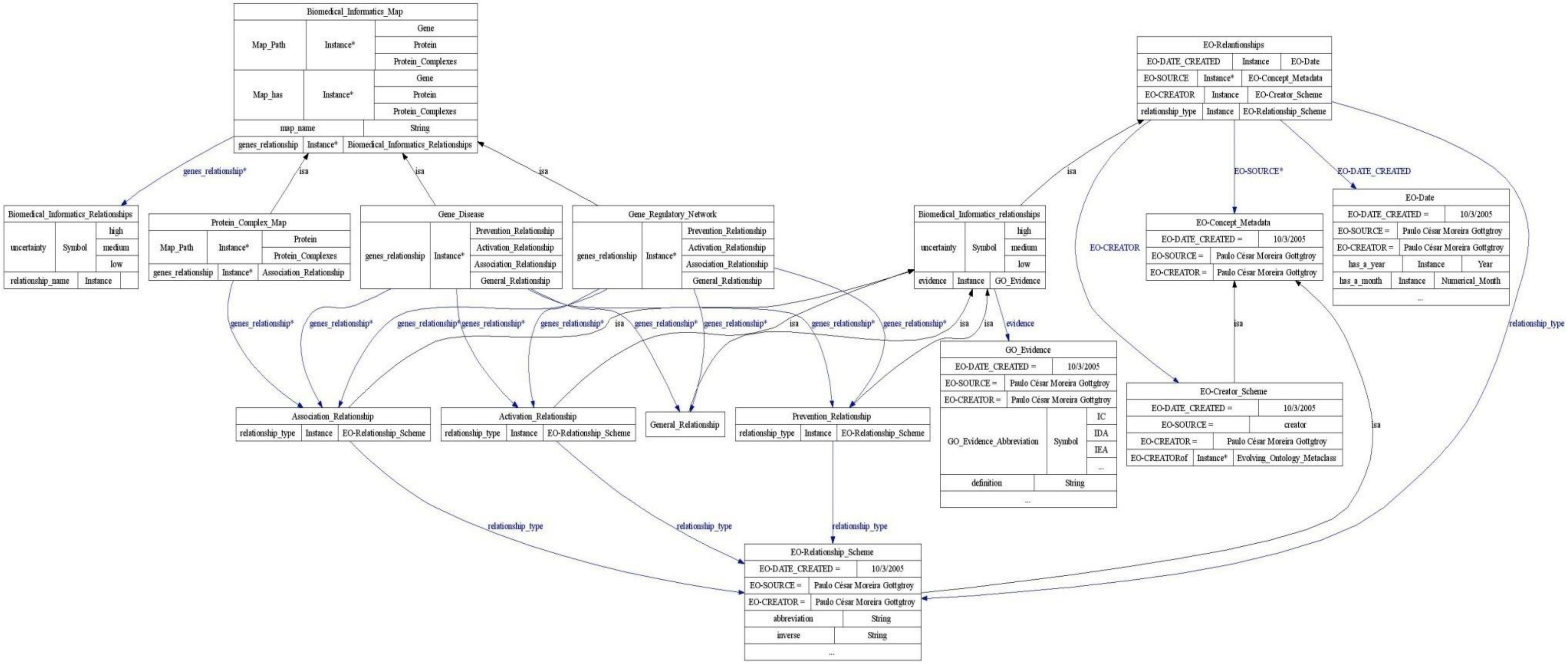


Figure 3-6 – An example of the evolving biomedical ontology reified relations model.

The next section briefly describes a case study developed in the course of this research that applied the main concepts of evolving ontology meta-knowledge and defined the requirements for the extension of the model in order to cope with future experiments developed in this research.

3.4. Biomedical Ontology case study on Leukaemia Cancer Data

The case study focused on the development of a novel multi-dimensional biomedical ontology linking genes to related diseases. The ontology utilises two of the most used biomedical knowledge sources, the Gene Ontology (GO) (Ashburner & Ball, 2000) and the Unified Medical Language System (UMLS) (NCI, 2003), as well as knowledge acquired manually from experiments undertaken across aligned research and published in the literature.

This case study, which addressed Leukaemia specifically, exhibited many of the requirements for an ontology driven knowledge discovery methodology when considering manual annotation at early stages of the thesis development. Therefore it was used in parallel with the iterative process of development, testing and evaluation of tools to support evolving ontology meta-knowledge models as defined in the research methodology (Chapter 1). Figure 3-7 shows the relationship between the conceptual models and ontologies used in the development and testing of the Leukaemia case study.

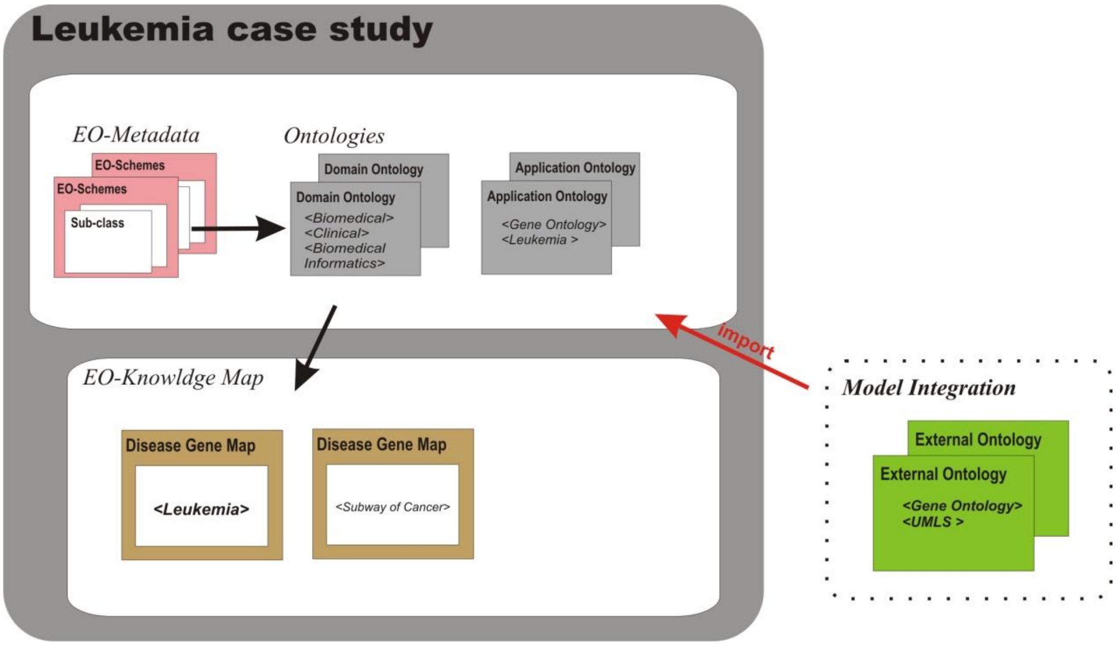
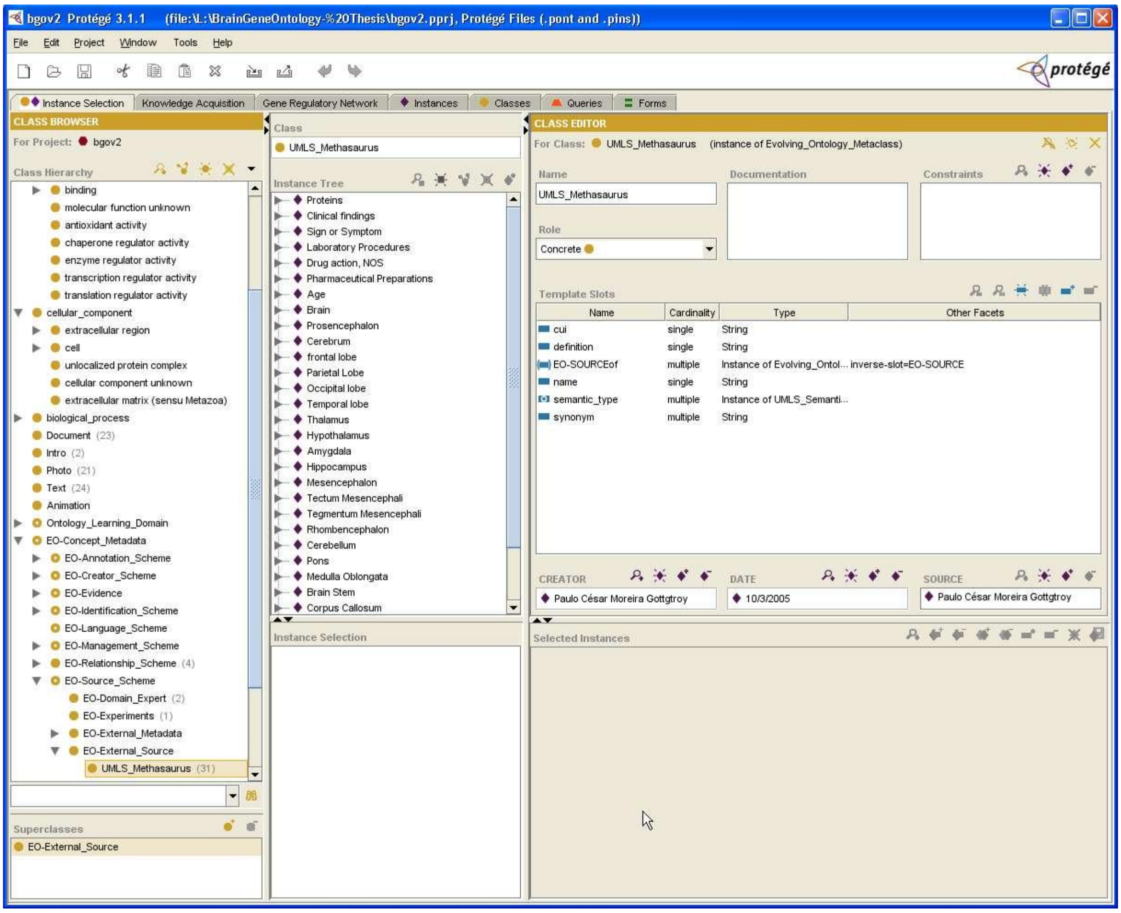


Figure 3-7 – Diagram of the Leukaemia biomedical ontology case study.

The approach has helped the thesis to:

- Establish the requirements for provenance of source, creation and date;
- Build a knowledge acquisition tool for manual construction of gene regulatory networks;
- Ensure support for evolving ontology model enhancement.



For Project: bgov2

- binding
- molecular function unknown
- antioxidant activity
- chaperone regulator activity
- enzyme regulator activity
- transcription regulator activity
- translation regulator activity
- cellular_component
  - extracellular region
  - cell
  - unlocalized protein complex
  - cellular component unknown
  - extracellular matrix (sensu Metazoa)
- biological_process
- Document (23)
- Intro (2)
- Photo (21)
- Text (24)
- Animation
- Ontology_Learning_Domain
- EO-Concept_Metadata
  - EO-Annotation_Scheme
  - EO-Creator_Scheme
  - EO-Evidence
  - EO-Identification_Scheme
  - EO-Language_Scheme
  - EO-Management_Scheme
  - EO-Relationship_Scheme (4)
  - EO-Source_Scheme
    - EO-Domain_Expert (2)
    - EO-Experiments (1)
    - EO-External_Metadata
    - EO-External_Source
      - UMLS_Methasaurus (31)

Superclasses: EO-External_Source

Class: UMLS_Methasaurus

- Proteins
- Clinical findings
- Sign or Symptom
- Laboratory Procedures
- Drug action, NOS
- Pharmaceutical Preparations
- Age
- Brain
- Prosencephalon
- Cerebrum
- frontal lobe
- Parietal Lobe
- Occipital lobe
- Temporal lobe
- Thalamus
- Hypothalamus
- Amygdala
- Hippocampus
- Mesencephalon
- Tectum Mesencephali
- Tegmentum Mesencephali
- Rhombencephalon
- Cerebellum
- Pons
- Medulla Oblongata
- Brain Stem
- Corpus Callosum

For Class: UMLS_Methasaurus (instance of Evolving_Ontology_Metaclass)

Name: UMLS_Methasaurus

Role: Concrete

Name	Cardinality	Type	Other Facets
cui	single	String	
definition	single	String	
EO-SOURCEof	multiple	Instance of Evolving_Ontol...	inverse-slot=EO-SOURCE
name	single	String	
semantic_type	multiple	Instance of UMLS_Semanti...	
synonym	multiple	String	

CREATOR: Paulo César Moreira Gottgtroy

DATE: 10/3/2005

SOURCE: Paulo César Moreira Gottgtroy

Figure 3-8 - GO and UMLS ontological representations in Protégé.

The ontological model was constructed iteratively based on the Evolving Ontology Meta-knowledge model to represent genes related to cancer and more specifically to leukaemia. Genetic knowledge was mainly acquired from GO and medical knowledge from UMLS with complementary knowledge acquired from literature – Figure 3-8. The left panel shows a number of concepts form the ontology and focus on the concepts acquired from UMLS methasaurus. The middle panel show a list with some concepts while the right panel shows information about the slots captured with UMLS data.

The ontology was developed in the Protégé ontology environment (Protege, 2006a) using Frames for knowledge representation. A manual knowledge acquisition tool was developed using extended Protégé plug-ins. Although initially developed as a manual acquisition tool, all reified relations requirements for automatic and semi-automatic acquisition were considered when developing the meta-knowledge model supported by the tool. (This approach has facilitated the extension of the model and tool to cope with data mining techniques developed later in the thesis – Chapters 6 and 7.)

The importation technique was used in the case study as it was compatible with the technologies and Protégé plug-ins available at the time of the project. This did not impact on the study as its goal was the evaluation of the evolving knowledge model rather than the software support for the framework (described in Chapters 6 and 7).

The domain ontology is composed of five ontologies, each representing a specific sub-domain in the biomedical informatics area as well as the initial version of the evolving ontology meta-knowledge.

Biomedical Domain[5]

This entity represents general biomedical knowledge. It includes abstract concepts, such as organism, and more concrete concepts, such as disease and its instances. Most of the knowledge acquired in this entity is referenced to UMLS.

Biomedical Informatics Domain

The biomedical informatics domain represents common knowledge shared by both the biomedical and bioinformatics domains. Each subclass of this entity, such as oncogene, inherits characteristics from the domain and properties related with the biomedical domain.

Clinical Domain

Clinical domain classes represent clinical knowledge contained in laboratories' results regarding treatment, drugs and so on.

The subclasses are mainly multi-inherited from the biomedical domain and its instances are directly updated from databases.

[5] All biomedical concepts are either based on the UMLS semantic network or based on knowledge acquired from domain experts and literature.

Gene Ontology

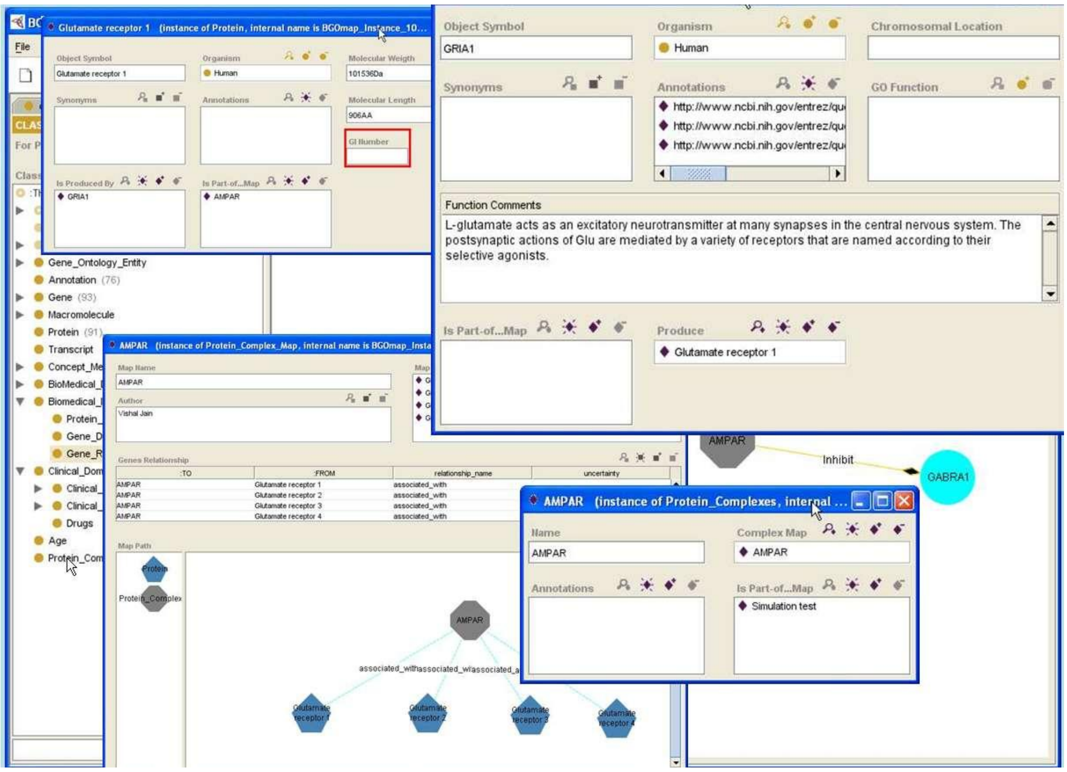
Gene ontology (GO) is adopted as the main source of bioinformatics knowledge. This entity is directly imported to the main ontology and its instances are included through software tools responsible for translating the GO annotations into the Protégé knowledge base.

All genes annotated in the leukaemia case study are referenced to the respective GO annotation and the published literature extracted from sources such as PubMed (NBCI, 2006).

Disease Gene Map

This ontology is the core of the leukaemia-specific case study. It enables the acquisition of gene/disease knowledge and links it to additional disease and gene knowledge such as molecular weights, and chromosome positions, through a graphical knowledge acquisition tool – Figure 3-9. Each instance of this ontology represents a disease gene map, which is traceable through a query language that makes it possible, for example, to answer questions such as; 'which genes are related to the occurrence of leukaemia?'

Figure 3-9 shows a series of forms created to acquire knowledge linking the gene relationship data. The expert starts with the creation of the disease map and then manually inserts new knowledge while building the ontology.



Glutamate receptor 1 (instance of Protein, internal name is BGOmap_Instance_10...

Object Symbol: Glutamate receptor 1

Organism: Human

Molecular Weigth: 101536Da

Synonyms:

Annotations:

Molecular Length: 906AA

GI Number:

Is Produced By: GRIA1

Is Part-of...Map: AMPAR

Object Symbol: GRIA1

Organism: Human

Chromosomal Location:

Synonyms:

Annotations:
- http://www.ncbi.nih.gov/entrez/qu
- http://www.ncbi.nih.gov/entrez/qu
- http://www.ncbi.nih.gov/entrez/qu

GO Function:

Function Comments: L-glutamate acts as an excitatory neurotransmitter at many synapses in the central nervous system. The postsynaptic actions of Glu are mediated by a variety of receptors that are named according to their selective agonists.

Is Part-of...Map:

Produce: Glutamate receptor 1

AMPAR (instance of Protein_Complex_Map, internal name is BGOmap_Insta

Map Name: AMPAR

Author: Vishal Jain

Genes Relationship

:TO	:FROM	relationship_name	uncertainty
AMPAR	Glutamate receptor 1	associated_with	
AMPAR	Glutamate receptor 2	associated_with	
AMPAR	Glutamate receptor 3	associated_with	
AMPAR	Glutamate receptor 4	associated_with	

Map Path

AMPAR (instance of Protein_Complexes, internal ...

Name: AMPAR

Complex Map: AMPAR

Annotations:

Is Part-of...Map: Simulation test

Figure 3-9 – Gene disease knowledge acquisition tool.

Each map graphically represents relationships in a way that enables visualisation of existing knowledge and the creation of new relationships. Additional properties, such as uncertainty and evidence, can be inferred and added by experts or programmatically.

The maps act as knowledge acquisition tools allowing the expert to annotate new knowledge acquired from external databases and creating new instances linked to external information. More than one expert can work on the same project or share different projects, having a 360 degree view of the domain. In this sense, the system works as a knowledge management tool, storing knowledge from diverse projects and sharing the knowledge discovered in each experiment. The maps can also be shown through different visual representations.

The next section explores the leukaemia map from a biological perspective. The knowledge represented in the map was reviewed by experts in biology and leukaemia. There were two experts on biology which were responsible for validating the disease data with experts and the related literature.

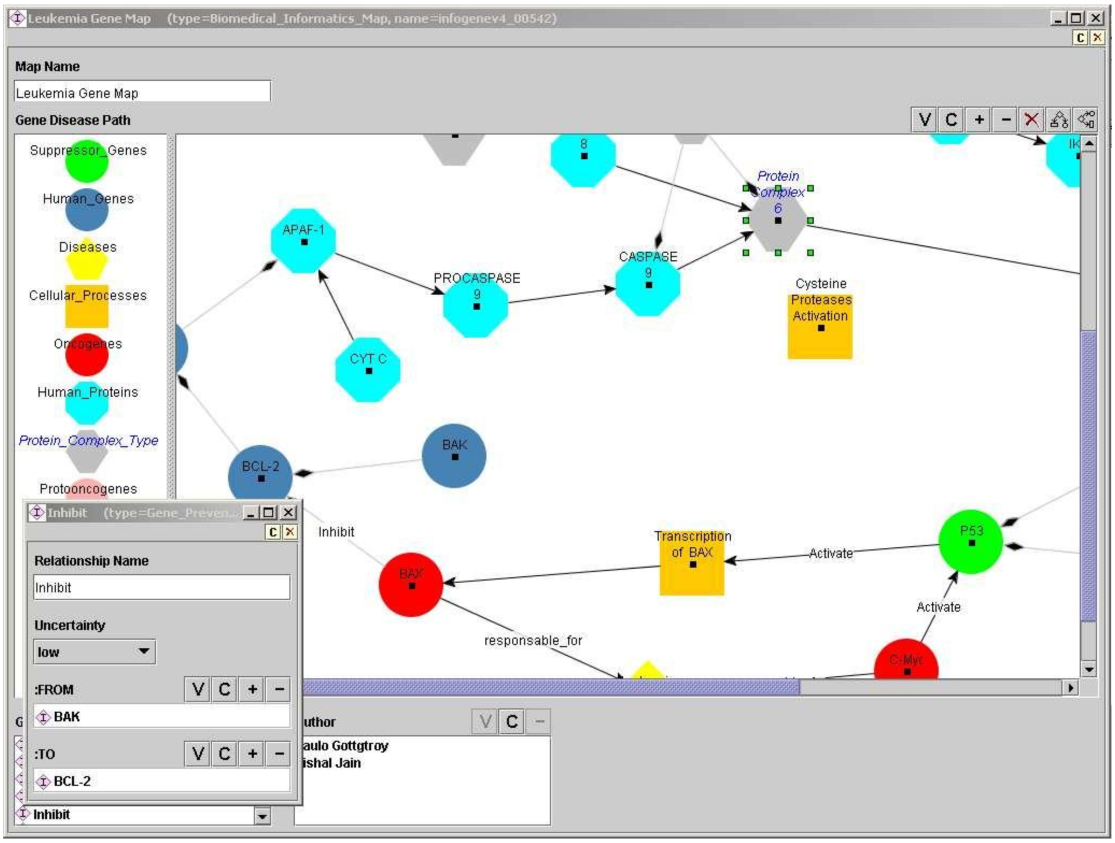
3.4.1. The Leukaemia Gene Regulatory Network Map

In medical research it has been shown that human macrophage-like cell lines spontaneously produce a suppressor factor that inhibits production of interleukin 2 (IL 2) by human blood T lymphocytes. Identifying regulatory genes and proteins from such cell lines an Infogene Map is obtained and is integrated with the apoptotic pathway.

The gene disease map, in this case, represents oncogenes, suppressor genes and genes that are related to leukaemia in a network that helps to explain the genetic pathway of the disease (see Figure 3-10). These maps are interconnected with other maps that express graphically the complex proteins and their roles in different diseases. Myc, for example, which is a powerful inducer of apoptosis, is a strategic controller of cell proliferation that acts pleiotropically to coordinate both cell growth and concomitant progression through the cell cycle. Myc is connected through the Leukaemia Infogene Map and the Subway Map of cancer pathways (Hahn &

Weinberg, 2002). It enables investigators, for instance, to answer the question ‘In which cancers is Myc involved?’

Each relationship between genes/proteins represents inhibitor or activator behaviour which is qualified by a degree of uncertainty (Figure 3-10) based on strength of evidence. In the case study reported here this uncertainty attribute was implemented as a set of rules defined using the number of publications and expert annotations acquired in the experiment. This evidence based approach followed other similar bioinformatics ontologies and databases such as Gene Ontology where domain expert input is used to score biomedical knowledge acquired form different sources.



Leukemia Gene Map (type=Biomedical_Informatics_Map, name=infogenev4_00542)

Map Name

Leukemia Gene Map

Gene Disease Path

Inhibit (type=Gene_Preven...

Relationship Name

Inhibit

Uncertainty

low

:FROM

BAK

:TO

BCL-2

Inhibit

uthor

aulo Gottgtroy
ishal Jain

Figure 3-10 – Leukaemia gene disease map.

The evidence-based approach is supported by the meta-knowledge created in the knowledge map definition (as depicted in Figure 3-5) as a combination of metadata such as the uncertainty value, eo-creator which provides sufficient information for the development of business rules around the quality of the information acquired. For instance, the leukaemia case study considered the quality of the sources of information and followed similar rules to the SwissProt project where manual annotation is used to improve the quality of knowledge acquired along with the number of publications supporting the knowledge inserted in the ontology.

The leukaemia case study was not only useful in gathering and confirming the requirements for the evolving ontology model and modelling gene regulatory networks as well as the ontology framework but also produced functional insights towards the understanding of hematopoietic cell differentiation, apoptosis and proliferation for the domain experts.

The requirements acquired in the development process of the gene disease maps, models and supportive tools were used to refine the evolving ontology model and define the software requirements for the Ontology Driven Knowledge Discovery framework described in the next chapter.

The case study was part of the iterative process of building this research. From a biological point of view the experts were able to link dispersed knowledge about Leukaemia within the ontology as well as creating a shared repository about the disease. This work was then published which formed the first set of biomedical knowledge to be included later in the brain gene-ontology. From an ontology engineering point of view, it helped us to define further requirements to enhance the conceptual model, methodologies and tools which are part of the thesis.

The most important outcome of the case study was the refinement of the conceptual model. It helped us to acquire critical knowledge to build a generic framework for ontology driven knowledge discovery.

The following section briefly introduces an further experiment utilizing the evolving ontology as part of a PhD research in the area of nutrigenomics. This research applies the evolving ontology model to the design of a biomedical ontology linking nutritional, genetic and disease knowledge.

3.5. Nutrigenomics evolving ontology case study

Nutrigenomics is a new and emerging field that studies nutrition and genomics i.e. how food affects gene expression. Diabetes is a very common disease of the modern (developed) world and Type-2 diabetes is the most common type of diabetes. There are several dimensions involved in the nutrigenomic analysis of Type-2 diabetes, such as the disease, its relationship to genes, the clinical symptoms, the nutritional data related to the disease to name a few. For example, it is believed that there are about 150 genes involved in the disease and just a few of these genes are involved at the expression level. Modelling this evolving and complex knowledge presents a substantial challenge.

This research focuses on organizing knowledge related to nutrition and Type-2 diabetes within an evolving ontology representation. It uses all the functionalities and tools of the Evolving Ontology model and ontology engineering tools.

The nutrigenomic knowledge is acquired from different sources, such as the Gene Ontology and the Unified Medical Language System (UMLS). Gene Ontology knowledge is included and updated monthly, keeping knowledge up to date. GO can also be acquired at shorter intervals depending on the project, through a plug-in that transforms GO into the Protégé format (described in Chapter 5). The ontology also includes microarray data and gene expression data for these genes. Further knowledge has been also acquired from clinical and nutritional data to create a personalized model.

The knowledge stored in the ontology should support further discovery through the integration of the knowledge base and artificial intelligence methods which will make it possible to pinpoint genes of interest and diet components of relevance in order to produce advice on healthy life-style and disease-preventing nutrition. It will integrate the Type-2 Diabetes Ontology with the Evolving Connectionist System (ECOS) developed at the Knowledge Engineering and Discover Research Institute (KEDRI) in order to develop a diagnostic system (Verma, Gottgtroy, Havukkala, & Kasabov, 2006).

3.6. Summary

Evolving processes are difficult to model because some of their parameters may not be known *a priori* or may not be available at the time a modelling exercise begins. In biological systems, for example, everything is interconnected, and ostensibly unrelated fields are related — the separation of biology into different disciplines is in some respects artificial and dangerous, or at least limiting, to the knowledge discovery process.

Conceptual research can encompass fields without limitation. So what is needed is a way to create and manage the context of the search, so that concepts having different meanings in different contexts can be represented appropriately. Scientists also need mechanisms to cross disciplines and search in areas outside their expertise, so that they can extract information critical to new discoveries.

Decision makers need tools that can intelligently assist them in the acquisition and representation of knowledge in a decision making process. Thus, the modelling of such evolving processes is a challenging but potentially fruitful task with many practical applications in complex domains. Modelling these data interactions, learning about them, extracting knowledge, and building a reusable knowledge base applying state-of-the-art AI and soft-computing will guide future research and practice that is at the core of this research.

In spite of the semantic power of ontologies, relationships, concepts, and understanding evolve over time. Ontology specifications must be able to represent the dynamics and changes of the real world otherwise there is a danger that building and using an ‘inflexible’ ontology, that is, an ontology that has a fixed structure, may fossilize current knowledge representation in a particular paradigm or point of view, so that only information that fits the paradigm is accepted by and accessible to the users.

The development and use of evolving ontologies is an attempt to solve these ‘static’ and ‘domain silo’ ontology challenges. This work defines evolving ontology as a partial, shared conceptualisation that supports knowledge discovery from databases as defined in this chapter. Partial means incomplete knowledge, as it is constructed, dynamic, changing, and evolving. Shared means consensual knowledge, maintained through possibly distributed knowledge management.

The evolving ontology meta-knowledge specifies a set of components able to cope with changing requirements. The meta-data is responsible for creating a set of schemes to annotate each concept in an ontology. The meta-class defines a minimum set of attributes responsible for keeping provenance of source, creation and date, allowing constant change management. The model integration feature establishes different methods for incorporating knowledge and keeping track of source changes. The knowledge map defines a meta-knowledge and a knowledge acquisition tool able to acquire knowledge in manual, automatic and semi-automatic ways as well as temporary storage for knowledge validation by humans and inference engines.

Although the Evolving Ontology model is a general model that can be applied to different domains, this thesis has mainly focused its application in the biomedical informatics domain. The selection of this domain was due to the complexity involved in biomedical knowledge discovery and the amount of research developed at the Knowledge Engineering and Discovery Research Institute (KEDRI) where this research was carried out.

Other projects have also adopted the Evolving Ontology model to various degrees, such as in the biblio-mining field (Parikshit, Pears, & Gottgtroy, 2006), ontology visualisation research (Wang & Gottgtroy, 2006), the application of learning objects as evolving models (Kasabov, Jain, Gottgtroy, Benuskova, & Joseph, 2007), Advancing Clinico Gnome Trials on Cancer: Open Grid Services for improving Medical Knowledge Discovery (Rüping, 2007), and in the design a common data model for autonomous vehicle (Davis, 2006).

Chapter 4

A methodology for knowledge discovery from Evolving Ontology

This chapter describes the Ontology Driven Knowledge Discovery process (ODKD). The primary outcome is a novel methodology and life cycle that considers the best practices of ontology engineering and knowledge discovery in databases (KDD) processes in order to answer the research question - ‘Can we integrate Ontology Engineering and KDD processes?’ As a process guideline this chapter does not concentrate on concrete examples of the application of the methodology. Such examples are illustrated in the next two following chapters and in Appendix B.

4.1. Introduction

This chapter is concerned with the interaction between prior knowledge (by means of ontologies) and the process of knowledge discovery. It describes a novel methodology and process model which integrates ontology engineering and KDD.

Instead of looking specifically at the application of ontology for KDD, such as the use of prior knowledge to improve results of clustering algorithms (Jinze, Wei, & Jiong, 2004), or to the application of data mining techniques for ontologies, such as ontology learning from text, the focus of this chapter is on the closed loop integration of ontologies in KDD processes.

This chapter complements the previous one (Evolving Ontology) in order to construct the conceptual contribution of this thesis. It is part of the design science methodology of the multi-methodological approach followed in this work (Figure 0-2) and discussed in Chapter 1.

Examples of the process followed in the development of the case study utilizing the methodology proposed in this research are presented in this Chapter to briefly illustrate the application of the methodology. Further detailed information about the application of the methodology is also presented in Chapter 6 and in Appendix B.

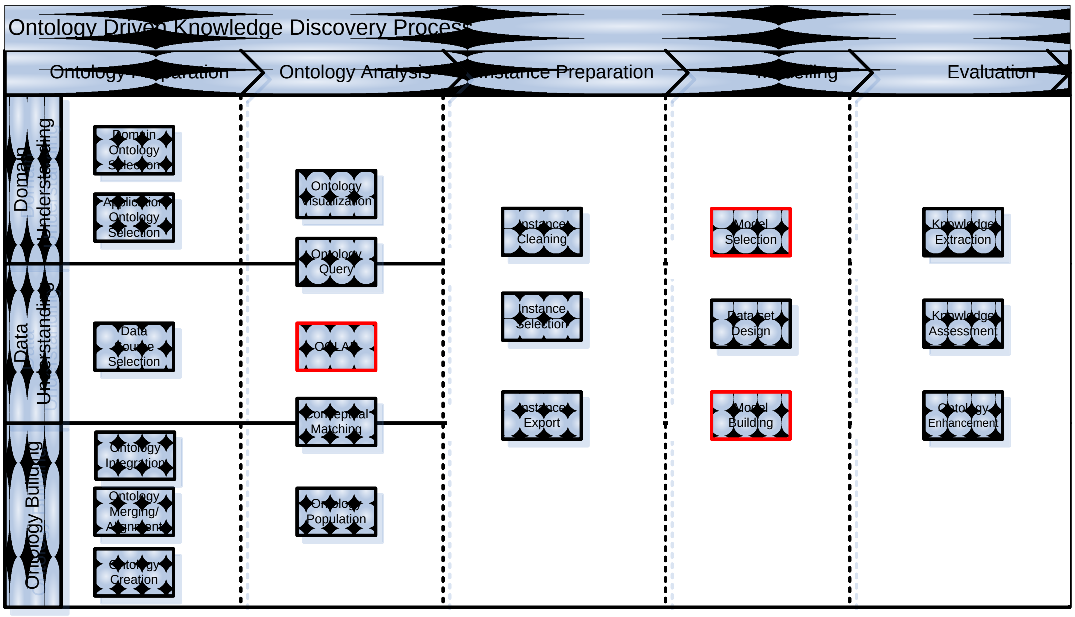
4.2. Ontology Driven Knowledge Discovery - ODKD

The Ontology Driven Knowledge Discovery (ODKD) is a methodology and process model that defines, at different levels, a life cycle integrating ontology engineering and knowledge discovery in database (KDD) tasks. It defines a novel hybrid life cycle which is lacking in the research field of integration of ontologies and knowledge discovery as described before.

The life cycle is built based on best industry practices from a data mining point of view and implements best design practices from multidimensional modelling and data warehouse design perspectives. The hybrid life cycle proposed is novel in both adopting standards and industry practices in ontology engineering as well as in proposing a methodology and process model for ontology building. It also advances the research on ontology engineering and its integration with KDD by proposing a full life cycle to close the loop in an ontology driven knowledge discovery process.

The hybrid life cycle proposed here is composed of five phases. Each phase is divided into tasks related, directly or indirectly, to ontology-specific building tasks and to CRISP-DM. The methodology and processes support the implementation of the life cycle tasks.

Table 4-1 The proposed Ontology Driven Knowledge Discovery phases and tasks.[6]



Although related to CRISP-DM from a "data mining" perspective, ODKD differs from it by being concerned with data at an abstract level – the concepts. It considers data as a representation of knowledge within a domain. In this way, it is looking at the data from a data-modelling perspective where the database is a formal representation of the domain knowledge, containing useful data and metadata within

[6] The tasks highlighted in red correspond to tasks not specifically covered in the thesis such as the exploratory work done investigating the integration of ontologies and Online Analytical Process (Appendix A) or tasks considered outside of the scope such as modelling which are exclusively related to data mining. The tasks appear in the life cycle as a reference since it is not the intention of this work to discuss which modelling technique, for example, is most appropriate to a given data mining exercise. Additional research is present in the literature in order to use ontologies as a way of informing best data mining techniques, this approach is more focused on ontology for KDD which is not part of the scope of this thesis as explained previously.

the model. This perspective aligns with the artificial intelligence view in which concepts are the most basic units of thought. Concepts are thus the building blocks and they and their relationships can represent knowledge about a domain.

As with CRISP-DM, ODKD is a cyclical process. Moving back and forth between different phases is always necessary. When analyzing real and complex problems, the knowledge discovered in one KDD task can be reused in another task, and this chain might define a decision-making workflow. ODKD also considers knowledge (represented as concepts and their relationships) as the starting-point of any knowledge discovery exercise; for example, the knowledge extracted from a microarray chip based on gene expression analysis of a diseased tissue can be used as input for a drug discovery investigation for a completely different disease.

ODKD then, does not presume that a ‘complete’ and sufficient dataset is available at the time a new data mining application is started. On the contrary, at a conceptual level, it considers that every discovery task generates new insights and therefore new domain knowledge might be required. The required knowledge will bring new types of variables, which will improve the understanding of both the problem and the domain.

Furthermore, this research also reflects the view that data mining is not simply the one-time application of a program to a new dataset. In this work, data mining is an ongoing hypothesis-checking exercise. It frequently starts with small pilot studies and manual space search, including feature construction. With a preliminary confirmation that the process can find some interesting relationships, more data and greater expectations are introduced, leading to another round of investigation utilizing a larger amount of data, or at least a better understanding of the problem.

In order to consider this dynamic behaviour, this research applies the guidelines of the Cross-Industry Standard Process for Data Mining (CRISP-DM) to the ODKD life cycle definition.

As in CRISP-DM, the ontology driven knowledge discovery process hierarchically breaks down into tasks (as shown in Table 4-1). These tasks are responsible for the execution of specific ontology engineering and data modelling processes as well as for enabling a parallel integration of both knowledge and ontology engineering processes.

Some of the phases such as ontology preparation and evaluation might be compared with individual tasks proposed in the literature such as ontology creation or ontology integration and probably considered not novel. However, some of these previously identified tasks have been proposed just under the umbrella of a general ontology engineering approach and sometimes even as just possible independent steps of a life cycle. Therefore the ontology driven process proposed in this research must be viewed in the context of the integration of Ontology engineering and KDD. The ODKD then advances the research by adopting and integrating best practices in a closed loop process model as well as by proposing new phases and tasks related to the requirements of a KDD process, such as instance preparation and ontology analysis.

There are, then, five phases in the Ontology Driven Knowledge Discovery process: Ontology Preparation, Ontology Analysis, Instance Preparation, Modelling, and Evaluation.

The first two phases, Ontology Preparation and Ontology Analysis, are closely related to the ontology engineering process of assessment, selection and creation of an ontological model able to represent a domain and a problem. These phases are subdivided into three pipelines as seen in Table 4-1: domain understanding, data understanding, and ontology building. Both domain and data understanding are related to the understanding of the problem through, for example, the selection of the knowledge/data source for the target problem and visualisation of the ontologies, while ontology building is related to the creation and population of ontological models.

The Instance Preparation and Modelling phases are related to the data mining exercise, with Instance Preparation particularly concerned with preparing data for a modelling task while selecting the most appropriate data mining technique for the problem. The last phase of the ODKD process, Evaluation, is the most integrative phase, where the results of a modelling exercise must be evaluated based on previous knowledge and new instances are inserted into the ontological model.

ODKD’s phases are not one-to-one mapped to CRISP-DM’s phases. Due to specific ontology engineering requirements, some ODKD phases are designed to cover more than one CRISP-DM phase. For example, the Ontology Preparation phase, which includes the ontology and data focuses, is related to CRISP-DM's business and data understanding.

On the other hand, when broken down into tasks both ODKD and CRISP-DM phases have their tasks aligned in order to facilitate use by the industry and to avoid misinterpretation.

As a conceptual contribution, this and subsequent sections describe an optimistic in-principle portrayal of the ODKD process. The methodology, as described, proposes an abstract and 'ideal' way (as do all methodologies) and assumes that all will proceed without major obstacles or delays. The next sections then should be read as a process guideline.

Figure 4-1 presents the ODKD process model. The diagram distinguishes the ontology engineer and domain expert role as main users; however other roles are also involved and will be described in the text in the following sections. The document outputs derived from the ontology preparation phase are also represented. Some phases are more sequential such as the ontology preparation others don't restrict the order of the tasks such as the ontology analysis phase.[7].

[7] The process model is using a BPMN 1.2 notation. Refer to http://www.bpmn.org for notation information.

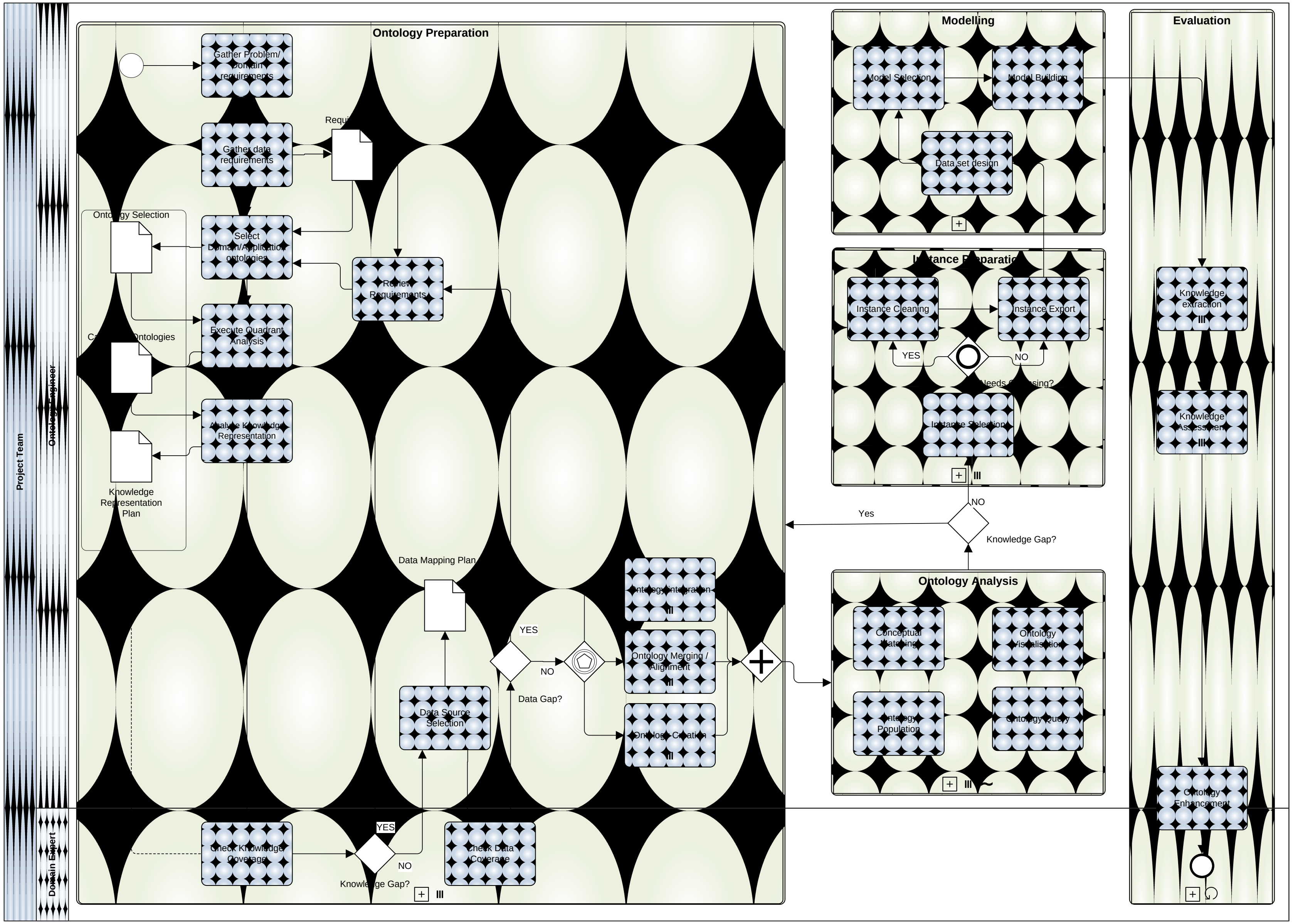


Figure 4-1 – ODKD process model diagram

The next sections describe in detail the ODKD phases and discuss the constituent tasks involved in an ontology driven knowledge discovery exercise. It considers an ideal process in order to facilitate the understanding of the methodology. As noted before. in practice, the methodology may be more challenging in terms of numbers of iterations and tasks effort, such as in the case of a lack of consensus when understanding a domain which may require an extra number of iterations as well as in the instance selection phase where a number of iterations and cleansing is normally necessary depending on the project requirements and quality of the instances previously acquired in the ontology building pipeline.

4.2.1. Ontology Preparation

The prerequisites of a knowledge discovery exercise are data and contextual understanding. Without data and contextual understanding, no algorithm, regardless of its sophistication, is going to provide meaningful or useful results. Without this understanding a user/system will not be able to identify the problems he/she/it is trying to solve, prepare the data for mining, or correctly interpret the results.

This initial phase focuses on understanding the knowledge discovery objectives and the domain onto which it is going to be applied. It defines requirements from a context perspective and acquires initial domain knowledge through the investigation and selection of related domain and application ontologies.

The ODKD methodology is initiated by a requirement-gathering exercise which involves several stakeholders such as subject matter experts, business users and the business problem owners, followed by the selection of problem related data and ontology sources. The ODKD methodology suggests a dual-pronged approach to the requirement gathering sessions: meeting sessions with the business user representatives and subject matter experts (SME) to gain knowledge about the problem/domain, and data sessions with source system experts and SMEs when available.

The first round of interviews with business users forms the basis for building an understanding of the problem. It is followed by data sessions where the goal is to assess the existent data in order to define the data requirement. At the end of these initial interviews, a facilitation meeting is scheduled to reach a group consensus around problem definition and data requirements.

After reaching a consensus around the problem and the source requirements, the ontology preparation starts the process of source selection which will form the basis for the construction of the ontological model and subsequent instance population.

The source selection is composed of three tasks: ‘domain ontology selection’, ‘application ontology selection’ and ‘data source selection’. *Domain ontology selection* aims to select knowledge that covers a broader problem perspective, for example, general biomedical ontologies when dealing with a medical problem. On the other hand, *Application Ontology Selection* aims at selecting specific ontological models for a problem, for example, ontologies related to a specific disease when classifying/diagnosing patients, such as the Leukaemia case study described in Chapter 3.

The selection process is similar for both domain and application ontologies. It involves several key steps, including assessing and selecting candidate ontologies (ontology investigation), knowledge quadrant analysis, and a knowledge representation mapping plan.

The *Data Source Selection* then focuses on the candidate ontologies to identify data source candidates from a technical and availability point of view as described later in this chapter.

In this research, biologists, bioinformaticians and neuro-scientists were engaged in the process of ontology preparation. The definition of the domain area and application domain guided the selection of initial candidate ontologies and databases and also guided the selection of initial case studies, including the Leukaemia case study described previously.

Ontology Selection

In this task, various available ontology sources are searched and investigated. The goal is to find ontological models that cover the knowledge required to represent the problem from both broad and narrow perspectives. The outcome is a document containing a list of all available ontologies related to the problem/domain and some specific properties of each.

The ontologies are annotated with the following minimum set of properties: source identification, type (domain or application), representation formalism, and availability and knowledge coverage (from the ontology engineer initial perspective). Depending on the KDD task and on characteristics of the problem some extra information might also be required, such as cost for acquisition, licensing, and privacy. This extra information might help in making a decision about the adoption of specific ontologies as well as in the planning phase of the KDD exercise.

Along with the compilation itself, another very important outcome of this exploratory process is the gain in domain understanding and consequent refinement of the problem statement. This gain is due to the fact that a domain can frequently be modelled in several ways and from different perspectives in ontologies; for example, there are various biomedical ontologies available now, some encompassing chemical and/or genetic knowledge while others are focused on nutritional and environmental factors. These multiple perspectives enrich the background knowledge of a problem, adding to the already established expert knowledge.

There is no limitation to the number of ontologies gathered in this task, nor should any technical or representational constraint should be considered at this stage. The gene ontology and UMLS were the first candidates selected in this research due to their extensive coverage of biomedical knowledge. All considerations, constraints and factors are analysed in the knowledge quadrant analysis, as follows.

Knowledge Quadrant Analysis

The ontology investigation document produced in the previous task serves as the basis for the knowledge quadrant analysis. The analysis begins with a brief presentation of the compilation of ontologies available by the ontology engineer. The KDD team then discusses all the candidate ontologies from a contextual and technical perspective, adding pertinent comments to the list. This first interaction aims to reach a common understanding of the problem, knowledge source availability and relevant source characteristics.

Once the ontology list is reviewed, prioritization begins. ODKD's methodology employs a simple and flexible quadrant analysis technique (see Figure 4-2). The technique adopted is similar to techniques widely used in data warehousing and IS development to prioritize business processes to be implemented. The vertical axis refers to the knowledge coverage of an ontological model. The horizontal axis might vary in accordance to the problem, but normally refers to the availability of the ontology.

The ODKD methodology also suggests the adoption of a second quadrant analysis which encompasses knowledge coverage and the possibility of its use, expressed as 'feasibility'. Feasibility takes into consideration the problem requirements, the resources available and the constraints for the implementation of a successful KDD task.

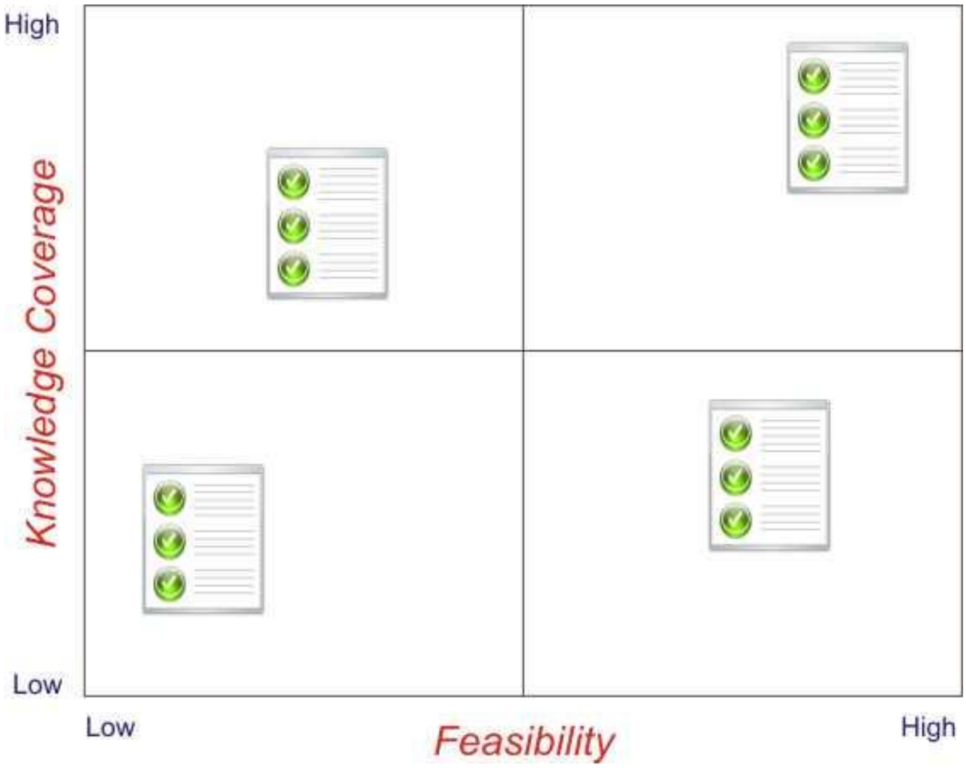


Figure 4-2 - Knowledge coverage vs. feasibility

This additional quadrant analysis is especially important in problems where a broad commitment is necessary due to financial considerations, time restrictions, technical resources and/or contractual constraints. The Unified Medical Language System (UMLS) selected in this research, for example, required the establishment of a contractual agreement for its use and extension. This also helps in the risk analysis process, where major risks can be identified and/or overcome. (Although not adopted in our experiments, there is some evidence that complex scenarios may justify the use of a three dimensional graph for the prioritization process. It would include the three axes: knowledge coverage, availability and feasibility.)

Ontologies positioned in the upper right quadrant are selected and discussed further since they hold crucial knowledge representation of the domain. Ontologies in the upper left quadrant are annotated with the requirements that would improve their availability and feasibility. Ontologies in the lower right quadrant are annotated for possible/secondary use as their coverage might not be adequate. Ontologies in the lower left quadrant are discarded as they do not represent the domain/problem sufficiently and are difficult to use.

Assuming not all candidate ontologies are discarded, a final list is compiled into a document representing the consensus reached and the additional tasks required to improve the possible use of one or more ontologies. These required tasks might also be part of the risk assessment document used for project management. The final document is then distributed to all participants.

It is important to note that this document will probably change as further requirements will be introduced later in the life cycle. However the analysis conducted at this stage represents a common understanding between all project participants and guides further ontology adoption later in the ODKD process.

This iterative process of identifying ontologies and data sources was strongly evident in the development of this research. The initial set of ontologies was collected and then further application ontologies were included as needed, such as those related to specific brain disease knowledge as described in Chapter 6.

Knowledge Representation Mapping Plan

One aspect garnered from the ontology investigation task is the representation formalism. Although not explicitly considered in the quadrant analysis, as knowledge coverage is the main factor of interest, it is very likely that ontologies selected in the quadrant analysis are modelled on different formalisms. Thus a technical analysis of the ontology formalism is needed. The *knowledge representation mapping plan* is responsible for this analysis.

For each ontology selected as a candidate the ontology engineer annotates along with the knowledge representation formalism the actions required to translate all the various representation formalisms into a common formalism. For example, in some cases it might be necessary to develop specific import/translation mechanisms, in other cases the KDD and/or the ontology environment might already have these features. The next chapter describes the environment developed to implement the ODKD which is able to import most of the currently available ontology formats.

At the end of this task, a plan for the incorporation of the selected ontologies is finalized and serves as basis, alongside the data source selection described in the next section, for the ontology building pipeline which will create the ontology model for the KDD task.

Data Source Selection

In terms of ontology preparation, the data understanding pipeline consists of a Data Source Selection task. This task starts with an initial data collection based on the results of data sessions previously executed in the requirement gathering exercise and ontology selection. As in the ontology–related tasks, the data source selection proceeds with activities to achieve familiarity with the data, to identify data quality problems and to map the data to the ontologies selected.

The main outcome of this pipeline is a data mapping document which assesses the data available, and establishes possible links among the data collected and different branches of the ontologies selected. This document is used as a guideline in the ontology population and data integration tasks of the ontology analysis and instance preparation phases respectively (both phases are described later in this chapter). A number of candidate data sources were selected in the biomedical case study described, such as gene cards, NCBI and PubMed for annotations (a complete list is provided in Chapter 6).

This task also helps in the discovery of the first insights into the data and contributes to the detection of interesting subsets for forming hypotheses for hidden information. These insights normally trigger new knowledge acquisition exercises which can suggest the creation of new concepts in the subsequent ontology building tasks.

Ontology building is the last step in the ontology preparation phase. This pipeline consists of three main tasks: Ontology Integration, Ontology Merging/Alignment and Ontology Creation. Ontology building is concerned with the integration, inclusion and importation of the ontologies selected in the quadrant analysis and/or described in the knowledge representation mapping, and the creation of the concepts identified in the data mapping plan.

As described in Chapter 3, the process of building ontologies is underpinned by the concept of Evolving Ontologies, therefore the extension or inclusion of new knowledge (concepts and relationships) is facilitated.

Ontology Integration

Ontology integration is the incorporation of the ontologies identified by the domain understanding pipeline. The main objective of this task is to reuse ontologies already developed and identified as available in the previous selection process, in order to add a broader perspective to the problem domain. This approach can be observed in the integration of the Gene Ontology as described in the Leukaemia case study and further extended in the biomedical case study presented in Chapter 6 and Appendix B.

There are two main ways to incorporate and reuse ontologies in the ODKD methodology: ontology import/inclusion and ontology annotation. To reiterate, the former is concerned with the incorporation of a complete ontology and the latter is related to the incorporation of specific concepts of a selected ontology source.

Ontology Merging/Alignment

In short, ontology merging/alignment is *"a mapping of concepts and relations between two ontologies"* (Sowa, 2005). This task is a research topic in itself with various approaches described in the literature (Euzenat et al., 2004). In the context of the ontology driven knowledge discovery methodology, the main goal is to define a task responsible for the incorporation of existent ontologies to form a body of concepts and relationships able to represent the domain and the specific problem as defined by the ontology selection task.

The next chapter describes the framework and the merging technique available in the ODKD implementation framework. Some extra merging techniques are also briefly presented in Chapter 6 where a biomedical application is described.

Ontology Creation

In spite of the current availability of large ontologies covering a very wide range of knowledge in several domains, it is very likely that the ontologies acquired in the ontology merging/alignment phase will not fully cover the knowledge necessary for a given KDD task. In the biomedical discovery field, for example, even though the amount of knowledge provided by the Gene Ontology and the National Center for Biotechnology Information (NCBI) Ontology is immense, there will probably be a set of concepts and relationships that are not covered or well defined for an investigation of a specific disease (Bodenreider, 2002). As illustrated in subsequent chapters, several concepts needed to be defined in this research in the context of the Brain Gene Onotlogy.

Ontology creation is then concerned with the incorporation of problem-specific knowledge and the creation of concepts not covered by the ontological model built in the previous task (merging/alignment). It is also responsible for the translation of different formalisms into the formalism defined for the ontological model being created.

The creation task might be considered as a specific manual process of building concepts and relationships not incorporated through the inclusion of the selected ontologies. As in the ontology merging/alignment task, there are different ways for creating an ontological model. The literature suggests various ontology building methodologies (Gómez-Pérez, Fernández-López, & Corcho, 2004).

The literature and early experiments developed in this research suggest that the best building methodology for a KDD task is highly dependent on the problem, ontology application tool and the representation formalism, among other things. However it has also been suggested that the adoption of a methodology to build a first ontology ‘skeleton’ alongside the integration of the various ontologies in the merging phase is very productive.

This research has adopted the Significant Conceptual Modelling Process (Gottgtroy & Gottgtroy, 2001). Instead of concentrating on the engineering problem, this methodology focuses on the identification of relevant initial concepts of a problem domain able to link the knowledge acquired with the mental model of the ontology engineer and business problem owners. The technique is based on the Physiology Education Theory of David Ausubel. A primary process in modelling is subsumption in which new material is related to relevant ideas in the existing cognitive structure on a substantive, non-verbatim basis. (Detailed discussion of this modelling technique is out of scope of this research, further material can be found in the paper or on Ausubel’s theory site.)

In summary, this first phase (ontology preparation) begins with the assessment and selection of both available ontologies and data sources relevant to the target problem/domain. It is followed by the construction of an ontological model able to evolve and support a KDD task.

After this initial conceptual modelling task, a detailed analysis of the knowledge representation (ontological model) is undertaken in order to gain domain knowledge and to populate the model with data/ontology instances to form the knowledge base.

As noted before, the ontology driven methodology and process model proposed in this research is a guideline which focuses on the necessary ontology engineering tasks to build a hybrid ontology driven knowledge discovery process. As a guideline it recognizes that the outcomes, such as a data mapping plan, will be integrated within normal project management practices to highlight dependencies and risks in a project development. Therefore the ODKD does not specify a set of templates. It is focused on the tasks and minimal set of information necessary to capture the requirements to build an ontology in the context of KDD. The practice of this research has shown that the documentation used has become live documents which are enhanced through the life cycle of the projects (which is consistent with CRISP-DM and other methodologies).

4.2.2. Ontology Analysis

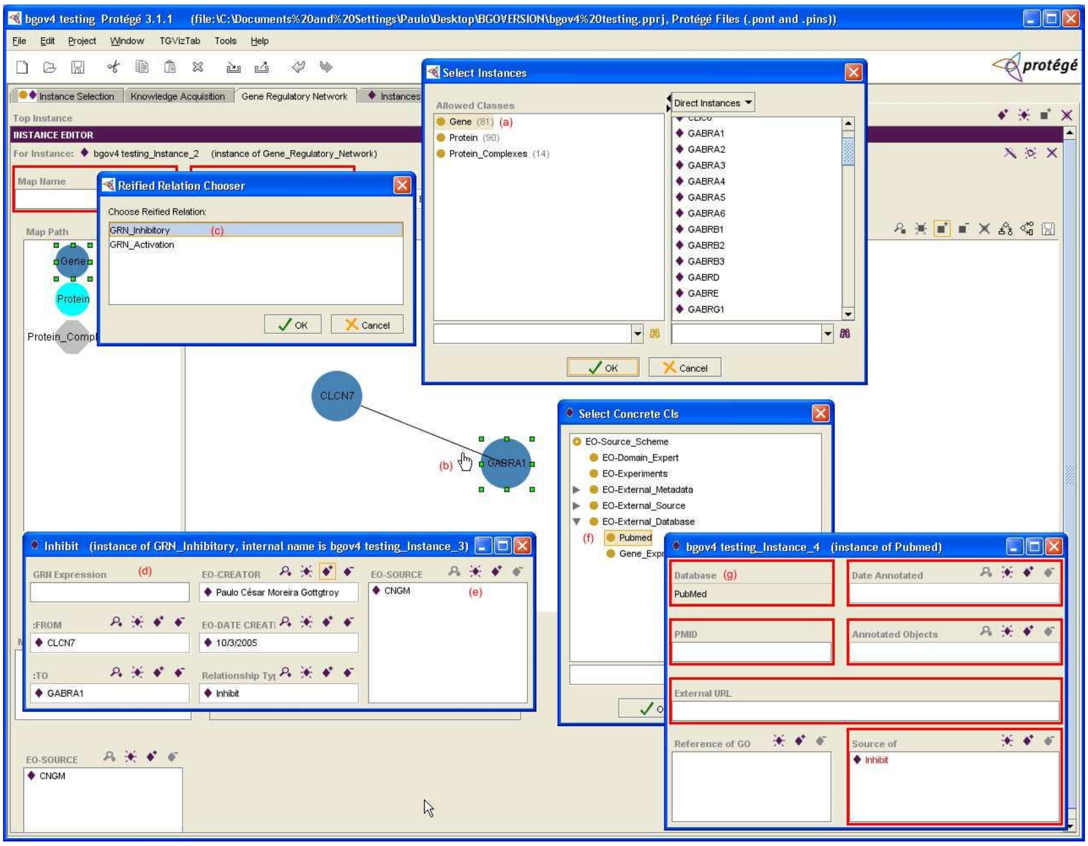
The Ontology Analysis phase is related to the discovery of the first insights into the ontological model as well as to the investigation and/or checking of the initial hypotheses created through the disclosure of hidden information in the developed model.

As depicted in the table 4-1, the ODKD process defines four tasks in this phase: Ontology Visualization, Ontology Query, Conceptual Matching and Ontology Population. The first two tasks are related to the exploration of the ontological model by means of visualisation, searching and analytical process. The latter two tasks are related to the construction of the knowledge base through the insertion of data from the available databases and the population of instances from the selected ontologies.

Ontology Visualisation

Ontology visualisation is the analysis of the knowledge representation by means of visualisation techniques. The goal of this task is to facilitate the exploration of the ontology from different perspectives and the navigation of the emerging complex knowledge network.

There are different ontology visualization techniques available in the literature, and some built into ontology environments (Wang & Gottgtroy, 2006). The selection of the most appropriate technique is highly dependent on the problem's characteristics and on the ontological model (Wang, 2006), different problems and/or knowledge representations may require extra or specially designed visualisations. Therefore, the ODKD methodology suggests the adoption of at least two techniques in a KDD process: tree navigation and network visualisation.



bgov4 testing Protégé 3.1.1 (file:\C:\Documents%20and%20Settings\Paulo\Desktop\BGOVERSION\bgov4%20testing.pprj, Protégé Files (.pont and .pins))

For Instance: bgov4 testing_Instance_2 (instance of Gene_Regulatory_Network)

Reified Relation Chooser

Choose Reified Relation:

GRN_Inhibitory (c)

GRN_Activation

Select Instances

Allowed Classes: Gene (81) (a), Protein (90), Protein_Complexes (14)

Direct Instances: GABRA1, GABRA2, GABRA3, GABRA4, GABRA5, GABRA6, GABRB1, GABRB2, GABRB3, GABRD, GABRE, GABRG1

Select Concrete Cls

EO-Source_Scheme
- EO-Domain_Expert
- EO-Experiments
- EO-External_Metadata
- EO-External_Source
- EO-External_Database
  - (f) Pubmed
  - Gene_Expr

Inhibit (instance of GRN_Inhibitory, internal name is bgov4 testing_Instance_3)

GRN Expression (d)

EO-CREATOR: Paulo César Moreira Gottgtroy

EO-SOURCE: CNGM (e)

:FROM: CLCN7

EO-DATE CREATI: 10/3/2005

:TO: GABRA1

Relationship Ty: Inhibit

bgov4 testing_Instance_4 (instance of Pubmed)

Database (g): PubMed

Date Annotated

PMID

Annotated Objects

External URL

Reference of GO

Source of: Inhibit

EO-SOURCE: CNGM

Figure 4-3 - Several visualizations available in the BGO System: A sample of a GRN editing process: (a)the CLCN& and GABRA1 genes are created (b) a relationship is created between these genes (c)an inhibitory behaviour is selected (d) the new inhibitory relationship is opened (e) an annotation is added (f) the source of the annotation is selected (g) a PubMed annotation is created.

Ontology Query

In spite of the advantages of exploring a complex knowledge structure by means of visualization, a query mechanism is still required. This is evident when navigating in domains where concepts are unfamiliar or the taxonomy varies in accordance with knowledge background. The bioinformatics field, for example, is a multidisciplinary subject, which inherits concepts from biology, chemistry, medicine among others;

therefore querying a concept by synonyms and their relationships helps to rapidly increase understanding of the domain.

Ontology query is therefore a task used in the ODKD methodology that enables navigation in an ontological model by using ‘questions’ to verify hypotheses and/or select instances from a knowledge base. It is implemented in the framework and is used to facilitate understanding, for example, it helps to answer questions such as; ‘what are the genes located in part X of the brain? ‘which genes are highly expressed in the brains of patients with disease Y?’.

The next chapter describes and presents the visualization techniques and query mechanism adopted in the ODKD. Appendix B shows some associated visualization and querying screenshots related to the biomedical case study (Chapter 6).

Conceptual Matching

Conceptual Matching is the task that maps a database to an ontological model. After the construction of an ontological model a database model is mapped to the ontological model and the database records are then imported as instances of the ontology to build the knowledge base. This conceptual matching gives semantic meaning to the data. This has been referred to as the major contribution of the so called semantic technologies (Foster, 2006).

There are various approaches to matching ontologies and databases (Gottgtroy, Kasabov, & MacDonell, 2003b). Instead of trying to link a record at the level of data, as developed by syntactical matching techniques such as XML mapping, conceptual matching integrates a record at its conceptual level, leveraging both the integration as well as the understanding of a data model.

This task plays a major role in the Ontology Driven Knowledge Discovery methodology by mapping the databases identified in the *data mapping plan*, created in the data understanding pipeline (data understanding section), to the ontological model selected during quadrant analysis. It also adds another conceptual layer to the data by integrating different sources of data to the same ontological representation.

In this research publications were inserted into the ontological model using gene synonyms and GO-terms to match the data sources and publications to the ontology. RNA expression data acquired from the GNF expression database, for example, were also inserted in the brain gene ontology (as described in Table 6-1).

Ontology Population

This task populates a knowledge base with instances of the ontologies previously merged and/or aligned. It is the next natural step after the conceptual model matching described in the previous section.

In brief, after the exploration of the ontological model and/or after the gaining/acquisition of new knowledge, the knowledge base can be populated with instances from the ontologies studied. As an example, after the Gene Ontology analysis a researcher might decide to incorporate into a knowledge base instances of genes found in the genomic database of a plant, as well as those found in an animal that are related to the disease being investigated in a KDD exercise. These instances will then build linked knowledge, integrating knowledge from different experimental contexts.

The experiments undertaken in this research have shown that instance scoping might also be needed in order to prune the amount of data currently available in the domain ontologies. There will always be an amount of information, represented by instances, which is not, in principle, relevant to a specific problem; therefore instead of incorporating a full set of instances from a domain ontology, the best practice is to incorporate instances after the initial knowledge discovery task. The results will then indicate the need for new instances of even newer knowledge.

In this research, for example, a Gene Ontology Slim version relevant to brain gene disease knowledge was selected and populated in the brain gene ontology built as part of the biomedical case study.

Chapter 6 describes the integration of knowledge and KDD results into one biomedical ontology (Gottgtroy, 2003; Verma, Gottgtroy, Havukkala, & Kasabov, 2006). The ontological model and its knowledge base aim to link knowledge about genes, nutritional characteristics and the brain in order to link fields, which are, in principle, unrelated, but in the light of further investigation may show a strong relationship in their environmental, nutritional and genetic aspects.

Figure 4-1 summarizes the Ontology Analysis phase and its respective tasks where the exploratory work is done and data and instances are acquired from the ontologies and databases identified during Ontology Preparation.

4.2.3. Instance Preparation

The instance preparation phase covers all activities that construct the final dataset that will be fed into the selected modelling tool(s). Instance preparation tasks, as data preparation in conventional data mining life cycle, are likely to be executed multiple times depending on the model being targeted. Tasks include concept, instance and slot (or attribute in a database taxonomy) selection, transformation and the cleaning of instances as well as exportation to modelling tools.

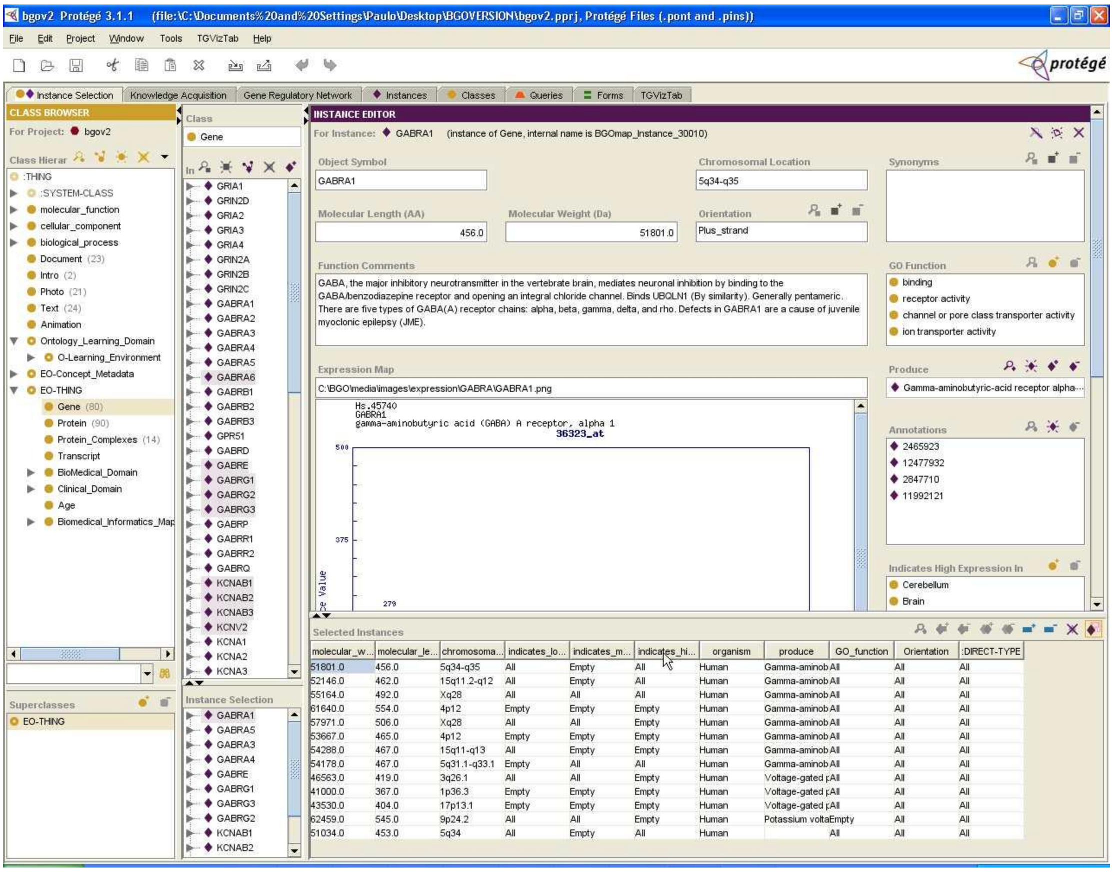
This phase leverages the power of ontologies in the ODKD methodology by adding a semantic layer to the data. It supports selection of the best candidates for a data mining exercise from multidimensional perspectives, reducing the features needed for a successful KDD task. The ontology driven methodology is composed of three main tasks in this phase: Instance Cleaning, Instance Selection and Instance Export.

Instance Cleaning

Instance cleaning can be thought of as a special sort of data cleaning, also called data cleansing or scrubbing. It deals with the detecting and removing of errors and inconsistencies from instances in order to improve the quality of features being selected for a modelling task (Rahm & DO, 2000).

Instance quality problems are mainly inherited from errors present in data collections, such as files and databases, e.g., due to misspellings during data entry, missing information or other invalid data. Other sources of instance error are incomplete information and invalid information.

When multiple knowledge/data sources are integrated in an ontological model the need for data cleaning increases significantly due to the diversity of sources. This is because the sources often contain redundant information and/or the uncertainty around the information is high, due to a lack of evidence to support it. Instance cleaning is concerned then with providing access to accurate and consistent data.



For Instance: GABRA1 (instance of Gene, internal name is BGOmap_Instance_30010)

Object Symbol: GABRA1

Chromosomal Location: 5q34-q35

Synonyms:

Molecular Length (AA): 456.0

Molecular Weight (Da): 51801.0

Orientation: Plus_strand

Function Comments: GABA, the major inhibitory neurotransmitter in the vertebrate brain, mediates neuronal inhibition by binding to the GABA/benzodiazepine receptor and opening an integral chloride channel. Binds UBQLN1 (By similarity). Generally pentameric. There are five types of GABA(A) receptor chains: alpha, beta, gamma, delta, and rho. Defects in GABRA1 are a cause of juvenile myoclonic epilepsy (JME).

GO Function:
- binding
- receptor activity
- channel or pore class transporter activity
- ion transporter activity

Expression Map: C:\BGO\media\images\expression\GABRA\GABRA1.png

Produce: Gamma-aminobutyric-acid receptor alpha-...

Annotations:
- 2465923
- 12477932
- 2847710
- 11992121

Indicates High Expression In:
- Cerebellum
- Brain

Selected Instances

molecular_w...	molecular_le...	chromosoma...	indicates_lo...	indicates_m...	indicates_hi...	organism	produce	GO_function	Orientation	:DIRECT-TYPE
51801.0	456.0	5q34-q35	All	Empty	All	Human	Gamma-aminob	All	All	All
52146.0	462.0	15q11.2-q12	All	Empty	All	Human	Gamma-aminob	All	All	All
55164.0	492.0	Xq28	All	All	All	Human	Gamma-aminob	All	All	All
61640.0	554.0	4p12	Empty	Empty	Empty	Human	Gamma-aminob	All	All	All
57971.0	506.0	Xq28	All	All	Empty	Human	Gamma-aminob	All	All	All
53667.0	465.0	4p12	Empty	Empty	Empty	Human	Gamma-aminob	All	All	All
54288.0	467.0	15q11-q13	All	Empty	Empty	Human	Gamma-aminob	All	All	All
54178.0	467.0	5q31.1-q33.1	Empty	All	All	Human	Gamma-aminob	All	All	All
46563.0	419.0	3q26.1	All	All	Empty	Human	Voltage-gated p	All	All	All
41000.0	367.0	1p36.3	Empty	Empty	Empty	Human	Voltage-gated p	All	All	All
43530.0	404.0	17p13.1	Empty	Empty	Empty	Human	Voltage-gated p	All	All	All
62459.0	545.0	9p24.2	All	All	Empty	Human	Potassium volta	Empty	All	All
51034.0	453.0	5q34	All	Empty	All	Human		All	All	All

Figure 4-4 - A set of data selected in the instance selection tool showing a number of missing values (described as "Empt") as candidates for cleaning.

The ODKD methodology also supports the instance cleaning task through the use of quadrant analysis to classify the candidate ontologies for the domain and problem representation as well as through ontology analysis when selecting instances to populate the knowledge base. This selection should reduce problems related to the quality of the information acquired as well as allowing the identification of candidate instances for the cleaning task.

Instance Selection

The instance selection task brings one of the most important contributions to the ontology driven knowledge discovery process. Instead of concentrating effort on analysing data characteristics, this task concentrates its effort on giving meaning to data and reducing the number of features for a data mining task.

Opportunities arise from the simple act of connecting different facts, instances and points of view that have been created for one purpose, but in light of ontological analysis, can be reused in a quite different context, to form new data set for modelling and creation of new hypothesis. In gene expression analysis, for example, the number of samples may be small while the number of genes can be huge, thus any support in identifying those genes significant to the problem analysis is very important. Looking at gene expressions in different tissues and parts of the body, as well as at the molecular function level, might cluster genes into small classes which may reduce the amount of computing time needed for a modelling task, and can provide optimum meaning for the results. Instance preparation in this example plays an important role in acquiring previous knowledge that can help the bioinformatician in the selection of such features.

Instance and feature selection have become the focus of much research in areas of application where datasets and knowledge bases with tens or hundreds of thousands of variables are available. The objective of the ontology driven instance selection task is to improve the prediction performance of predictors, providing faster and more cost-effective predictions, by providing a better understanding of the underlying process that generated the data and its conceptual interpretation.

The instance selection tool can also initially select instance candidates which are then exported (described in the next section) for further analysis in data mining workbenches. For example a set of instances can be selected in the ontology and then exported to NeuCom for further analysis by the *evolving clustering algorithm* (Kasabov, 2008) as follows.

The ECM: algorithm:

- Create the first cluster C_1 by simply taking the position of the first instance from the exported data as the first cluster centre C_{c1}, and setting a value 0 for its cluster radius $R_{u1;}$

- If all instances from the exported data have been processed, the clustering process finishes. Else, the current input instance, x_i, is taken and the normalised Euclidean distance D_{ij}, between this instance and all n already

In this way, the maximum distance from any cluster centre to the farthest instance that belongs to this cluster is kept less than the threshold value, *Dthr*, though the algorithm does not keep any information on passed examples. The objective (goal) function here is a very simple one and it is set to ensure that for every data example (instance) x_i there is cluster centre *Cj* such that the distance between x_i and the cluster centre *Ccj* is less than a predefined threshold *Dthr* (Note that while Euclidean distance is used in this example, a range of measures exists e.g. Hamming, cosine. Other measures may be preferred where, for instance, the data at hand is not linearly separable.).

The resultant clusters can then be imported, analysed and validated by the expert using the ontological knowledge. This approach therefore utilises the best techniques of both semantic-based instance selection and quantitative selection available in the data mining workbench.

Where the process of instance selection is impeded by missing values in the data, imputation can be used to enhance the viability of as many vectors as possible, as follows.

Algorithm for missing values imputation:

- Assume that the value x_{im} is missing in a vector (sample) $S_m = (x_{1m}, x_{2m}, \ldots, x_{im}, \ldots x_{nm})$
- Find the closest *K* samples to sample S_m based on the distance measured with the use only of the available variables (x_i is not included) – set S_{mk}
- Substitute $x_{im} = \sum_{j=1,k} (1-d_j)x_{ij} / \sum_{j=1,k} (1-d_j)$, where d_j is the distance between sample S_m and sample S_j from the set S_{mk}
- For every new input vector *x*, find the closest *K* samples to build a model (the new vector *x* is a center of a cluster and find *K* closest members of this cluster)

Instance Export

The instance export task is concerned with the translation of the ontological model into a format used by one or more data mining workbenches. The ontology driven knowledge discovery methodology then supports the reverse transformation of an ontological model into a dataset by exporting the knowledge base into a database format (see Figure 4.4).

Figure 4-1 summarizes all tasks involved in the Instance Preparation phase. The next chapter describes the tools developed to support this task alongside the instance cleaning and instance selection tasks. Appendix B shows relevant screenshots for this phase when applied to the biomedical case study.

4.2.4. Modelling

In this phase, modelling techniques are selected, different test sets are formed and the models are applied. Typically, several techniques are applied to the same problem in order to validate the patterns found through different perspectives.

The modelling phase in the ODKD methodology is composed of three tasks: model selection, data set design and model building. These three tasks are closely related to the instance preparation phase.

Model Selection

Model Selection is driven by an understanding of the problem, its domain and the knowledge bases available for instance selection. In this task, various modelling techniques may be evaluated and selected as candidates for the data mining task. Normally, more than one technique is suitable for the same problem type. Some mining techniques have specific requirements for the form of the data. Therefore, stepping back to the “instance preparation phase” is often necessary.

Data Set Design

Data set design is related to the creation of training and test sets for the model building phase. This task interacts intensively with the instance export task with the latter generating different sets of data from the ontological model to be used in the modelling exercise. Usually more than one set of data is generated before the definition of a data pipeline is established for deployment of a solution.

Model Building

Model building or Onto-mining is generally recognised as ontology learning from instances in ODKD. It is the application of data mining techniques to extract knowledge from datasets represented as instances within the knowledge base.

Most research in onto-mining concentrates on learning from textual and semi-structured resources in the process of building an ontology. However, there is huge potential in the investigation of the relationships between concepts in an ontological model. Link analysis, social networks and graphs may play an important role as inference mechanisms. This reasoning may help mining techniques produce better results.

Ontologies can help mining techniques in the same way the application of mining techniques in a good data warehouse can lead to better results than compared with those applied to raw data. This gives another level of abstraction by connecting domain and problem knowledge in a shared model.

This research has come to the conclusion that both the ontology engineering and data mining fields have much to contribute to each other and a broad vision is needed to integrate them. The biomedical application described in chapter 6 presents the results of an integration of ontologies and a novel data mining technique which leverages both ontology engineering and KDD.

4.2.5. Evaluation

This phase is concerned with the evaluation and acquisition of knowledge extracted by the data mining model generated in the modelling phase. The knowledge acquired must be mapped and/or translated into the ontology and then evaluated before its final incorporation into the ontological model. This phase is divided into three tasks in the ODKD process: knowledge extraction, knowledge assessment and ontology enhancement.

Knowledge Extraction

This task is responsible for the incorporation of the knowledge extracted by data mining into the ontological model. It might also be considered an ‘automatic knowledge acquisition’ task. After the acquisition of the knowledge, a new cycle of analysis begins to assess the knowledge incorporated.

There are different techniques reported in the literature which may be used for the process of mapping knowledge from a data mining technique into an ontological model. Some of the research in this area includes merging and mapping techniques inherited from the investigation of ontology integration techniques, such as Formal Concept Analysis (FCA) (Ganter, Wille, & Stumme, 2005). This can also be undertaken according to similarities among graphs through the use of graph theories.

Although this research does not emphasise the difference between formal and informal ontology languages, it should be noted that the formal approaches can exploit the advantage of logic to enable automatic reasoning when new knowledge is acquired in the knowledge extraction task. Description logic (Nardi & Brachman, 2002), for example, is able to automatically classify new knowledge extracted through its reasoning mechanisms. However that knowledge must be first converted to the logic formalism in order to enable the reasoning to occur.

The semantic web research community is dealing with most aspects related to formal ontologies and reasoning. Active groups are establishing patterns that aim to address some of the critical points in this area. Expressing and reasoning with imprecise knowledge, for example, is an active area where non-classical logic plays a key role.

The following two chapters describe the techniques used in the prototype of the ODKD framework and in the biomedical application to deal with knowledge extraction, uncertainty and assessment.

Knowledge Assessment

At this stage of the ODKD process the user has built a model (or models) that appears to be of high quality from a data analysis perspective. Before proceeding to the final deployment of the model through the enhancement task, it is important that the team more thoroughly evaluate the created model from a conceptual point of view, to be certain it is also meaningful from a domain perspective. In the biomedical discovery field, for example, it is important to verify whether the knowledge extracted is biologically plausible or if there are evident facts that might suggest a review of the steps executed to construct the model, select the instances and so on.

A key objective is to determine whether there is some major problem that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached.

Uncertainty must also be considered during this assessment. Knowledge acquired at this stage might not be easily assessed due to lack of understanding of the problem/domain, lack of sufficient data and the like. The ODKD process then defines a “quarantine structure” where knowledge acquired from a modelling exercise can be stored for extra analysis in the ontology enhancement task without impacting on the main ontological model.

Depending on requirements, this task can be as simple as generating a report on the knowledge extracted by the modelling technique or as complex as moving the mining model into full production use. It might involve the analysis of the current ontological model to validate the knowledge acquired or even acquire more domain knowledge to support the findings and/or identify the need to go back to ontology analysis to perform extra analysis and select new test sets.

The biomedical application chapter demonstrates this cyclic process by integrating different sources of knowledge, applying a data mining algorithm, extracting knowledge, validating this knowledge using previous knowledge or triggering new mining exercises leading to the integration of different experiments.

Ontology enhancement

As stated previously, creation of the model and its assessment is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that the user can employ effectively. The ontology enhancement task is responsible for the final acceptance of the knowledge generated and where any “quarantine structure” is considered stable enough to be used in other live KDD tasks.

In one sense, this ontology enhancement task might be compared to the deployment phase of the CRISP-DM process. At this stage, the KDD team decides whether they agree with the knowledge assessed and the quality of the model, and therefore whether they should use the knowledge represented in the quarantine structure to update the ontological model. This decision might be made using difference inferences such as rules based on expert knowledge, heuristics, specific classification algorithms (for example a classification model which takes into consideration the quality of the annotations, provenance of creation, number and strength of evidence), or even graph based algorithms which may find high dense connected sub-graphs which may indicate a cluster or an important information element enabling an expert user to make an informed decision. For example, the pattern found in a modelling technique might indicate that some molecular functions are responsible for triggering a disease. After the learning task, this new molecular knowledge can then be used to support the decision of a medical team in the treatment of the disease.

In summary, this Evaluation phase (Figure 4-1) closes the ODKD cycle where knowledge is generated and made available for other experiments or KDD tasks.

This ODKD section has described the novel hybrid life cycle developed to integrate both ontology engineering and the knowledge discovery in databases processes.

4.3. Summary

The fact that today large amounts of data exist in many domains and that knowledge can be induced from these data using appropriate algorithms has been the driving force of the evolution of KDD systems. Ontology aims to facilitate the understanding, sharing, reuse and integration of knowledge, thereby helping to address many of the difficulties currently experienced in managing large distributed on-line information resources. People already look less and less at raw data, and, as the volumes accumulate, few, if any of us, will have the time or the mental capacity to assimilate the new data, structure them in a meaningful way, and extract information without first processing the data through an ontology or some other similar machine-based organisational aid.

Computer scientists have typically been focusing their attention on learning algorithms, which provide the core capability to generalise large numbers of specific facts into useful patterns and extract high-level rules. Although learning techniques play a central role in a knowledge engineering process and seem to hold the most excitement, in a real-world discovery task with several problem dimensions, such as in the biomedical discovery domain, it is clear that KDD can be extremely complex, and that low-level data mining is only one step in the knowledge engineering process.

In order to deal with real-world data mining tasks with high dimensionality there is a need to reduce the problem space search while increasing the domain knowledge. This chapter has presented a novel hybrid approach which combines the best practices of both ontology engineering and the KDD field.

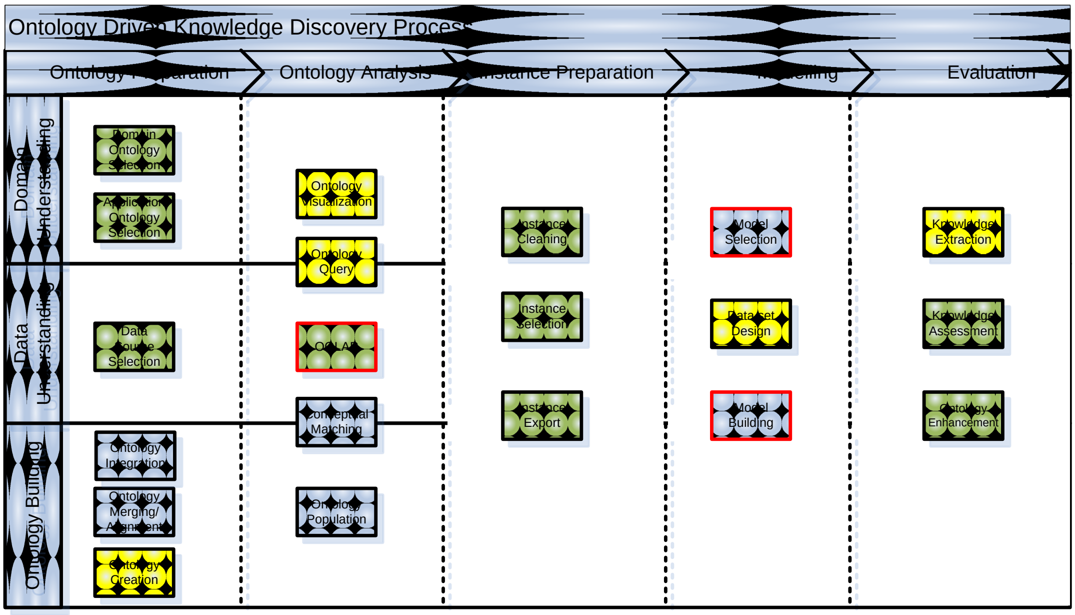
The Ontology Driven Knowledge Discovery Process is a conceptual model which, alongside the Evolving Ontology meta-knowledge model, is the basis for the development of a framework able to enhance the process of knowledge discovery from data by adding a high level of abstraction to the process, which allows better reuse of previous knowledge as well as the creation and evaluation of new knowledge.

The main outcome of this chapter is a hybrid knowledge discovery process which defines a life cycle based on the principles of the most widely adopted KDD process in industry as well as supporting the principles of evolving ontology which enable continual enhancement and acquisition of knowledge in a discovery exercise.

The process is divided into five phases which are composed of different tasks. The tasks have some methodological guidelines that indicate some of the best practices identified through experimentation within this research. These practices can be extended in accordance with the problem domain and KDD task. The process also presents guidelines based on both industry-specific published experiences and the research developed in this thesis. As a cyclic process it does not have a pre-defined or rigid order. On the contrary, the main objective is to create a reference life cycle that can be extended while keeping the most important tasks for the integration of ontology engineering and KDD. The outcomes of each phase can be used as milestones to ensure a consistent use of the process while enabling extension and flexibility of the life cycle use and application.

The guidelines must be used jointly with normal project management standards and practices, therefore the outcomes of each sub-task are considered live documents and therefore can and should be enhanced through the life-cycle of a project.

Table 4-2 – A colour coded Ontology Driven Knowledge Discovery phases and tasks.



As described early in this Chapter (section 4.2), the ODKD methodology is a novel hybrid model which integrates all steps necessary in the integration of ontology engineering and KDD. The table above (Table 4-2) is colour coded to more clearly identify the contributions of this research from an ontology engineering point of view. Some of the tasks are novel and have been included considering the KDD process, some have been adapted from the ontology engineering field and are used in the novel context of the ODKD, others have been adapted but maintain their original ontology engineering feature, and some are not part of the core research but must be included as part of the closed loop proposal. The green colour represents a novel element, the yellow colour represents a modification, the blue colour represents the adoption of well established tasks in a different context and tasks with red borders are not part of the core research.

This methodology and process model defines a minimal set of tasks and outcomes necessary to integrate ontology engineering and knowledge discovery in databases. The application of this life cycle in terms of concrete examples is demonstrated in Chapter 6, while the ontological framework and its tools are presented in the next chapter. Further illustration can be found in Appendix B where relevant screenshots are presented to demonstrate the supportive tools developed and utilized in this research for each phase of the ODKD methodology. The study’s conclusions and the discussion of future research present other methodological approaches also considered in the research.

Chapter 5

Ontology Driven Knowledge Discovery Implementation Framework

This chapter describes the Ontology Driven Knowledge Discovery Implementation Framework. The chapter begins by defining the specification criteria of the software framework followed by a description of the ontology environment and the set of plug-ins developed or adopted in this research to support the Ontology Driven Knowledge Discovery process (ODKD) and used in the development of the biomedical case study. The chapter finishes with a validation of the software specification criteria.

5.1. Introduction

The process of building a full domain ontology tends to be very complex. Even a very ad-hoc process may involve solutions using numerous ontologies, both existing and newly developed. These ontologies will then progress through a series of upgrades and extensions which must be carefully managed in order to keep track of the modifications as well as their sources (Denny, 2004).

A good tool to support domain ontology building should enable mapping, merging, comparison, and linkage between different ontologies as well as their conversion into several formats to form an ontological model (Gandon, 2006; Gómez-Perez, 1999; Gómez-Pérez, 2002). This model may then be utilized in different applications, such as knowledge discovery tasks where the acquisition, organization, and visualization of domain knowledge are necessary before a data mining exercise is undertaken. Further features are also needed to incorporate and evaluate the knowledge discovered in a data mining task.

This chapter presents the implementation framework adopted to support the *Ontology Driven Knowledge Discovery process* as well as a set of specific tools and meta-models developed in this thesis to support the *Evolving Ontology meta-knowledge model* and the *biomedical case study described* in the next chapter.

Although generic and applicable to different domains, the prototype framework was built primarily to support biomedical knowledge discovery processes and case studies developed in this research, therefore the plug-ins developed and/or extended in this research are mainly focused on the biomedical domain. On the other hand, the adopted plug-ins described in this chapter are just an example of plug-ins available and developed by the Protégé open source community. For example, the visualization plug-ins covered in this chapter are mainly those which were modified to fulfil the requirements elicited during the development of the case studies. Some extra visualization tools such as Jambalaya were reviewed and tested and while they are not part of the framework described here they can be viewed in and adopted from the Protégé community website (Protege, 2006a).

In a similar vein, rather than cover several different techniques and algorithms that may or may not be relevant to a given problem, the focus of this exercise is the implementation and selection of a set of candidate techniques suitable for the biomedical domain. The framework explores the extensible conceptual model and architecture of the selected ontology environment in order to cope with both generic and specific requirements.

The chapter begins by linking the framework with the multi-methodological research methodology adopted in this thesis. This is followed by the definition of both requirements and system development evaluation criteria. The chapter continues with the description of the Ontology Driven Knowledge Discovery Implementation Framework which is presented alongside some screenshot samples of the system used in the biomedical case study. Finally, the chapter is summarized and research outcomes are presented.

5.1.1. Research methodology

This chapter, along with the biomedical application in Chapter 6, comprise the practical contribution of this thesis (see Figure 0-2). It is part of the system development phase of the constructive approach followed in this work, which enables the building of a research framework by undertaking technical development and refining domain concepts, as discussed in Chapter 1.

This chapter covers the main stages of the **systems development** phase. It is concerned with theory testing, and brings about a realistic technological evaluation of the system developed and its potential for acceptance as described in Chapter 1.

Nunamaker et al (1991) emphasize that every researcher of IS engineering firstly needs to determine and limit the scope of his/her research. Such a step is important when integrating ontologies and data mining in a knowledge engineering process. On one side, the system must be generic and suitable for any domain. On the other hand, due to the characteristics of certain domains, such as the biomedical domain, and due to the applicability of various algorithms for specific modelling problems, the system must clearly define its scope while being extensible and flexible enough to incorporate several techniques and various modelling methods.

The design criteria described in the next sections are used to guide the development of the system architecture, the prototype, and the final product, they are: Scope, System Design, Product and Technology Transfer. These criteria both inform and are informed by the research scope definition and design specification stated previously in Section 1.4.2. They are based on ontology engineering requirements explained in Section 5.1, open source and Protege development best practices (Protege, 2006a; Knublauch, 2003; Noy, Fergerson & Musen,2000), and the ODKD requirements discussed in Chapter 4.

As stated before, these criteria reflect the aim to increase theoretical knowledge and improve practice through the development of a prototype which can relate both theoretical and practical contribution of the thesis (Adams & Courtney, 2004). For example, the prototype must cover all phases defined by the methodology and should follow open source development best practices relating to reuse, extensibility, loosely coupled modules, and so on.

As discussed previously, the criteria also collaborate to the reuse and extensibility features of the prototype while contributing to the wider Protégé open source community (Protégé 2006a).

Scope Criteria

These criteria are mainly based on the objectives of the research and research methodology. To meet these objectives, the framework should:

- Cover all phases of the Ontology Driven Knowledge Discovery process in order to test the proposed framework and methodology as covered in the research methodology in section 1.3 - Experimentation;

- Focus on the ontology engineering problems rather than on the specifics of data mining techniques since the ontological framework is the focus of this thesis;

- Be generic as an ontology engineering tool but specific enough to cover the biomedical knowledge discovery requirements in order to be reused in different contexts such as those explored in the context of teaching.

System Design Criteria

These criteria are derived from the ODKD methodology requirements (Chapter 4), the research objectives (Chapter 1) and software development best practices developed within the open source and ontology engineering community (Gandon, 2006; Gómez-Perez, 1999; A. Gómez-Pérez, Fernández-López, & Corcho, 2004). To meet these requirements, objectives and practices, the system should:

- Have an extensible and flexible architecture;

- Be able to integrate different workbenches (analysis tools);

- Have a search language based on queries;

- Have an integrated ontological navigation mechanism;

- Support different visualization techniques.

Product Criteria

The prototype and final product should:

- Be based on a widely accepted ontology environment;
- Be representation formalism independent;
- Reuse software components and meta-data definitions;
- Be fully testable and usable.

Technology Transfer Criteria

These criteria were developed to meet research and ontology engineering requirements as well expectations that the work should clearly reside in the public domain. The system should then:

- Be published as an open source project;
- Emphasize new ODKD functionalities;
- Be a suitable ontology engineering platform for data mining research and projects;
- Leverage specific ontology engineering research;

- Be able to be published in modules in order to be applied to different domains.

5.2. ODKD Implementation Framework

The Ontology Driven Knowledge Discovery Implementation Framework is a set of ontology engineering tools developed, extended and/or adopted to support the evolving ontology hybrid life cycle for knowledge discovery tasks. The ontology driven environment is composed of an Ontology Editor tool, a set of plug-ins, and meta-models, as per the sample depicted in Figure 5-1.

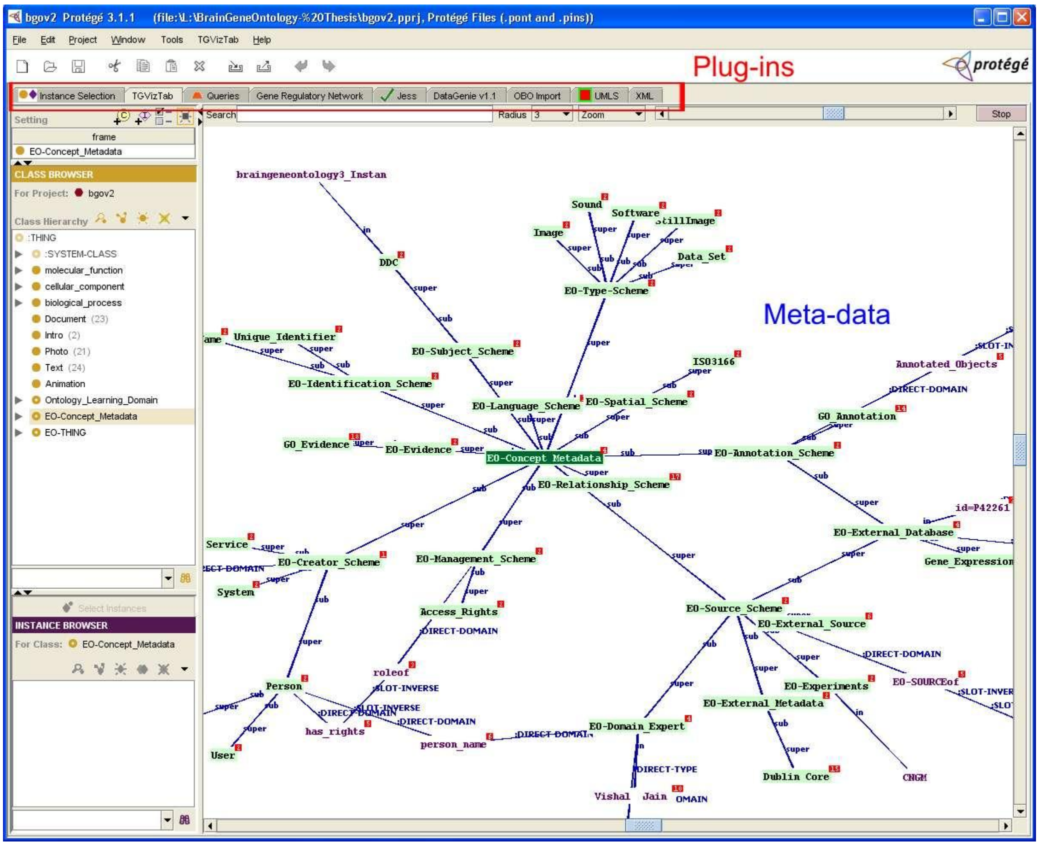


Figure 5-1 – ODKD Framework set of sample tools

The tabs bar highlighted in the figure above shows some of the plug-ins developed for and utilized in the biomedical case study. The central panel shows a visualization of the novel metadata model developed using a modified TGViz plug-in, while the screenshot overall represents an example of the framework environment user experience.

This chapter concentrates mainly on the ontology engineering tools constructed during this research. The underlying meta-models, the meta-classes and the evolving ontology meta-data have been described in Chapter 3.

Although tailored for the biomedical case studies developed in this thesis, the framework is extensible and generic enough to be applied to other domains and applications, such as an educational application based on the concept of learning objects (Kasabov, Jain, Gottgtroy, Benuskova, & Joseph, 2007) and the development of an application in the context of biblio-mining (Parikshit, Pears, & Gottgtroy, 2006).

5.2.1. System Architecture

The system architecture is based on the modular approach embedded in the Protégé environment and is composed of four layers: Evolving Ontology Layer, Access Layer, ODKD Layer and Application Layer (see Figure 5-2).

The Evolving Ontology layer is composed of the meta-knowledge model and the knowledge base accessible through Protégé’s application program interface (API).

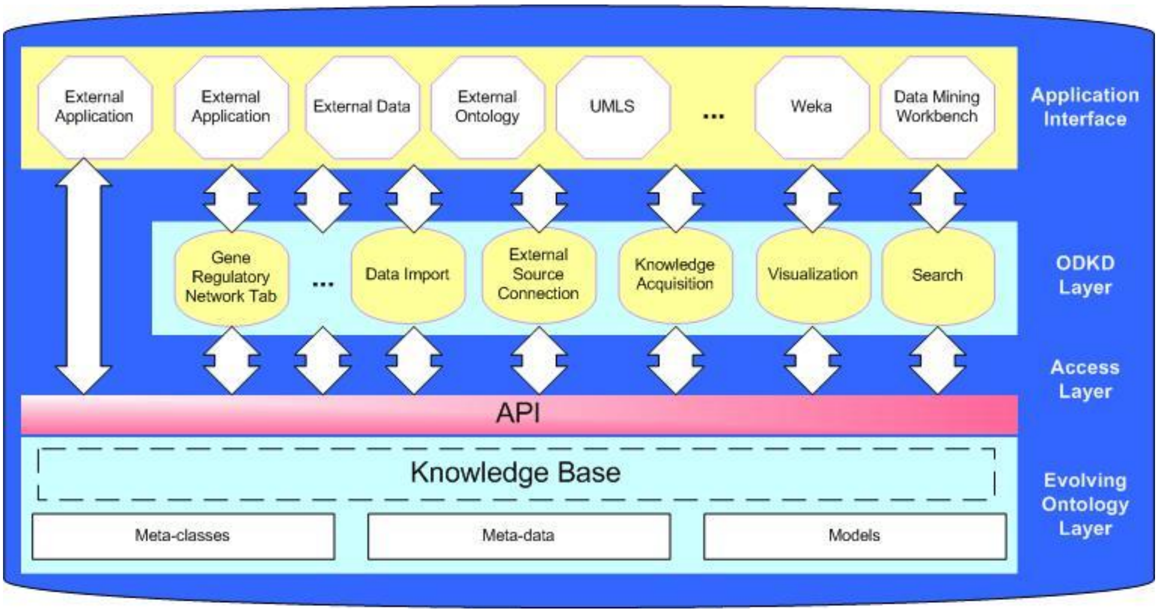


Figure 5-2 – ODKD Framework System Architecture.

Protégé's API is responsible for exposing core functionalities related to the manipulation of the knowledge base and meta-models. It also enables the extension of the environment by exposing a common interface for rapid prototyping and application development. The API is directly accessed by external applications as well as being indirectly accessed through the ODKD layer.

The ODKD layer is composed of a set of tools, developed as plug-ins, which are responsible for several functionalities such as external connection to applications, knowledge acquisition, and visualisation and so on.

The ODKD layer is the ontology driven bus which accesses both the application layer and the API in order to provide the functionalities required by the Ontology Driven Knowledge Discovery process.

The Application Layer is the external connection to the ODKD layer. It is composed of external ontology sources, such as ontology repositories and domain ontologies, as well as specific data mining algorithms and KDD tools and workbenches, such as Weka (Witten & Frank, 2005) and Neucom (KEDRI, 2006b).

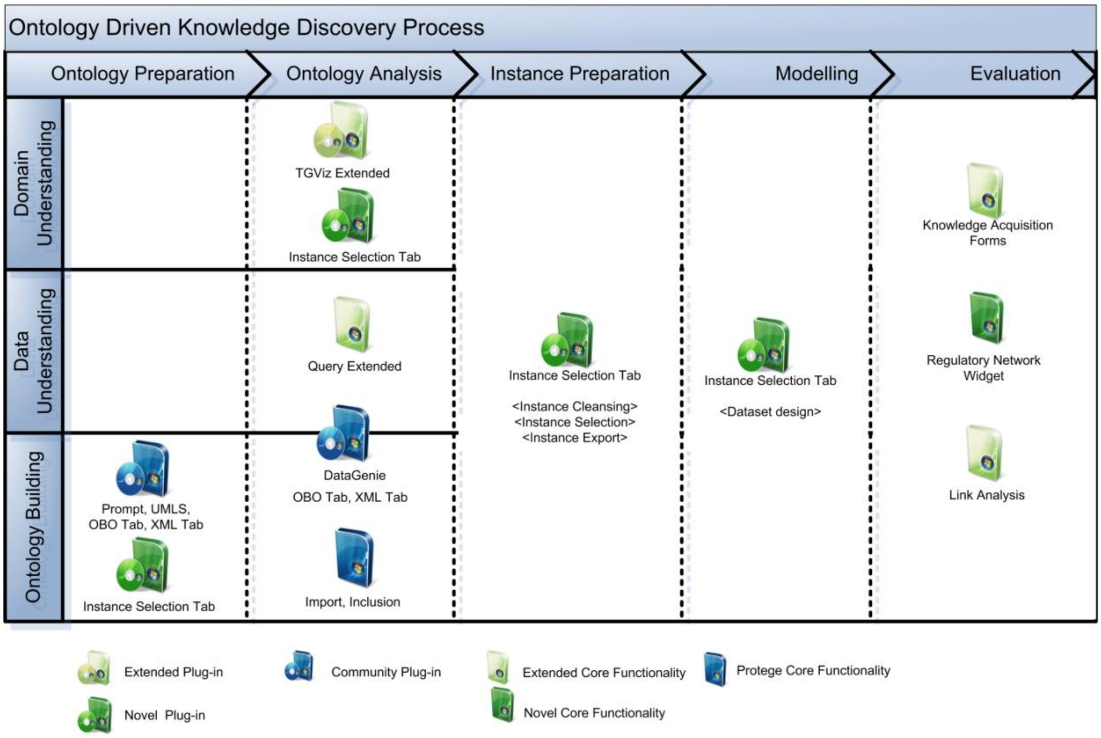
The Evolving Ontology layer was fully developed in this research and uses Protege's API and system architecture to interact with the other layers. The ODKD layer represented in the figure also shows plug-ins fully developed in this research such as the Gene Regulatory Network Tab and Knowledge Acquisition forms, plug-ins modified in different levels to fulfil the requirements defined for the framework such as those supporting visualization and search, and plug-ins directly adopted such as those for data import which were mainly selected from Protege's toolset[8]. The Application Interface layer also incorporates examples of bespoke developed features such as the Weka export feature developed as part of the Instance Selection tool and adopted tools such as the UMLS tab used to import medical data.

The framework supports all five steps of the Ontology Driven Knowledge Discovery process: Ontology Preparation, Ontology Analysis, Instance Preparation, Modelling, and Evaluation (Table 5-1). As stated previously, the main focuses of this

[8] As described before, the Protégé community is in constant growth therefore new plug-ins may become available to fulfil any other identified domain requirements such as the need for a new task, a new visualization tool, or even a new data source importing tool.

research are the ontology related steps; therefore specific modelling techniques are not covered in this chapter. The next chapter does, however, describe the use of a data mining technique in the brain-gene biomedical application.

Table 5-1 – ODKD implementation framework.



Ontology Driven Knowledge Discovery Process					
	Ontology Preparation	Ontology Analysis	Instance Preparation	Modelling	Evaluation
Domain Understanding		TGViz Extended Instance Selection Tab	Instance Selection Tab <Instance Cleansing> <Instance Selection> <Instance Export>	Instance Selection Tab <Dataset design>	Knowledge Acquisition Forms
Data Understanding		Query Extended			Regulatory Network Widget
Ontology Building	Prompt, UMLS, OBO Tab, XML Tab Instance Selection Tab	DataGenie OBO Tab, XML Tab Import, Inclusion			Link Analysis

Extended Plug-in
Novel Plug-in
Community Plug-in
Extended Core Functionality
Novel Core Functionality
Protege Core Functionality

The framework is composed of newly developed plug-ins, extended plug-ins, and adopted plug-ins. Some plug-ins support specific tasks such as ontology query but others support more than one task, or even a whole ODKD phase such as the Instance Preparation phase. The table above uses colour coding to clearly identify the contributions of this research. Coloured icons have been used to represent the novelty and level of functionality - the green colour represents a novel plug-in, the light green represents an extended plug-in while the blue colour represents the adoption of existing features. The icons represent the level of modification being those applied to the core Protégé feature set or developed as plug-ins which must be installed.

While a variety of plug-ins were built, tested and evaluated throughout this research, this chapter largely describes those finally adopted to cope with functionalities required by a biomedical knowledge discovery process as well as those able to fully support the cyclical Ontology Driven Knowledge Discovery process. To this end, the next sections are organized into the phases of the ODKD process.

For ease of understanding, the ODKD process from Chapter 4 has been re-shown as Table 5-1 and brief descriptions of the phases are represented again in each of the following subsections. A full description of the process can be found in Chapter 4. A few of the illustrative screenshots presented in this chapter are based on the newspaper organization sample provided with Protégé. However, most of the screenshots are based on the biomedical domain in order to gradually introduce aspects of the biomedical application described in Chapter 6.

5.2.2. Ontology Preparation

The Ontology preparation phase is composed of three main pipelines: domain understanding, data understanding, and ontology building. The first two, domain and data understanding, are related to problem understanding which is achieved by the investigation and selection of candidate ontologies for a problem domain. This candidate ontology selection process is guided by ODKD methodology, as described in the previous chapter.

The ontology building task is the creation, inclusion and/or import of ontological models as well as the creation of specific concepts not covered by the previously selected ontologies. The building process involves three main sub-tasks: Ontology Integration, Ontology Merging/Alignment and Ontology Creation.

Ontology Integration

Ontology integration is the incorporation of ontologies identified in the domain understanding tasks. There are two mechanisms available in the Protege environment that incorporate and reuse ontologies in the framework: ontology import/inclusion and ontology annotation. The former is the incorporation of a complete ontology and the latter is the incorporation of some specific concepts of a selected ontology source.

Import/Inclusion

Import/Inclusion tools are usually specific plug-ins developed to import data from ontology storage, such as Jena persistent storage. There are several import plug-ins available in Protégé (Protege, 2006b).

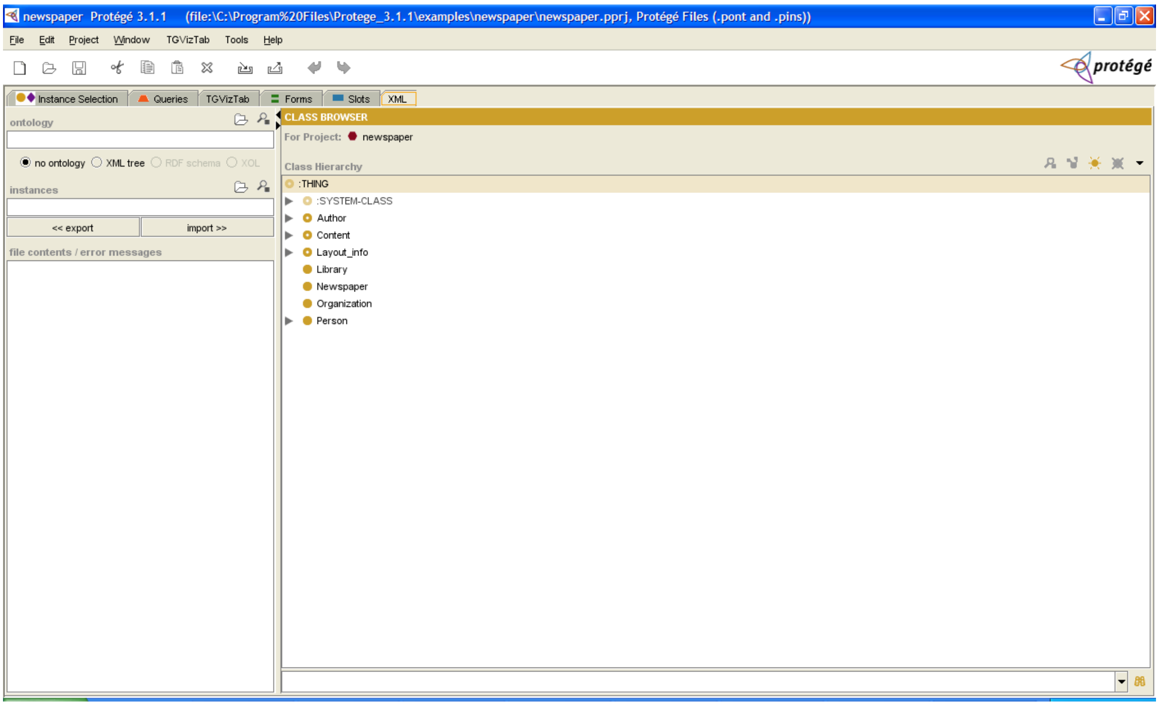


Figure 5-3 – XML Tab.

Several import plug-ins, such as the XML Tab (Figure 5-3) (Sintek, 2006), were tested and/or used in the development of the thesis' experiments.

In the biomedical domain, different standardization efforts, such as the Open Biomedical Ontologies (OBO), have also facilitated the process of import/inclusion of ontologies. This feature represents well the strategy followed in this research when developing new tools or adopting external ones. The general adoption of the OBO standard at the late stage of the thesis and its adoption by biomedical ontologies such as the Gene Ontology (GO) (GO, 2006), for example, led to replacement of the GO-specific import program code developed in the early stage of this study (Gottgtroy, Kasabov, & MacDonell, 2003a) with the OBO tab (Silberfein & Gennari, 2006).

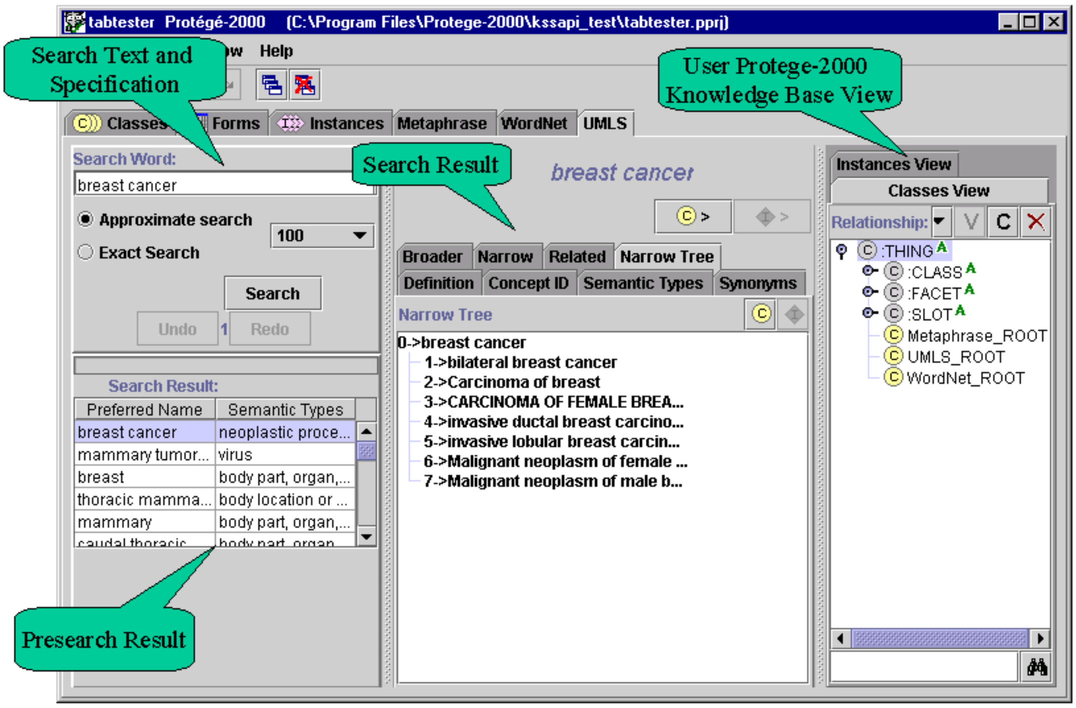
The strategy of adoption, reuse or development of new tools and features was therefore based on availability, requirement fulfilment and standardization. In the import scenario as described above our early developments were substituted by several generic plug-ins based on the creation and maturity of open standards such as XML and OBO.

Ontology Annotation

Ontology annotation is the process of annotating concepts from large domain ontologies, such as WordNet, into an application ontology. Annotation is required when only a part of an ontology is necessary or when a domain ontology has a specific representation formalism and its importance justifies the development of a specific tool to translate the source formalism into the target formalism.

Annotation is generally achieved in Protégé by the development of specific tools able to connect to well-established ontologies and copy certain sets of concepts into the model being developed. Instead of merging or importing a complete ontology, these tools allow the navigation, selection and import of specific concepts from a domain ontology. The UMLS tab (UMLS, 2006), for example, is used in this research to connect to the Unified Medical Language System and import biomedical data into Protégé (see Figure 5-4).

Along with the use of this externally developed tool, templates and forms were developed in this research to manually annotate the knowledge base, such as for specific data about gene expression in areas of the brain as utilized in the biomedical case study.



Search Word: breast cancer

● Approximate search 100

○ Exact Search

breast cancer

Preferred Name	Semantic Types
breast cancer	neoplastic proce...
mammary tumor...	virus
breast	body part, organ,...
thoracic mamma...	body location or ...
mammary	body part, organ,...
caudal thoracic	body part, organ

```
0->breast cancer
   1->bilateral breast cancer
   2->Carcinoma of breast
   3->CARCINOMA OF FEMALE BREA...
   4->invasive ductal breast carcino...
   5->invasive lobular breast carcin...
   6->Malignant neoplasm of female ...
   7->Malignant neoplasm of male b...
```

```
:THING
   :CLASS
   :FACET
   :SLOT
   Metaphrase_ROOT
   UMLS_ROOT
   WordNet_ROOT
```

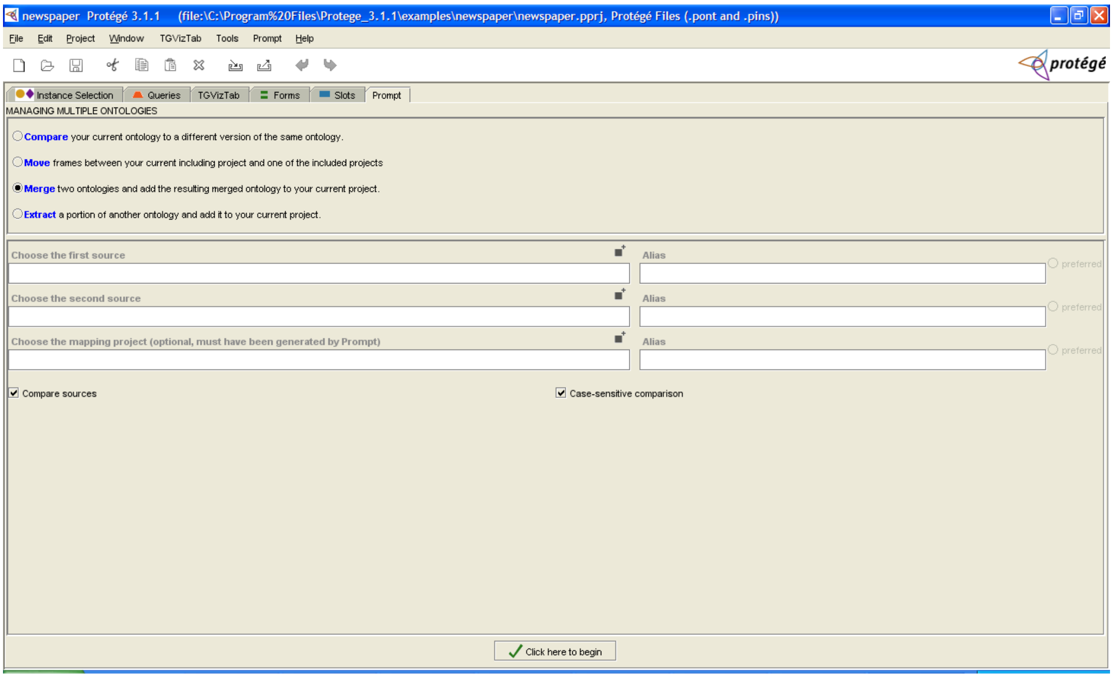
Figure 5-4 – Unified Medical Language System - UMLS Tab

The figure above shows the UMLS tab plug-in widely used by the Protégé community and adopted in this research to annotate medical knowledge.

Ontology Merging/Alignment

The merge feature adopted in this research is made available by the Prompt plug-in (Natalya & Mark, 2003). This plug-in, developed at Stanford University, has been referenced as one of the main merge techniques currently available in the ontology engineering field. The suite provides users with a uniform framework for comparing, aligning, and merging ontologies.

The plug-in, depicted in Figure 5-5, enables the management of multiple ontologies by the ontology engineer. It can enable the user to compare versions of the same ontology, map one ontology to another, move frames between included and including projects, merge two ontologies into one, and extract part of an ontology.



newspaper Protégé 3.1.1 (file:\C:\Program%20Files\Protege_3.1.1\examples\newspaper\newspaper.pprj, Protégé Files (.pont and .pins))
File
Edit
Project
Window
TGVizTab
Tools
Prompt
Help
protégé
Instance Selection
Queries
TGVizTab
Forms
Slots
Prompt

MANAGING MULTIPLE ONTOLOGIES

○ **Compare** your current ontology to a different version of the same ontology.

○ **Move** frames between your current including project and one of the included projects

● **Merge** two ontologies and add the resulting merged ontology to your current project.

○ **Extract** a portion of another ontology and add it to your current project.

Choose the first source | Alias | ○ preferred

Choose the second source | Alias | ○ preferred

Choose the mapping project (optional, must have been generated by Prompt) | Alias | ○ preferred

☑ Compare sources ☑ Case-sensitive comparison

Figure 5-5 – Prompt Plug-in.

Following the Protégé environment software architecture, the last Prompt version is based on a plug-in architecture which enables the development of specific mapping algorithms for initial comparison of ontologies as well as for the development of user interface components.

Prompt is also aligned with the ODKD proposal as it combines suggestions made by similarity algorithms with human knowledge to reach a decision about the inclusion of some ontological knowledge delivering then a hybrid and partial automated system. Prompt's general algorithm can be represented as in the following figure.

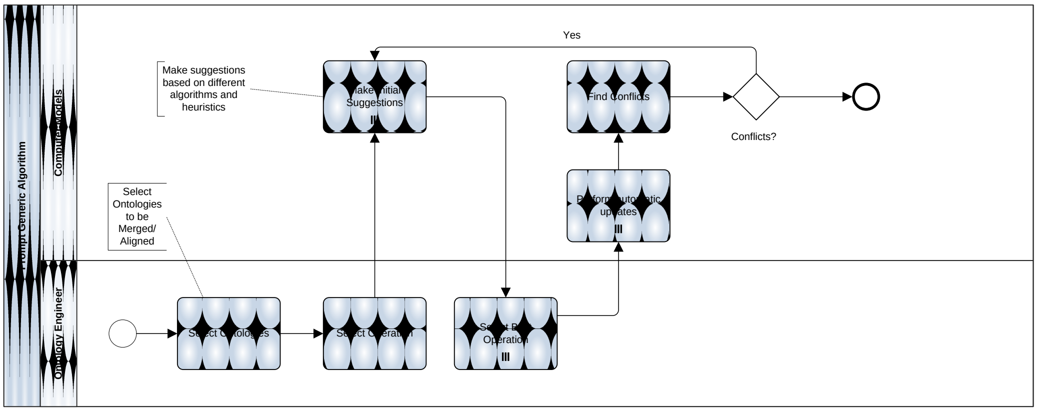


Figure 5-6 – Prompt Plug-in generic algorithm.

Although there was no need to extend Prompt in the biomedical case study, this new extensible architecture indicates that Prompt may be adopted as core functionality for the ontology driven knowledge discovery process in any domain. It can also act as an alternative user interface for the integration of learning algorithms based on graph algorithms in the ontology preparation phase.

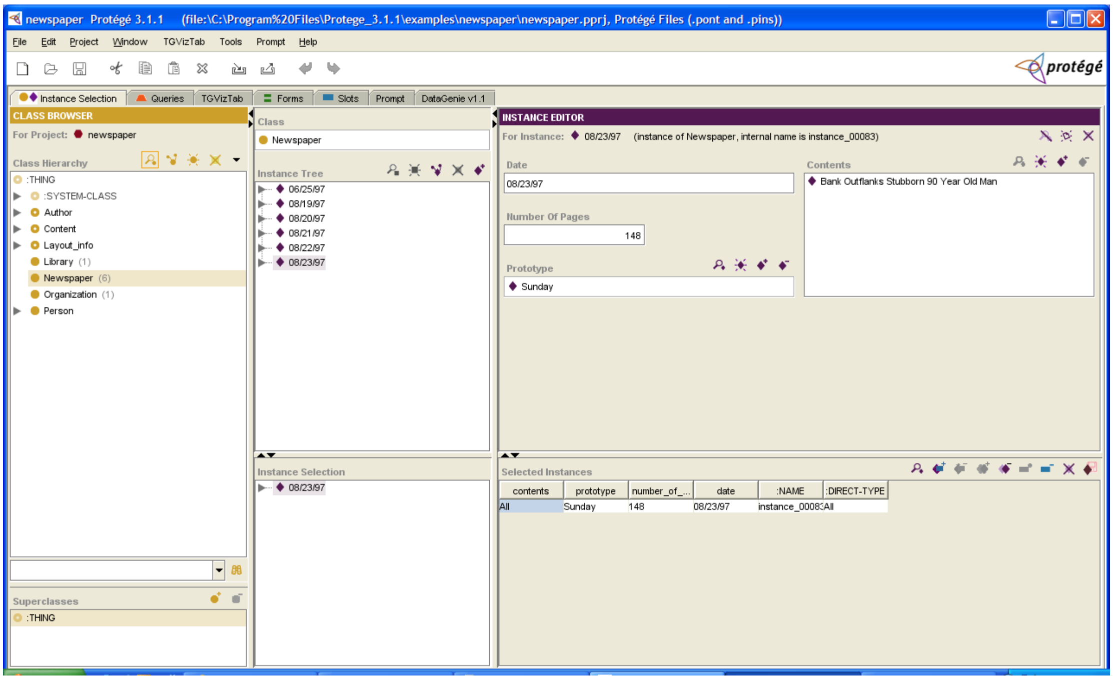
Ontology Creation

Protégé was primarily conceived as an ontology editor. Its flexible knowledge model enables the creation of an ontology in different formalisms, such as the ontology web language (OWL), as well as allowing the exportation of the ontology in different formats, such as XML and RDF. The ODKD Framework utilizes this robust knowledge model and the ontology editor features as a leverage factor for the integration of ontologies and KDD.

The novel **Instance Selection Tab** was developed in this research to provide support for the main editing functionalities related to the creation and modification of classes and instances (see Figure 5-7). The tab also extends the core Protégé editing features to cope with extra ODKD requirements as described later in this chapter.

The Instance Selection Tab integrates two core Protégé tabs: Classes and Instance tabs. It also allows some slot editing that is utilised in later phases of ODKD, such as in ontology export. However, due to specific ontology creation requirements, the methodology also adopts Protégé's *Slot* and *Form* Tabs in the ontology preparation phase. These tabs are intended to be used mainly by ontology engineers at the beginning of the building process to design forms and create special slots and are standalone basic tabs utilized in Protege to interact with the knowledge base and to configure the knowledge acquisition forms.

There is a second group of users, those who use the system to search, visualise and annotate the ontology among other tasks and who generally do not configure the environment. These users then do not use the basic Slot and Form tabs - they interact only with the instance selection tab, which will be explained later in this chapter.



For Instance: 08/23/97 (instance of Newspaper, internal name is instance_00083)

Date: 08/23/97

Number Of Pages: 148

Prototype: Sunday

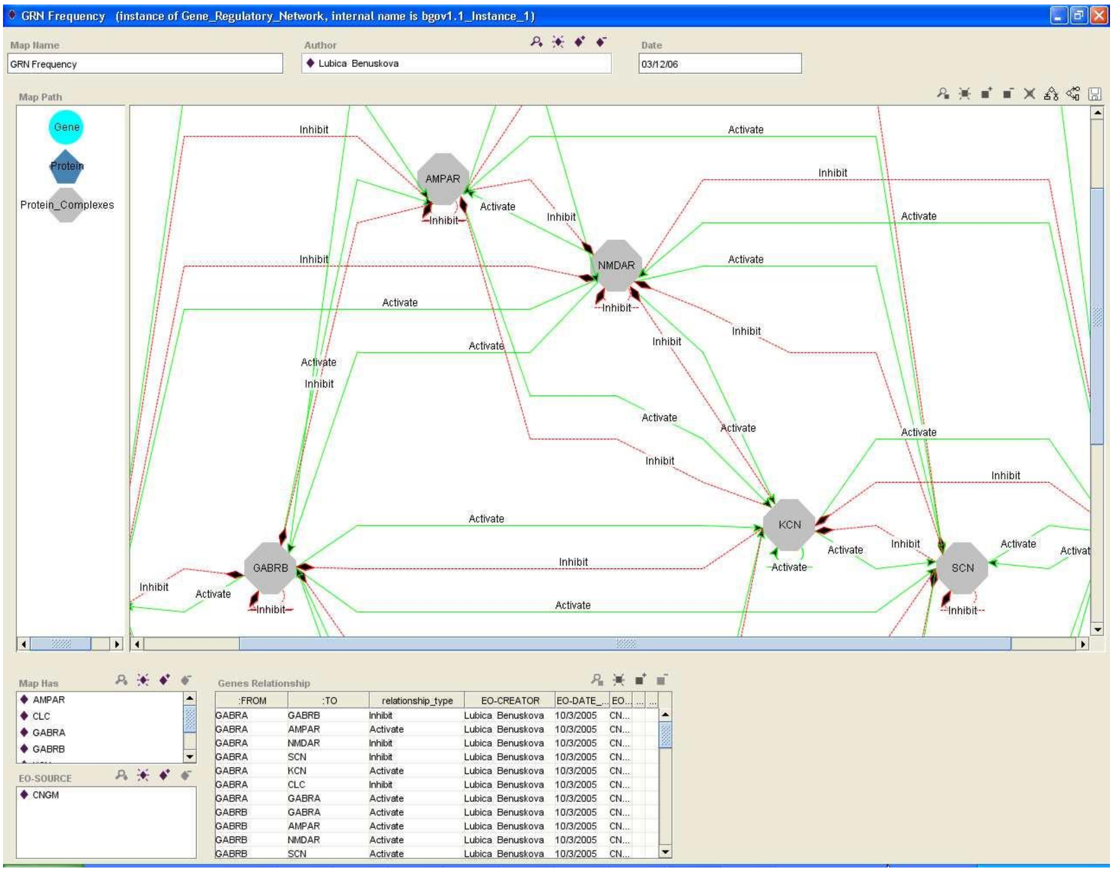
Contents: Bank Outflanks Stubborn 90 Year Old Man

contents	prototype	number_of_...	date	:NAME	:DIRECT-TYPE
All	Sunday	148	08/23/97	instance_0008:	All

Figure 5-7 – Instance Selection Tab.

The **Slot tab,** referenced in the paragraph above, is used by the ontology engineer to edit specific properties of the knowledge base such as configuration of sub-slots that bring a greater flexibility when modelling complex relationships and properties such as relationship-type, as shown in the example in Figure 5-8.

The Form, depicted in Figure 5-8, for example, was developed to capture gene regulatory network information. Forms are user interfaces which enable, among other things, the definition of the information design as well as the configuration of the slot widgets such as the central panel widget developed in this research to represent and automatically extract knowledge from the simulations described in the biomedical application in the Chapter 6. The form design feature is used in the development of knowledge acquisition tools as well as for the embedding of extra features in ontology driven applications, such as the ODKD framework link analysis widget.



GRN Frequency (instance of Gene_Regulatory_Network, internal name is bgov1.1_Instance_1)

Map Name: GRN Frequency

Author: Lubica Benuskova

Date: 03/12/06

Map Path

Map Has
- AMPAR
- CLC
- GABRA
- GABRB

EO-SOURCE
- CNGM

Genes Relationship

:FROM	:TO	relationship_type	EO-CREATOR	EO-DATE_...	EO...	...	...
GABRA	GABRB	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	AMPAR	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRA	NMDAR	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	SCN	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	KCN	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRA	CLC	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	GABRA	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	GABRA	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	AMPAR	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	NMDAR	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	SCN	Activate	Lubica Benuskova	10/3/2005	CN...		

Figure 5-8 – A sample biomedical user interface to represent gene regulatory information captured from simulations developed in this research.

5.2.3. Ontology Analysis

Ontology Analysis is the next phase in the ODKD process. It is related to the discovery of first insights into the ontological model as well as to the investigation and/or checking of initial hypotheses based on the disclosure of information hidden in the developed model.

The ODKD process defines four tasks in this phase: Ontology Visualization, Ontology Query, Conceptual Matching and Ontology Population. The first two tasks are the exploration of the ontological model by means of visualisation and searching.

The latter two tasks are the construction of the knowledge base through the insertion of data from the available databases and the population of instances from the selected ontologies.

Ontology Visualization

Ontology visualisation is the analysis of knowledge representation by means of visualisation techniques. The goal of this task is to facilitate exploration of different perspectives and navigation in a complex knowledge network.

The selection of the most appropriate visualisation techniques is highly dependent on the problem’s characteristics and on the ontological model. Therefore, the ODKD methodology suggests the adoption of at least two techniques in a KDD process: tree navigation and network visualisation.

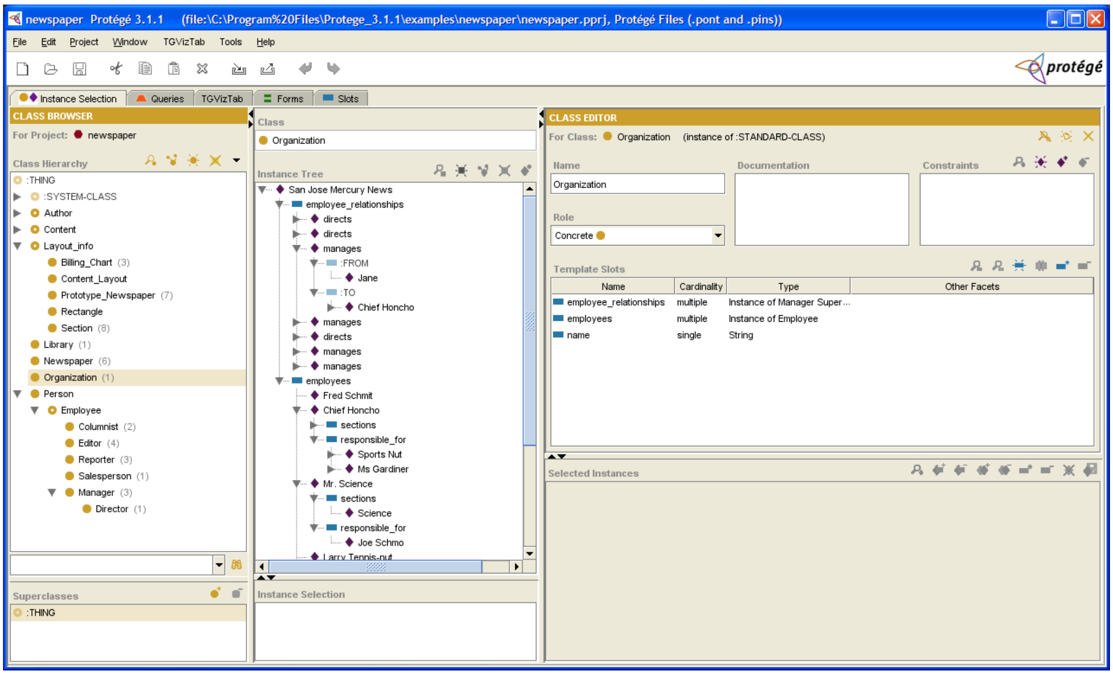
Tree Navigation

The multiplicity of tabs responsible for the navigation of classes, instances and slots present in Protégé were substituted with a novel tree navigation interface in the ODKD framework.

Tree navigation is mainly used in taxonomies to display hierarchical relationships and also generally used for classification representation. The Protégé user experience is mainly based on trees since most of the data is presented as a taxonomy of classes and the relationships are represented in slots which are also displayed in tree form.

Even though there are other ways of representing ontologies, such as graphs, and others also covered in this chapter, the tree has become pervasive in basic user interfaces, such as to represent folder hierarchy, or web content. It has also been used by many open sources ontologies such as Gene Ontology, UMLS and others. This research leverages this metaphor by simplifying and integrating the basic instance manipulation using trees and tables.

The developed Instance Selection tab (Figure 5-9) provides an interface that facilitates the navigation of a complex tree representation of classes, instances and relationships. It also supports the selection of any instance through tree navigation, which is used in the selection task of the instance preparation phase. These features then leverage the user experience already established while facilitating interaction by adopting an integrated view of instances, classes and slots.



Name: Organization

Role: Concrete

Name	Cardinality	Type	Other Facets
employee_relationships	multiple	Instance of Manager Super...	
employees	multiple	Instance of Employee	
name	single	String	

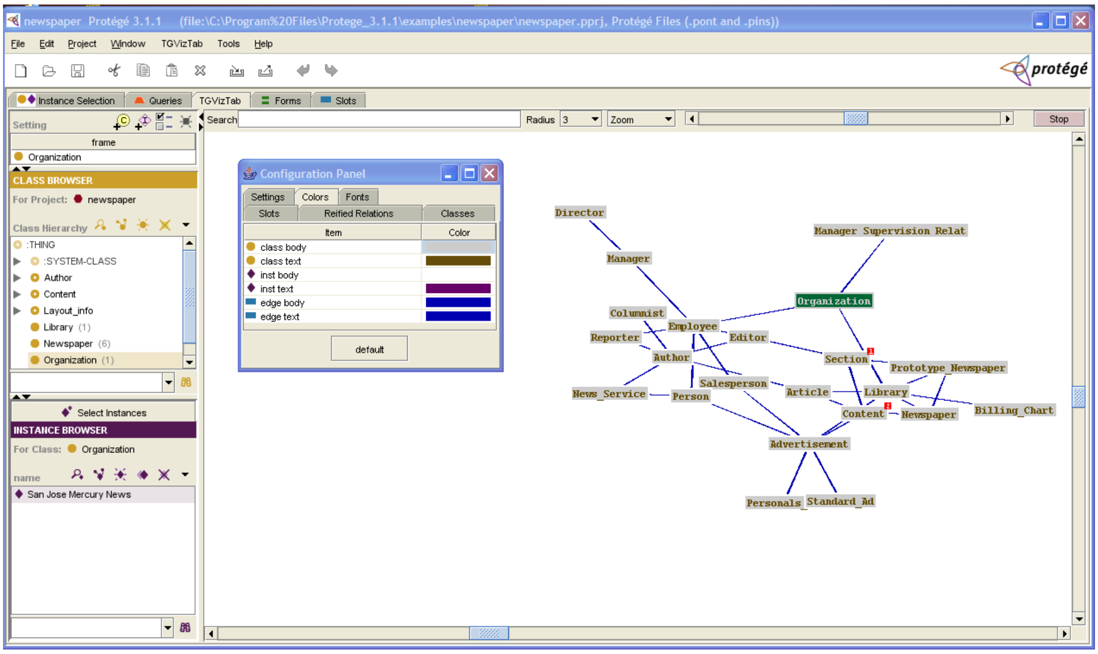
Figure 5-9 – A sample of both class, instance, and slot tree navigation in the Instance Selection Tab.

Network Visualization

The ODKD framework also extended the TGViz tab (Alani, 2006) (shown in Figure 5-10) as its main network visualization tool. The TGViz tab enables ontology visualisation through use of the TouchGraph java library (TouchGraph, 2006) which renders networks into interactive graphs.

This tab has different features such as search, zoom, and radius adjustment among others. It also has a set of properties that define which information should be displayed and how it should be displayed. These characteristics are very important to enabling the analysis of different aspects of a knowledge base; for example, it may set the visualisation to only instances and their relationships.

The radius function determines the number of levels to be displayed, for example in one type of analysis it might be important to drill down up to five levels of detail in order to have a more detailed view of a problem while in other circumstances a two level view may give a less granular and more aggregated view.



newspaper Protégé 3.1.1 (file:\C:\Program%20Files\Protege_3.1.1\examples\newspaper\newspaper.pprj, Protégé Files (.pont and .pins))

Radius 3

Item	Color
class body	▨
class text	■
inst body	
inst text	■
edge body	■
edge text	■

Figure 5-10 – A network visualization sample using the TGVIZ tab showing some of the visualization options available in this tool.

There are other visualization techniques available in Protégé which may be suitable for specific tasks and application domains (Protege, 2006b). This research has explored several of these techniques available in both the Protégé environment and in the literature. As stated in Chapter 4, this investigation is reported in (Wang & Gottgtroy, 2006) as a review of the most suitable techniques for biomedical informatics analysis.

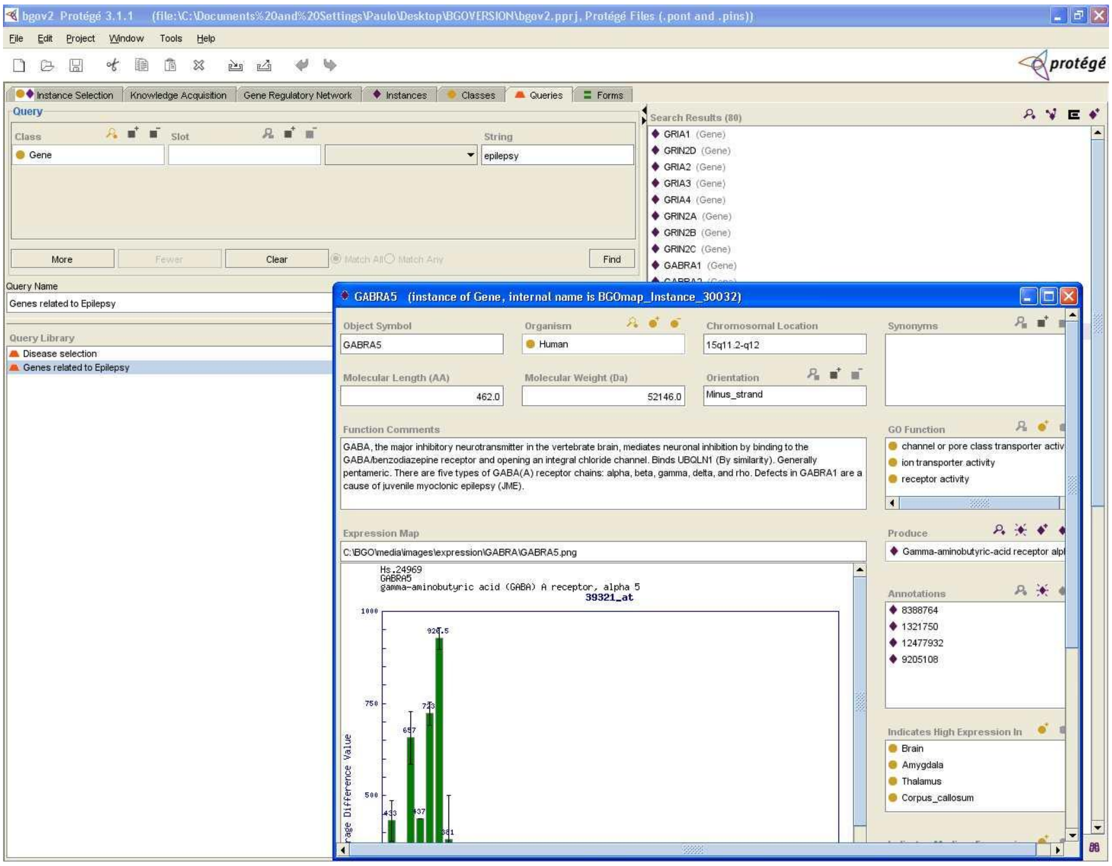
The ODKD framework makes extensive use of the features already developed in the tool; however this research has developed core functionality which allows the selection of the instances displayed in the graph for further investigation in the instance selection tab.

It is important to note that this research has as a fundamental principle the development of tools and features which leverage the well established user experience in both Protege's community and the biomedical ontology community in general. All plug-ins developed or extended and also evaluated for adoption considered the user experience as a main decision factor. In the case of graph visualization, one of the main weaknesses identified in existing components was the lack of link analysis features which were then developed and included in the gene regulatory network visualisation plug in.

Ontology Query

In spite of the advantages of exploring a complex knowledge structure by means of visualization, a query mechanism is still required when navigating domains where concepts are unfamiliar or the taxonomy varies in accordance with the fields being investigated.

The ODKD framework has adopted the search feature available in Protégé-Frames and has extended the basic functionalities by introducing new features and concepts in order to enable complex searches of an ontological representation as well as instance selection and export (see Figure 5-11).



Query

Class: Gene
Slot:
String: epilepsy

○ Match All ○ Match Any

Query Name: Genes related to Epilepsy

Query Library
- Disease selection
- Genes related to Epilepsy

Search Results (88)
- GRIA1 (Gene)
- GRIN2D (Gene)
- GRIA2 (Gene)
- GRIA3 (Gene)
- GRIA4 (Gene)
- GRIN2A (Gene)
- GRIN2B (Gene)
- GRIN2C (Gene)
- GABRA1 (Gene)

GABRA5 (instance of Gene, internal name is BGOmap_Instance_30032)

Object Symbol: GABRA5
Organism: Human
Chromosomal Location: 15q11.2-q12
Synonyms:
Molecular Length (AA): 462.0
Molecular Weight (Da): 52146.0
Orientation: Minus_strand

Function Comments: GABA, the major inhibitory neurotransmitter in the vertebrate brain, mediates neuronal inhibition by binding to the GABA/benzodiazepine receptor and opening an integral chloride channel. Binds UBQLN1 (By similarity). Generally pentameric. There are five types of GABA(A) receptor chains: alpha, beta, gamma, delta, and rho. Defects in GABRA1 are a cause of juvenile myoclonic epilepsy (JME).

GO Function:
- channel or pore class transporter activ
- ion transporter activity
- receptor activity

Expression Map: C:\BGO\media\images\expression\GABRA\GABRA5.png

Hs.24969
GABRA5
gamma-aminobutyric acid (GABA) A receptor, alpha 5
39321_at

Produce:
- Gamma-aminobutyric-acid receptor alpl

Annotations:
- 8388764
- 1321750
- 12477932
- 9205108

Indicates High Expression In:
- Brain
- Amygdala
- Thalamus
- Corpus_callosum

Figure 5-11 – A semantic query to find genes related to epilepsy and the visualisation of information of the gene selected.

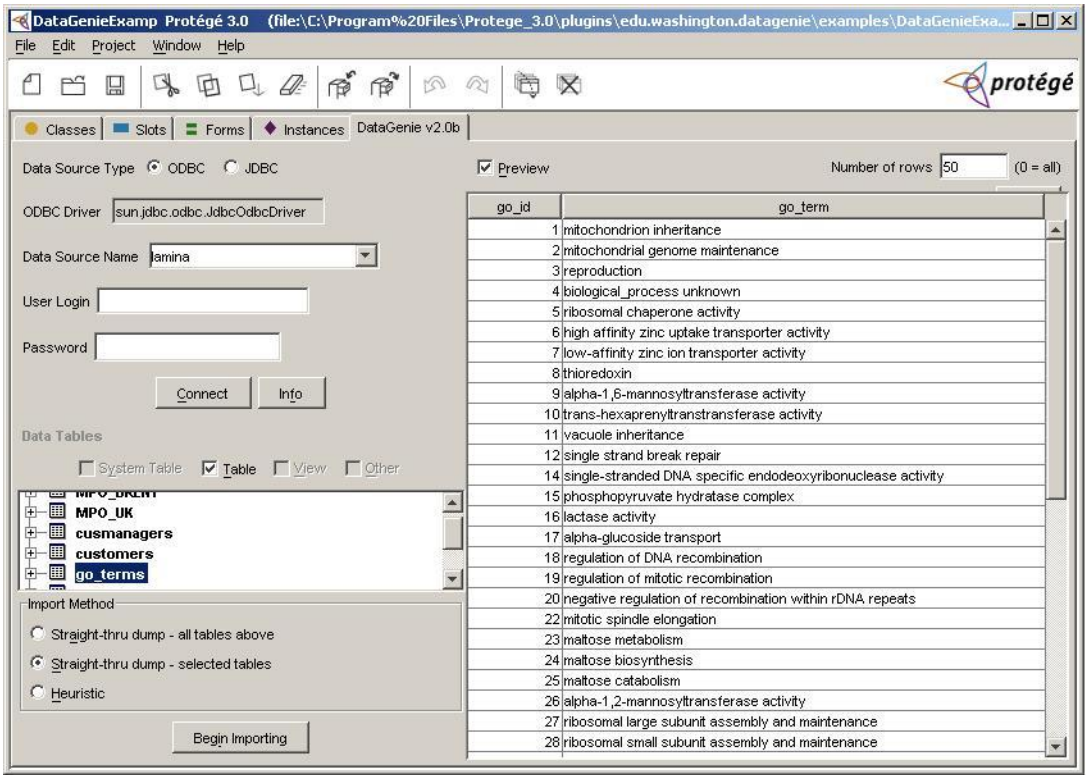
Figure 5-11 shows a generic query in which the user is researching all genes related to a particular disease. The left top panel describe the class of interest (Gene) while the slot, which defines the relationship, is empty which means that the user wants any instance that has any type of relationship with the string ‘epilepsy’. The result, in the left panel, is then a set of genes which has any relationship with the string ‘epilepsy’. The form on top of the search window then shows all information about the gene GABRA5 selected by the user which contains information about the gene expression and several parts of the brain as well as several annotations. The biological content was developed and acquired in the context of the biomedical case study. This mechanism enables the identification and validation of different biomedical hypotheses through the use and creation of semantic queries in a knowledge base.

The framework uses an extended version of the query tab available in Protege which was developed in the context of this research. The main modifications were introduced to allow the selection of the instance to be further investigated in the context of ontology analysis, for example, in the case described in the previous section a user could select all genes based on two variables: genes highly expressed in a particular area of the brain and genes related with diseases.

Conceptual Matching

Conceptual Matching is the act of mapping a database onto an ontological model, resulting in the importing of database records into the model. Conceptual matching is also used to integrate databases where records extracted from different sources are integrated by their semantic meaning as instances of a concept.

ODKD adopts the DataGenie plug-in for importing data from external databases (Figure 5-12). DataGenie enables the pulling of data from a database. It uses a JDBC and/or a JDBC-ODBC bridge to connect to a database and move portions (or all) of a database into an ontological model.



DataGenieExamp Protégé 3.0 (file:\C:\Program%20Files\Protege_3.0\plugins\edu.washington.datagenie\examples\DataGenieExa...
File Edit Project Window Help
protégé
Classes Slots Forms Instances DataGenie v2.0b
Connect Info
Begin Importing

Data Source Type ● ODBC ○ JDBC

ODBC Driver sun.jdbc.odbc.JdbcOdbcDriver

Data Source Name lamina

User Login

Password

Data Tables

□ System Table ☑ Table □ View □ Other

MPO_BRENT
MPO_UK
cusmanagers
customers
go_terms

Import Method

○ Straight-thru dump - all tables above
● Straight-thru dump - selected tables
○ Heuristic

☑ Preview Number of rows 50 (0 = all)

go_id	go_term
1	mitochondrion inheritance
2	mitochondrial genome maintenance
3	reproduction
4	biological_process unknown
5	ribosomal chaperone activity
6	high affinity zinc uptake transporter activity
7	low-affinity zinc ion transporter activity
8	thioredoxin
9	alpha-1,6-mannosyltransferase activity
10	trans-hexaprenyltranstransferase activity
11	vacuole inheritance
12	single strand break repair
14	single-stranded DNA specific endodeoxyribonuclease activity
15	phosphopyruvate hydratase complex
16	lactase activity
17	alpha-glucoside transport
18	regulation of DNA recombination
19	regulation of mitotic recombination
20	negative regulation of recombination within rDNA repeats
22	mitotic spindle elongation
23	maltose metabolism
24	maltose biosynthesis
25	maltose catabolism
26	alpha-1,2-mannosyltransferase activity
27	ribosomal large subunit assembly and maintenance
28	ribosomal small subunit assembly and maintenance

Figure 5-12 – DataGenie tab importing gene ontology data into the application ontology.

Generally, each table of a database becomes a class, each row becomes an instance, and each attribute becomes a slot. In addition, if a relational database table has foreign key references to other tables, these can be replaced by Protégé instance pointers when the database is converted into a knowledge base.

In Figure 5-12 data acquired from gene ontology is being imported into the biomedical ontology. The GO-ID is used to match the data while the GO-term which may be representing a molecular function is going to be imported in the internal GO-term slot. This process allows the mapping of a relational model into frames and the consequent acquisition of the instances from records available in the relational data.

There are other Protégé plug-ins available which may be suitable for domains other than the biomedical domain explored in this thesis. The OntoBase, for example, allows rapid construction of alternative representations of information contained inside relational databases (Ontospace, 2006).

As XML has become an industry standard for the structuring of information, the XML Tab (Figure 5-3) can also be used to import an XML document into Protégé, creating a set of classes and instances in a knowledge base which correspond to the entries in the XML document.

Ontology Population

Ontology population is similar to conceptual matching. It is the process of inserting instances from previously merged and/or aligned ontologies into the knowledge base. Ontology population normally occurs after conceptual model matching is executed. The main objective of this task is to include instances which are relevant for the problem domain or hypothesis being verified. For example, in a biomedical case study a user may want to include all genes which have a particular molecular function that may have been described in the literature or found in gene regulatory simulations as relevant to the study of a specific disease, instead of inserting all gene knowledge available in a biomedical ontology.

As described previously, this research had developed bespoke programs to import instances from biomedical ontologies at the beginning of the investigation. However, the creation and adoption of standards, such as OBO and subsequent development of the OBO Plug-inn along with Prompt plug-inn made this thesis decommission all previous code and adopted the plug-ins developed by the Protégé community which are suitable for this task with no current need for bespoke development.

5.2.4. Instance Preparation

The instance preparation phase covers all activities that construct the final dataset that will be fed into the modelling tool(s) from the ontological model. Tasks include concept, instance and slot (or attribute in a database taxonomy) selection, transformation and cleaning of instances as well as exportation to the modelling tools.

This phase helps the user to select the best candidates for a data mining exercise from a multidimensional perspective thereby reducing the features needed for a successful KDD task.

The Instance Preparation phase is supported by the Instance Selection Tab (Figure 5-7), which is able to execute all three ODKD tasks: instance cleaning, instance selection and instance export.

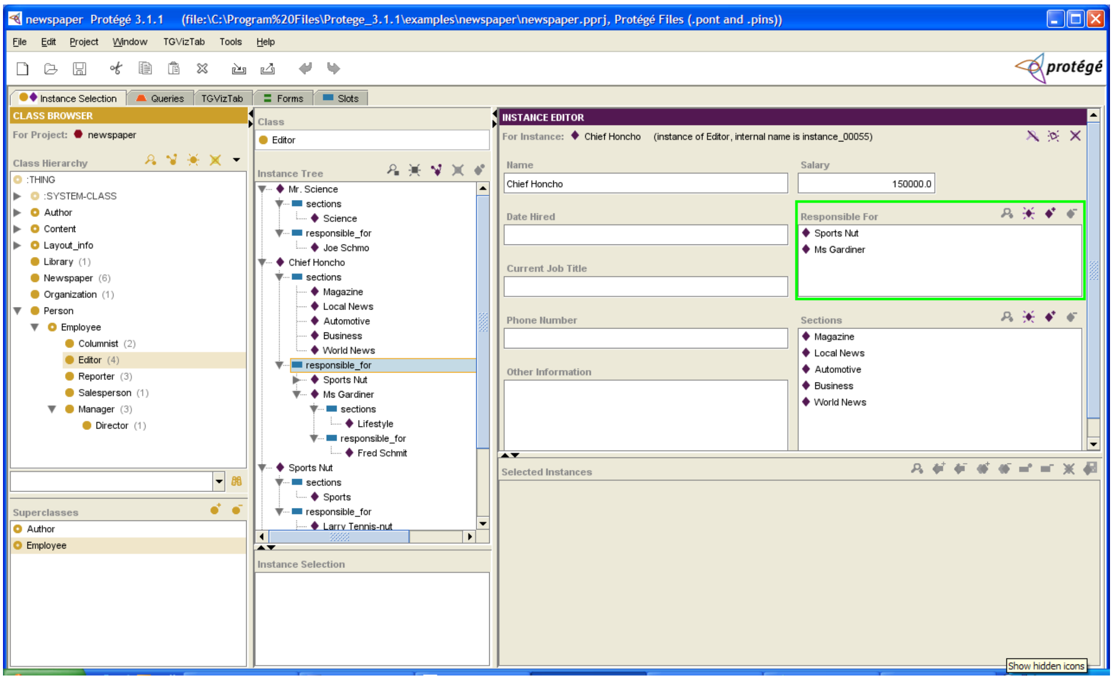
Instance Cleaning

Instance cleaning is the detection and removal of errors and inconsistencies from instances in a knowledge base (Rahm & DO, 2000).

Instance Cleaning Algorithm

```
Select a class  (Class Browser)
FOR each instance of the class
    FOR each slot of the instance (Instance tree browser)
        IF slot needs cleaning THEN
            Edit instance (Instance Editor)
        END IF
    END FOR
END FOR
```

The Instance Selection tab (Figure 5-13) enables direct editing of instances and slots (properties) of any instance in a knowledge base. The integrated class browser, instance editor, and instance tree allow complete navigation in an ontological model as well as the cleaning of the entire target knowledge base for a data mining task.



For Instance: Chief Honcho (instance of Editor, internal name is instance_00055)

Name: Chief Honcho

Salary: 150000.0

Date Hired:

Responsible For: Sports Nut, Ms Gardiner

Current Job Title:

Phone Number:

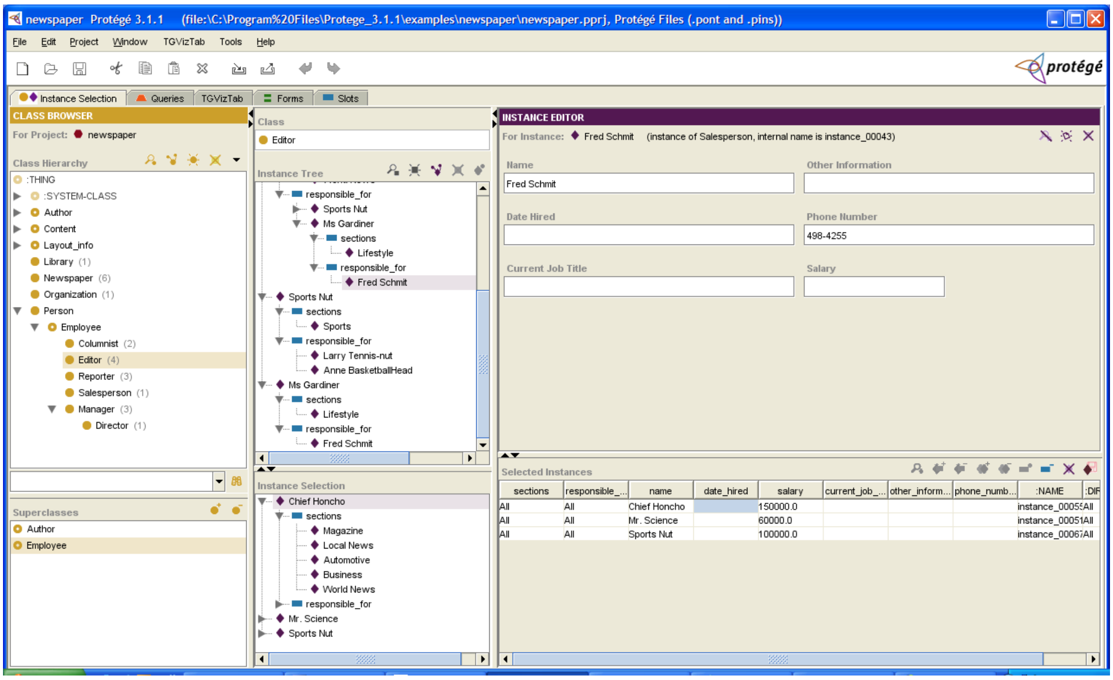
Sections: Magazine, Local News, Automotive, Business, World News

Other Information:

Figure 5-13 – A snapshot of an employee context in the Instance Selection Tab.

In Figure 5-13, for example, a complete view of an editor instance is depicted. It shows that *Chief Cocho* is a newspaper editor that inherits properties from both the ‘Author’ and ‘Employee’ classes. It also shows that *Chief Cocho* has two main relationships with other employees in the properties ‘responsible_for’ and ‘section’. This complete view allows for editing attributes not completed, such as date hired.

Instance cleaning, such as identification of missing values, becomes necessary when a set of instances are selected for a data mining task. In Figure 5-14, for example, the ‘date hired’ property would need to be filled in on all three instances in order to be able to be used in a data mining task. This shared, tabular visualisation enables the identification and editing of missing values and even highlights the need for cleaning the knowledge base.



For Instance: Fred Schmit (instance of Salesperson, internal name is instance_00043)

Name: Fred Schmit

Other Information:

Date Hired:

Phone Number: 498-4255

Current Job Title:

Salary:

sections	responsible_...	name	date_hired	salary	current_job_...	other_inform...	phone_numb...	:NAME	:DIR
All	All	Chief Honcho		150000.0				instance_0005!	All
All	All	Mr. Science		60000.0				instance_00051	All
All	All	Sports Nut		100000.0				instance_0006/	All

Figure 5-14 – A snapshot of a multiple selection in the Instance Selection tab.

Instance Selection

The instance selection task enables semantic-driven analysis of the knowledge base. It creates a semantic integration layer to all sources of data incorporated in the ontological model. This is done by representing data in terms of the ontological knowledge. Instead of concentrating effort on analysing data characteristics, this task concentrates its effort on giving meaning to data through domain understanding.

The objective of instance selection is threefold: to improve the prediction performance of predictors by analysing related perspectives, to provide faster and more cost-effective predictors by looking at a high-level abstract structure, and to provide a better understanding of the underlying process that generated the data.

There are three different approaches to selecting instances in the ODKD framework: tree navigation, network visualisation, and query selection.

Tree navigation is supported by the *Instance Selection* Tree, which is integrated with the Class Browser and the Instance Tree Browser (see Figure 5-15).

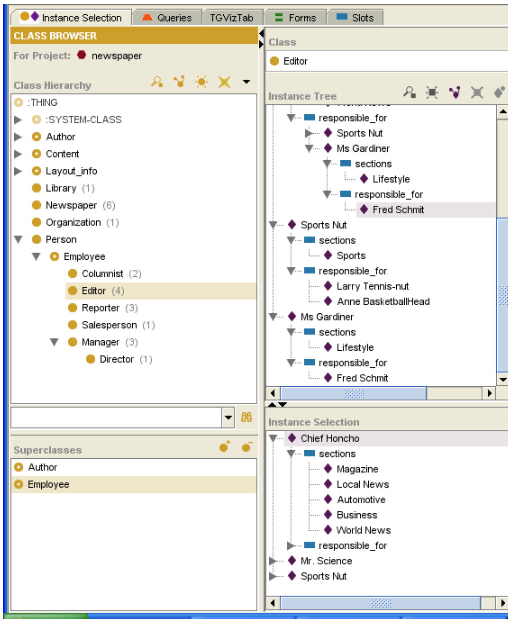


Figure 5-15 – Instance Tree.

The instance tree browser allows the selection of instances that are inserted into both the Instance Selection Tree component and the Selected Instances Grid for further manipulation and later export to data mining workbenches.

Network visualization selection is enabled by the TGVIZ tab extension (Figure 5-16). The addition of an instance selection feature to this tab enables visual analysis of the domain in order to select instance candidates for a data mining task.

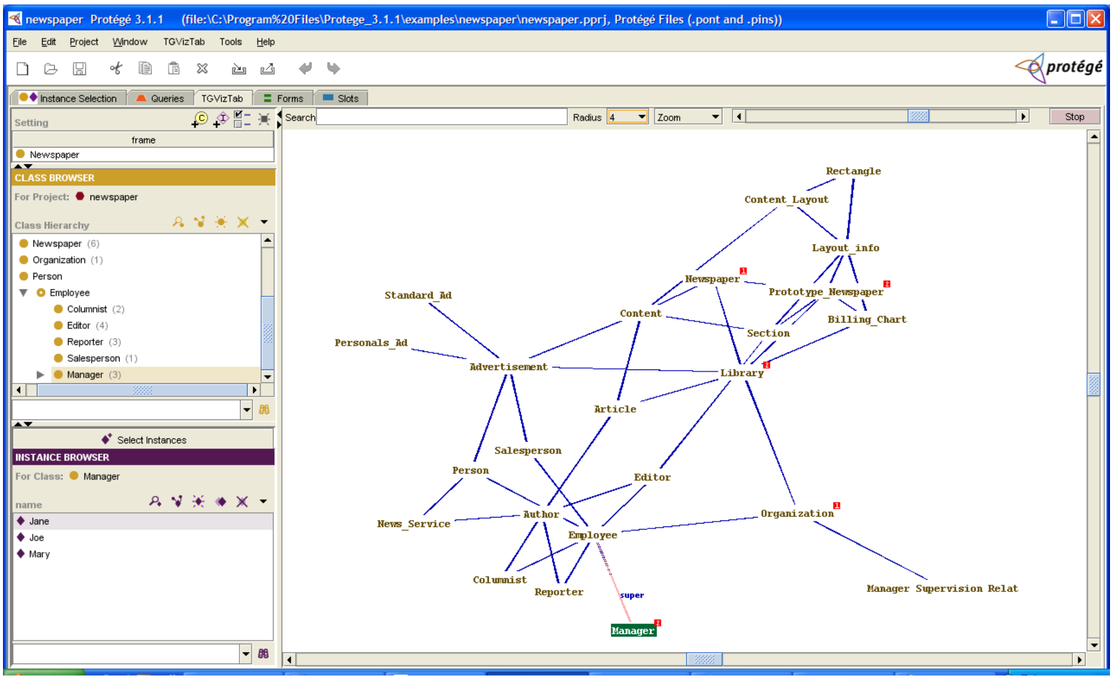


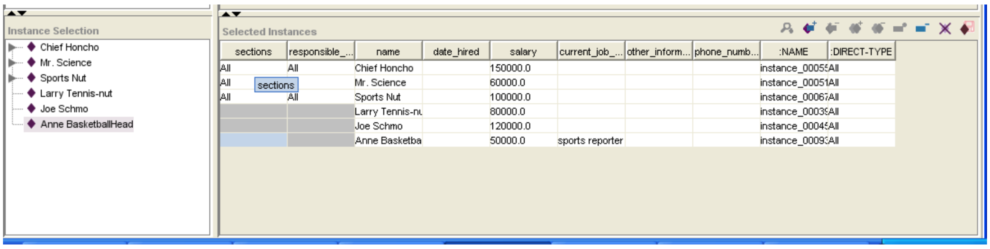
Figure 5-16 – The selection of a manager through the visualization of a complex network.

In the example shown in Figure 5-16 a set of managers can be selected through the identification of a class in network visualization. This feature is valuable when analysing the concentration of instances of a class. In this research it helped to identify classes which are predominant in a biomedical context promoting the selection of best candidates for data mining tasks such as clustering.

The network visualization, as described previously, helps to visualise a complex web of concepts and relationships, allowing both a high level view of the distribution of instances in a knowledge base as well as zoom in functionality in areas of interest for selection of concepts and or instances for further analysis. As explained in the methodology, the visualization and selection of instances is followed by the preparation of instances in a knowledge discovery process and proposed by ODKD and supported by the plug-ins developed in this research.

The **query selection** feature as described previously in this chapter is available in the framework through an extended plug-in based on the query tab (see Figure 5-11). The query feature searches the frame based knowledge representation using the class and slot frames based on the values of slots and/or classes. The results of the query are then selected using the query selection feature developed in this research.

After the initial selection of a set of candidates for a data mining task, the set can be further cleaned and/or analysed before it is exported. The following paragraphs and figures demonstrate some additional selection and manipulation scenarios using the "Selected Instances Grid" that is part of the Instance Selection Tab (Figure 5-17).



Instance Selection

- Chief Honcho
- Mr. Science
- Sports Nut
- Larry Tennis-nut
- Joe Schmo
- Anne BasketballHead

Selected Instances

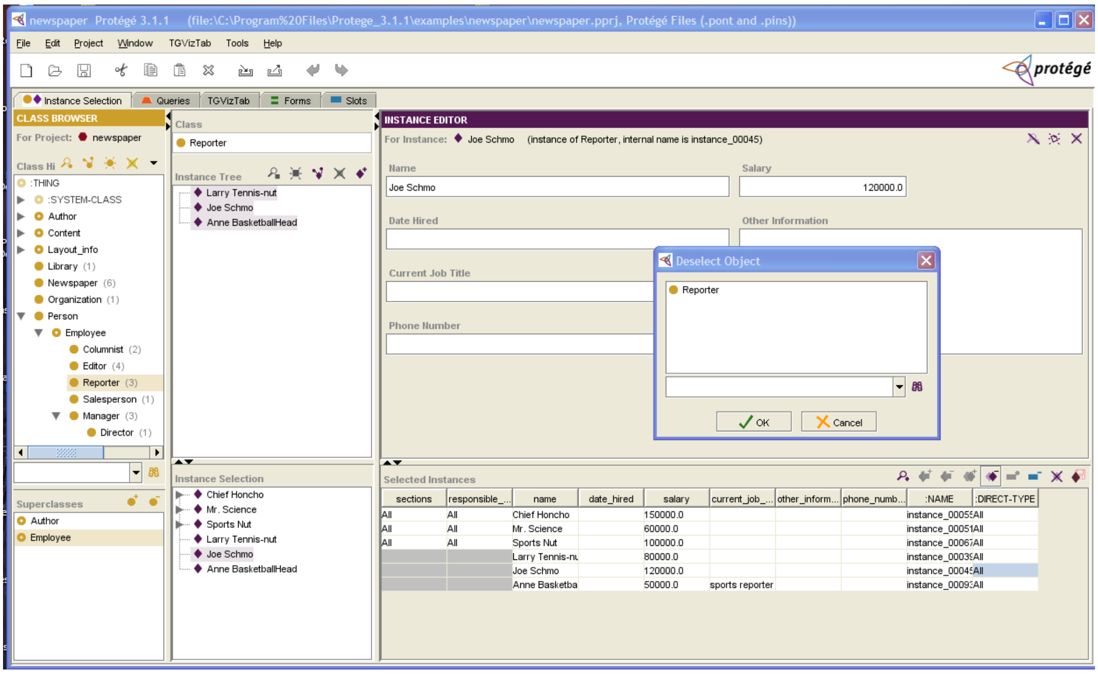
sections	responsible_...	name	date_hired	salary	current_job_...	other_inform...	phone_numb...	:NAME	:DIRECT-TYPE
All	All	Chief Honcho		150000.0				instance_00055	All
All		Mr. Science		60000.0				instance_00051	All
All	All	Sports Nut		100000.0				instance_00067	All
		Larry Tennis-nu		80000.0				instance_00039	All
		Joe Schmo		120000.0				instance_00045	All
		Anne Basketba		50000.0	sports reporter			instance_00093	All

Figure 5-17 – The Selected Instances Grid.

The tabular metaphor was adopted in the instance selection right bottom panel taking into consideration the user experience design and usability issues. In the context of knowledge discovery in database most data preparation tools use tables to select variables, to join records and so forth. The database community, for example, makes extensive use of tables in both design (relational modelling) and query and manipulation - most database vendors have their IDEs for data manipulation and query based on tables, as do most of the data mining workbenches such as Weka.

From a user interface perspective the instance selection is developed to represent the main tab in the ODKD framework. The user is able to navigate, annotate, manipulate, prepare and export instances from an unique window, therefore the table representation was considered the most appropriate to allow straightforward manipulation of instances selected by other features such as query and network visualization.

The grid enables several manipulations of the candidate instances, such as selection of specific properties, selection of sub-slots, selection of instances of a relationship based on some criteria and so forth. Figure 5-17 shows an example depicting employees who are not responsible for a section of a newspaper. In this case, the last three instances are grouped in a cluster representing the employees who are not editors - Figure 5-18 shows a reporter named Joe Schmo. This very basic clustering mechanism enables the grouping of instances based on their properties and classes as defined in the ontological model which enhances the semantic knowledge gained when preparing instances to be exported to a data mining workbench.



For Instance: Joe Schmo (instance of Reporter, internal name is instance_00045)

Name: Joe Schmo

Salary: 120000.0

Date Hired:

Other Information:

Current Job Title:

Phone Number:

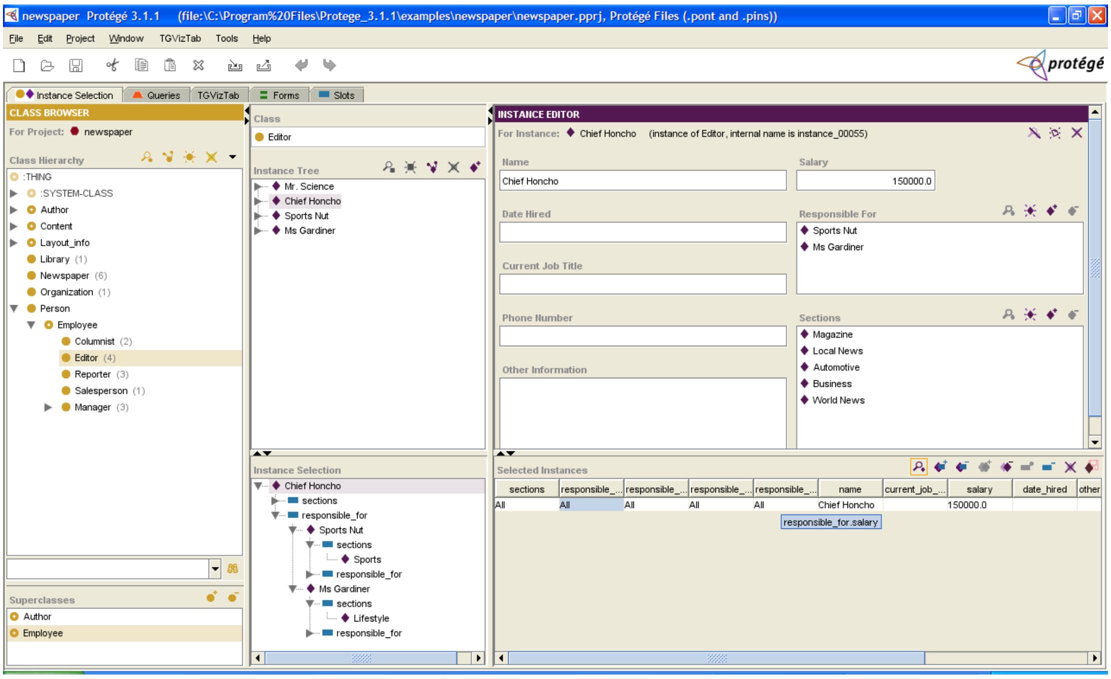
sections	responsible_...	name	date_hired	salary	current_job_...	other_inform...	phone_numb...	:NAME	:DIRECT-TYPE
All	All	Chief Honcho		150000.0				instance_00055	All
All	All	Mr. Science		60000.0				instance_00051	All
All	All	Sports Nut		100000.0				instance_00067	All
		Larry Tennis-nu		80000.0				instance_00039	All
		Joe Schmo		120000.0				instance_00045	All
		Anne Basketba		50000.0	sports reporter			instance_00093	All

Figure 5-18 –The class of an instance clustered by the grid functionality based on attribute sharing.

Instance Selection Ontological Clustering Algorithm

```
        FOR each visualization tool
            Select Instances
            FOR each instance
                    Select class of the instance
                    IF class is a cluster THEN
                            Insert instance as last of the cluster
                    ELSE
                            IF  direct-superclass is a cluster THEN
                                    Insert instance as last of the cluster
                            ELSE
                                    Create a new class cluster
                                    FOR each slot of the instance
                                            IF slot is required THEN
                                                    Select Slot
                                            ELSE
                                                    Unselect slot
                                            END IF
                                    Insert Instance after the last cluster
                                    END FOR
                            END IF
                    END IF
            END FOR
    END FOR
```

The *Selected Instances Grid* enables the selection of specific instances of a slot which has multiple values such as the slot *responsible_for* as shown in Figure 5-19. In this case, an employee, Chief Honcho, is responsible for two other employees, Sports Nut and Ms Gardiner. It also permits the selection of specific slots from these employees, such as the salary of all employees for whom Chief Honcho is responsible.



newspaper Protégé 3.1.1 (file:\C:\Program%20Files\Protege_3.1.1\examples\newspaper\newspaper.pprj, Protégé Files (.pont and .pins))

For Project: newspaper

Class: Editor

For Instance: Chief Honcho (instance of Editor, internal name is instance_00055)

Name: Chief Honcho

Salary: 150000.0

Responsible For: Sports Nut, Ms Gardiner

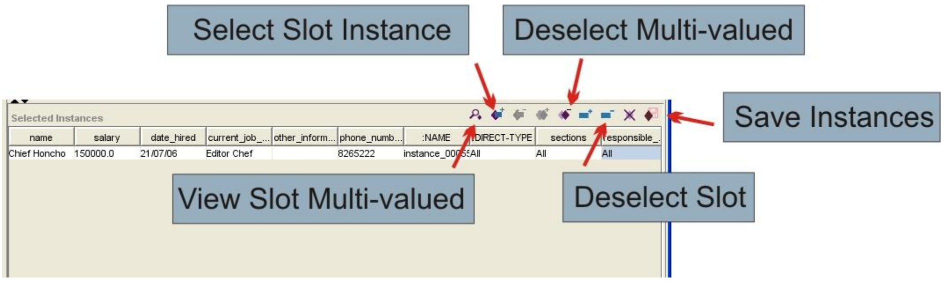
Sections: Magazine, Local News, Automotive, Business, World News

sections	responsible_...	responsible_...	responsible_...	responsible_...	name	current_job_...	salary	date_hired	other
All	All	All	All	All	Chief Honcho		150000.0		

responsible_for.salary

Figure 5-19 – A multi-dimensional selection of slots by the selected instances grid.

The *Selected Instance Grid* also enables the visualisation of specific slot values, the inclusion and exclusion of multiple value instances, the exclusion and inclusion of slots, reordering of slots and other functionalities (see Figure 5-20).



Selected Instances

name	salary	date_hired	current_job_...	other_inform...	phone_numb...	:NAME	DIRECT-TYPE	sections	responsible_...
Chief Honcho	150000.0	21/07/06	Editor Chef		8265222	instance_00	All	All	All

Figure 5-20 – A sample of different grid functionalities.

There are several features present in the instance selection tab which were developed to facilitate the manipulation, annotation and selection of variables (instances, slots and classes) as data preparation features for data mining from a semantic point of view using a single user interface.

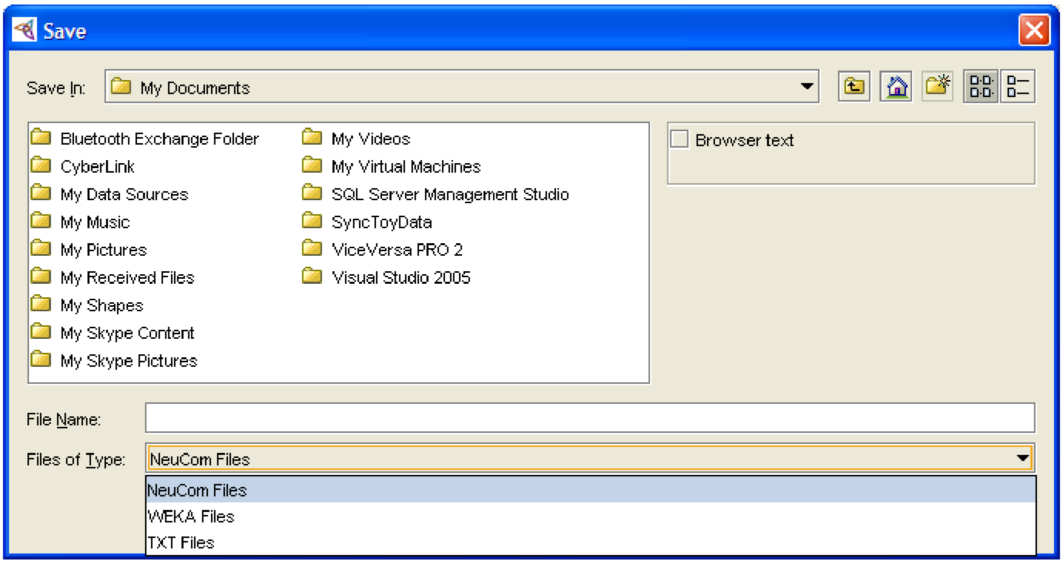
The next chapter shows an example in which a disease is investigated from a genetic perspective by selecting specific gene attributes that were not available previously in source databases. In this biomedical application, the molecular function of all genes related to a brain disease was exposed to a data mining technique for further analysis. The multi dimensional selection functionality is particularly useful for data mining tasks where the number of samples is small and the number of features large, such as in micro-array analysis.

In summary, these selection functionalities alongside the cleaning functionalities were developed to enable the preparation and selection of a set of samples and features for a data mining task in an ontology driven fashion as proposed by the ODKD methodology. This selected set can then be exported to enable further quantitative analyses and/or modelling as described in the next section.

Instance Export

Instance export is the next sub-task of Instance Selection of the ODKD process. It is concerned with the translation of the selected set of instances into a format used by a target data mining tool or workbench. The ODKD Framework enables the transformation of an ontological model into three main formats (see Figure 5-21): Weka (Witten & Frank, 2005), Neucom (KEDRI, 2006b) and flat file.

Although this thesis has developed features to support formats currently widely used by the data mining workbenches most adopted by both academia and industry, if a new format or standard is required its inclusion as an instance export functionality is very easy to be implemented since the design respected all Protégé architecture and design principles and remains based on a pluggable architecture. This design strategy has been part of the design practices throughout the development of new plug-ins as well as in the extension of basic Protégé functionalities.



Save

Save In: My Documents

□ Browser text

File Name:

Files of Type: NeuCom Files

Figure 5-21 – Export formats currently supported.

The **Weka format** utilized as a pattern to develop the export feature is based on the Attribute-Relation File Format (ARFF) specification defined in Witten (2005). The sample below (in Table 5-2) illustrates the information generated by the Weka export feature. Missing and null values are automatically included, as well as the attribute types as defined in the ontological model.

Table 5-2 – Sample of Weka output.

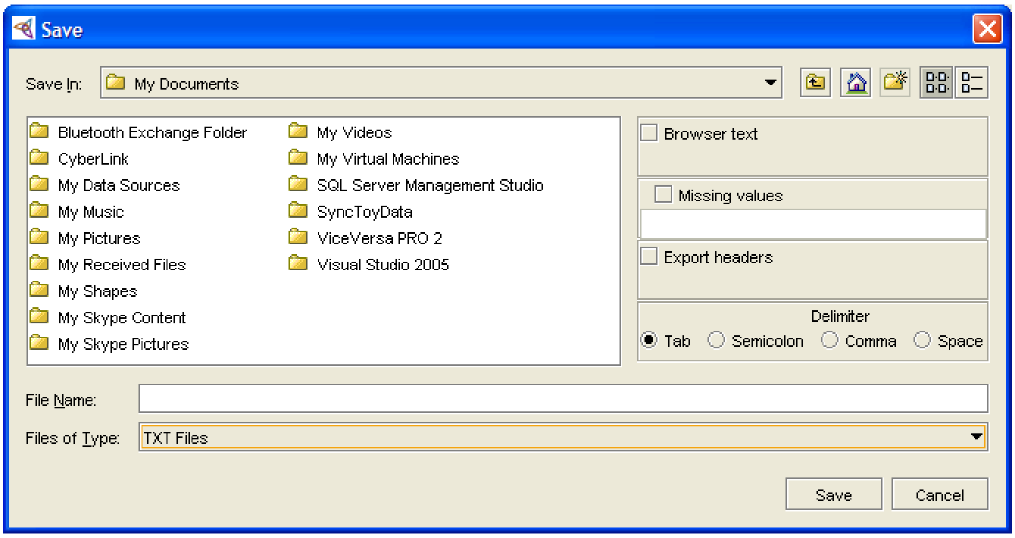
```
@relation WekaSample.arff
@attribute responsible_for
@attribute responsible_for.name {'Sports Nut','Ms Gardiner',&null}
@attribute responsible_for.salary {100000.0,45000.0,&null}
@attribute name string
@attribute salary numeric
@attribute date_hired string
@attribute :DIRECT-TYPE {Editor,&null}
@data
```

The **Neucom format** is based on a special text-based format, which is compatible with the Matlab format (MathWorks, 2006). The Neucom filter also automatically adds missing and null values in order to allow further data pre-processing by the workbench (see appendix C for more information about Neucom).

The browser text check box defines whether the internal name or a friendly name will be exported – samples below. This functionality is available for all formats (see Figure 5-22).

@attribute responsible_for {instance_00067,instance_00068,&null}

@attribute responsible_for {'Sports Nut','Ms Gardiner',&null}



Save In: My Documents

□ Browser text

□ Missing values

□ Export headers

Delimiter

● Tab ○ Semicolon ○ Comma ○ Space

File Name:

Files of Type: TXT Files

Figure 5-22 – Export options for text outputs.

Text format can be further manipulated in order to comply with different workbench formats. It enables the definition of special characters for missing values, the definition of delimiters, and *export headers* selection which defines the inclusion of a first line with the names of the attributes.

The instance export feature can be used to extract various samples of the same data set that can then be utilised by different workbenches. This feature is particularly valuable in data mining tasks where it is very likely that different modelling techniques and workbenches will be utilised in the deployment of a solution as defined in the CRISP-DM and in the Ontology Driven Knowledge Discovery process.

5.2.5. Modelling

The modelling task is driven by the selection of various modelling techniques as well as by the design and creation of test sets based on the problem before constructing data mining models. The modelling phase of the ODKD methodology is composed of three tasks: Model Selection, Test Set Design and Model Building. These three tasks are closely related to the Instance Preparation Phase, as different data mining models may require different data set formats.

The **Test Set Design** is the creation of training and test sets for the model building phase. This task works closely with the Instance Export sub-task by enabling the generation of different sets of data from the ontological model. Sets of data are usually generated before the definition of a data pipeline that will be deployed on a problem.

Model building or Onto-mining is learning from instances in the ODKD Framework. It is the application of data mining techniques to extract knowledge from the datasets acquired from the knowledge base.

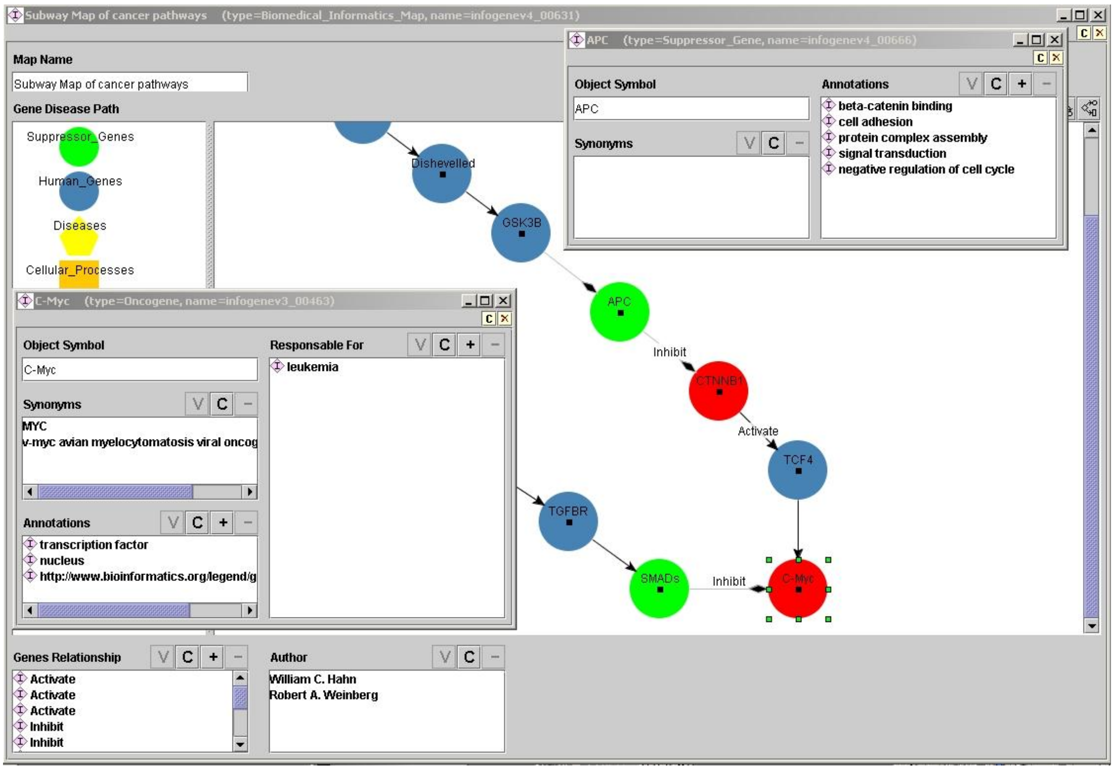
The ODKD Framework acts as a link between an ontology model and a workbench. It is not concerned with mining techniques themselves. It enables the integration of a multi dimensional ontological model and mining techniques by exporting data and acquiring knowledge as described in the next section.

5.2.6. Evaluation

This phase comprises the evaluation and acquisition of knowledge extracted by a data mining model. It is divided into three tasks: Knowledge Extraction, Knowledge Assessment and Ontology Learning.

Knowledge Extraction is responsible for the incorporation of knowledge extracted by the data mining technique into the ontological model. There are several different techniques reported in the literature which may be used for the process of mapping knowledge from a data mining technique to an ontological model. The next chapter describes the set of novel plug-ins developed to extract knowledge from a data mining technique in the specific context of a biomedical application.

Knowledge Assessment is the analysis of results obtained from a modelling technique. The framework supports a conceptual analysis of the results by creating a “quarantine structure”, as described in Chapter 3, where knowledge acquired from a modelling exercise can be stored for extra analysis in the ontology learning task without impacting on the main ontological model.


Subway Map of cancer pathways (type=Biomedical_Informatics_Map, name=infogenev4_00631)

Map Name: Subway Map of cancer pathways

Gene Disease Path

APC (type=Suppressor_Gene, name=infogenev4_00666)

Object Symbol: APC

Synonyms

Annotations:
- beta-catenin binding
- cell adhesion
- protein complex assembly
- signal transduction
- negative regulation of cell cycle

C-Myc (type=Oncogene, name=infogenev3_00463)

Object Symbol: C-Myc

Synonyms:
- MYC
- v-myc avian myelocytomatosis viral oncog

Responsable For:
- leukemia

Annotations:
- transcription factor
- nucleus
- http://www.bioinformatics.org/legend/g

Genes Relationship:
- Activate
- Activate
- Activate
- Inhibit
- Inhibit

Author:
- William C. Hahn
- Robert A. Weinberg

Figure 5-23 – A quarantine area represented as an Infogene Map.

The quarantine structure (illustrated in use in Figure 5-23), based on the *evolving ontology meta-class,* and the plug-ins developed to enable link analysis, are described in detail in the next chapter. Although demonstrated in the biomedical application developed in this research, the plug-ins and meta-class can support any task related to link analysis and relationship discovery.

Creation of the model and its assessment are generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a valuable form to the user. The **ontology learning task** enables final acceptance of the knowledge once the knowledge in the quarantine structure is considered stable enough to be used in other live KDD tasks.

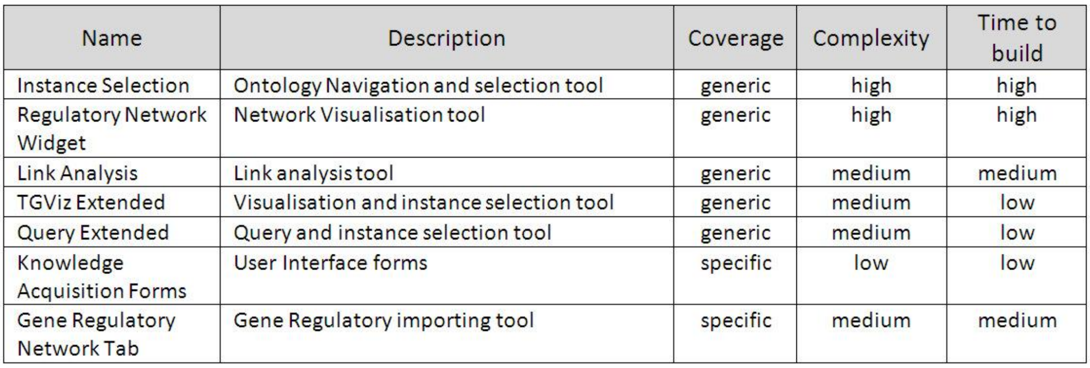
The biomedical application chapter (Chapter 6) demonstrates this cyclic process by integrating different sources of knowledge, applying a data mining algorithm, extracting knowledge, validating this knowledge using previous knowledge or triggering new mining exercises to integrate more experiments. There are a number of specific tools and widgets developed in this research to support the biomedical knowledge discovery tool and these are described in the next chapter.

5.3. Summary

The Ontology Driven Knowledge Discovery Implementation Framework provides a set of tools to support the hybrid ontology driven KDD process designed in this research. Each phase is supported by one or more Protégé plug-ins developed for and/or adopted from the Protégé plug-in open source library.

The table below (Table 5-3) describes the plug-ins developed in this research in terms of their indicative coverage, relative complexity and relative time to build. The description attribute states the key functionality provided by the plug-in. Coverage is related to the range of ODKD tasks supported by the plug-in and also considers its applicability on generic ontology engineering tasks. Complexity and time-to-build reflect the relative extent of new features developed and amount of code developed. Further information about the plug-ins can be found on the accompanying CD, where the source code is available along with some screenshots of the application, and also in Appendix B.

Table 5-3 – Plug-ins Summary.



Name	Description	Coverage	Complexity	Time to build
Instance Selection	Ontology Navigation and selection tool	generic	high	high
Regulatory Network Widget	Network Visualisation tool	generic	high	high
Link Analysis	Link analysis tool	generic	medium	medium
TGViz Extended	Visualisation and instance selection tool	generic	medium	low
Query Extended	Query and instance selection tool	generic	medium	low
Knowledge Acquisition Forms	User Interface forms	specific	low	low
Gene Regulatory Network Tab	Gene Regulatory importing tool	specific	medium	medium

The main goal of the framework is to create an extensible ontology environment able to support the cyclical knowledge-discovery-from-data process as well as integrate the environment with existing data mining workbenches.

The framework was implemented based on the design criteria established early in this research in Chapter 1. Each criterion is respected in the development of the tools, achieving the following results:

- The system supports all phases of the Ontology Driven Knowledge Discovery process and is based on the widely accepted and used Protégé ontology editor;

- The system reuses the evolving ontology meta-data and enables the adoption of new meta-data. It is also fully testable and representation formalism independent;

- The system is based on an extensible and flexible architecture which copes with the Biomedical Knowledge discovery domain and enables its extension by developing, extending, and/or adopting new plug-ins suitable for other domains;

- The system has an interface that supports the navigation of any ontological model;

- The system includes a set of query and visualization tools enabling the user to search and select instances using different plug-ins. It has also created a set of system objects that allow for the inclusion of these functionalities in any Protégé plug-in that is capable of displaying instances in their architecture;

- The system is able to export data to data mining workbenches. It also has a set of system objects which facilitates the development of new export formats if required;

- All the tools developed are published as independent and reusable modules in the Protégé open source library;

- The system has been used in different research projects as an ontology driven data mining platform as well leveraging other studies in the ontology engineering field.

The framework contributes to the development of the ontology field by making available a set of tools able to accomplish several ontology engineering tasks, such as import of data, visualization, selection and export of ontological knowledge. It contributes to the data mining research field by developing a tool able to integrate different sources of data for a data mining exercise. It also contributes to studies on the development of hybrid systems by integrating ontology engineering tools and data mining workbenches.

The best design practices for development and enhancement of Protege were adopted in this research. General best practices were also followed, such as those supporting reuse and loosely coupled components, as well as object oriented analysis and design principles.

An iterative approach was used, as described in the research methodology. The tools were built based on the requirements acquired from both the domain area and experts and also followed an iterative life cycle since the tools were developed jointly with the use of experiments and case studies.

The integration of different components was facilitated by two main factors. First, the Protégé open source community is mature and support and help are available through the community site and direct contact with the Protege team. Second, this research adopted whenever possible all core objects and functionalities available in the Protégé architecture and object model.

One of the main design principles was consistent user experience. Even though the prototype was mainly focused on the biomedical domain, the user interface design was developed to be generic and consistent with all Protégé guidelines. Some of the features especially developed to support the data mining tools and exercises used in the case study were also made generic to support some longstanding community requirements such as link analysis and instance export, for example the gene regulatory network analysis feature is also available for manual network acquisition and link analysis.

The application of the framework is demonstrated in the next chapter where a biomedical application is described. It has also been fully applied in other research projects such as in personalized modelling techniques used in the context of medical research (Verma, Song, & Kasabov, 2006).

Chapter 6

Biomedical Application

This chapter presents and discusses the biomedical knowledge discovery case study developed in this research that integrates a molecular biology knowledge base related to brain disease, the ODKD processes and tools, and a specific computational intelligence technique adopted as data mining technique to prototype the ontology driven knowledge discovery framework. It summarizes the practical contribution of this thesis and provides verification of the functionality of the system developed as part of the multi-methodological research methodology. The chapter starts by describing the case study life cycle then presents the Brain Gene Ontology (BGO), the Computational Neuro-Genetic Modelling simulator (CNGM) and continues with a description of the integration of both by the biomedical knowledge discovery tool.

6.1. Introduction

The steep explosion in volumes of biomedical data and the growing number of disparate data sources are exposing researchers in the field(s) to significant challenges in terms of the acquisition, maintenance and sharing of knowledge from large and distributed databases in the context of rapidly evolving research. Blagoskolonny and Perdee (2002), for example, presented the “Conceptual Biology” challenge: to build a knowledge repository capable of transforming the current data collection era into one of hypothesis–driven, experimental research. However their challenge was extended by the fact that in addition to research-informed literature, biomedical data is tremendously diverse and can consist of information stored in genetic code, identified in genomics and proteomics research by sequencing patterns, gene functions, and protein-protein interactions, and experimental results from various simulations. The latter, for instance, is conveyed by Bray in Theoretical Biology (2001):

> *“Computer models of action potentials, synaptic integration, heart contraction and even the movements of ions and molecules in cells are now so accurate that they can often be used as experimental objects in lieu of the thing they represent. Biologists can now design and test small genetic circuits in theory and then make them in actual living cells.”*

This chapter presents a novel biomedical knowledge discovery tool which takes into consideration both the Conceptual Biology challenge and the Theoretical Biology principles in order to build a hybrid intelligent system which acquires gene regulatory data from a simulation tool, validates the data using prior knowledge represented by the biomedical ontology, and supports human validation and annotation of both biomedical knowledge and the gene regulatory data acquired.

The knowledge discovery tool is applied in the context of the Brain Gene Ontology project (BGO) developed at the Knowledge Engineering and Discovery Research Institute (KEDRI) (KEDRI, 2006a) and leverages the power of the concepts, by means of ontology, and the power of theoretical simulations, using the Computational Neuro-Genetic Modelling simulator (CNGM) developed in the context of this research by neuroscientists and another researcher within a joint research project (Kasabov & Benuskova, 2004a).

The BGO is designed to be used for research, simulation and teaching at different levels of tertiary education. It is an evolving system that changes and develops with the addition of new facts and knowledge by both computer simulations and domain experts. It links selected structured bodies of physiological, genetic and computational information providing an analytical pathway for different types of users.

This biomedical application exemplifies the utilization of all the concepts and tools developed in this research and acts as an evaluation vehicle for the system development research methodology. It enables the validation of the complete life cycle of the Ontology Driven Knowledge Discovery process, described in Chapter 4, as well as focusing on gene regulatory network simulations that are incorporated as new knowledge in the ontological framework.

The chapter begins with the description of the process followed in the case study then the Brain-Gene ontology developed in this research, its design criteria, and some sample data. This is followed by a brief description of the ontology model and the CNGM simulator developed in the context of KEDRI's research. The biomedical knowledge discovery tool, integrating the Brain-Gene Ontology, CNGM simulations and the ODKD framework, is then described and the incorporation of knowledge from the simulation is presented.

6.2. Case study life cycle

This section briefly explains the instantiation of ODKD methodology when applied to this biomedical case study. It aims to facilitate the reading of the text by creating a mapping between the generic ODKD methodology and its application duri

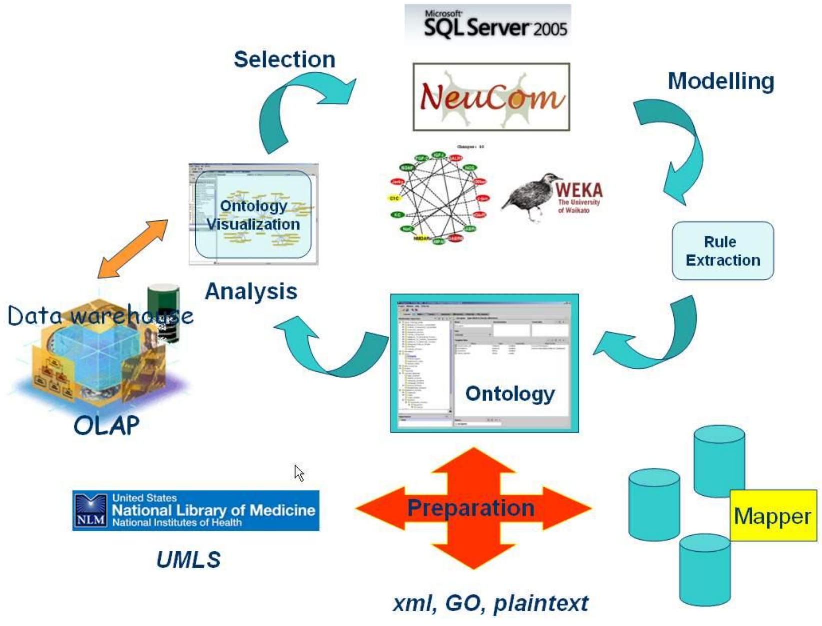


Figure 6-1 – BGO case study life cycle sample.

Figure 6-1 represents a high level description of the process followed when developing the case study. As described previously, the case study followed an iterative research methodology, and therefore the body of knowledge evolved and developed throughout the thesis. For instance, the medical knowledge acquired in the Leukaemia case study which was enhanced to also cover the brain disease knowledge.

The identification of the domain, brain gene related knowledge, formed the basis for the application ontology selection while the medical knowledge was acquired mainly from the UMLS, as shown in Figure 6-1. Several other data sources were also identified (described in the next section) and then used to populate the biomedical ontology.

The set of visualization, knowledge acquisition interfaces and query tools were used by the experts to acquire and annotate new relevant knowledge related to the brain gene data. Some early experiments were also developed in the context of aligning the multidimensional ontology knowledge and online analytical processes available in the business intelligence domain, called here OOLAP. A description of the experiment is presented in the Appendix A as a reference for future work and thesis contribution since this is not part of the core investigation of this research.

The knowledge acquired and iteratively and jointly created in the biomedical 3ontology was then used to support the selection and tuning of parameters used in the simulation.

The Computational Neuro-Genetic Modelling simulator was used to simulate possible gene regulatory networks which mimic real epilepsy data. The best gene regulatory network candidates were imported into the biomedical ontology using the set of especially developed plug-ins to acquire gene regulatory data. Experts and students could then access the data for further investigative work as well as to understand the genetic triggers of the brain disease.

This process then closed the loop of the biomedical knowledge discovery, as suggested by Blagoskolonny and Perdee (2002), and Bray (2001).

Appendix B contains a series of screenshots related to these tools with real examples developed in the case study. It also includes a table which maps the ODKD process and supportive tools presenting all adopted, extended and newly developed plug-ins and features.

6.3. The Brain Gene Ontology - BGO

The goals of the Brain Gene Ontology project are the development and representation of knowledge regarding genes and proteins that are related to specific brain disorders such as epilepsy and schizophrenia. The current stage of development focuses on important neuronal parameters such as AMPA, CLC, GABA, and KCN through their direct or indirect interactions with other genes and proteins as reported in (Benuskova & Kasabov, 2007). The ontological model provides the conceptual framework and the knowledge itself to enable understanding of relationships between those genes and their links to brain disorders. It also provides a semantic repository of systematically ordered relevant concepts in molecular biology. In particular, BGO provides conceptual links between data from seemingly disparate fields, which include, for example, information collected from the literature and gene sequence analysis.

According to Bodenreider and his colleagues (Bodenreider, 2001), biomedical ontologies organize the concepts *"involved in biological entities and processes in a system of hierarchical and associative relations that allows reasoning about biomedical knowledge."* The knowledge in the BGO thus integrates gene information from a number of distributed biomedical databases. The data collection and data entry of BGO are based on a series of pre-established criteria which were defined taking into consideration domain knowledge and knowledge validated by domain experts. The domain expert team was composed of two biologists (Dr. Vishal Jain and Dr. Ilkka Havukkala) and one neuroscientist (Dr. Lubica Benuskova). These criteria ensure the accuracy and authority of the stored gene knowledge as described in the following sub-sections.

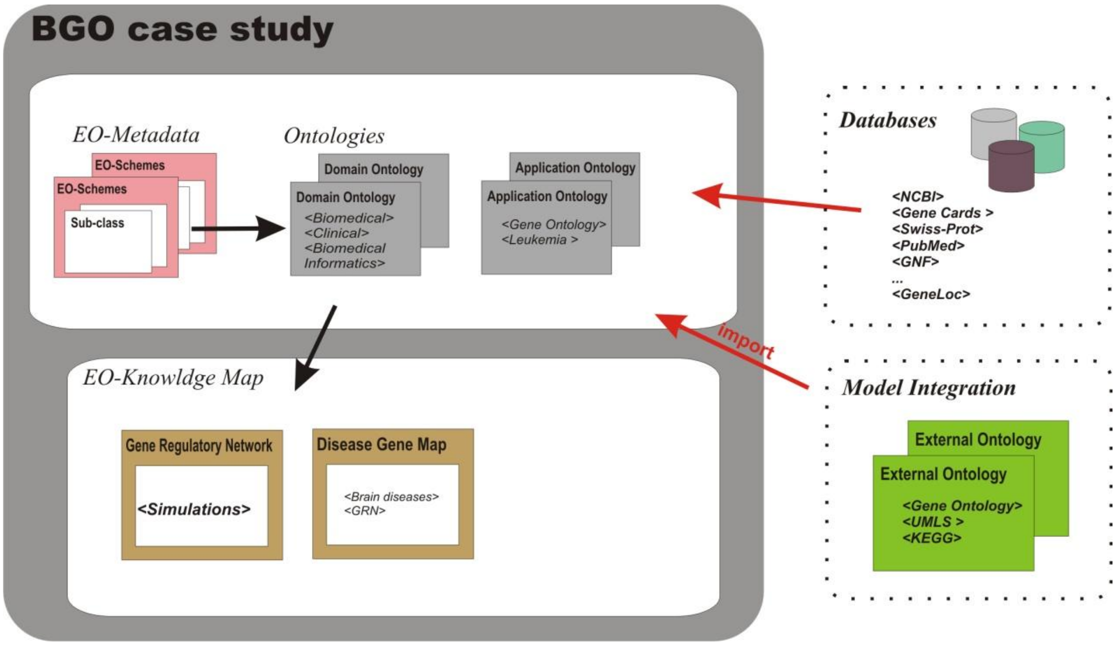


Figure 6-2 – The Brain-Gene Ontology diagram

Figure 6-2 represents a diagrammatic overview of the brain gene ontology knowledge base. The diagram shows different data sources and models, for example, those data sources used to annotate the brain gene related knowledge.

6.3.1. Biomedical Informatics Sources

The Brain Gene Ontology is based on widely used biomedical knowledge sources. It reuses and integrates many different specific ontologies such as the Unified Medical Language System (UMLS) (NCI, 2003) and the Gene Ontology (GO, 2006) as well as standard form databases.

The **Gene Ontology** provides the main knowledge repository of genes' and proteins' roles in cells. The BGO incorporates three relevant sub-ontologies of GO: molecular function, biological process and cellular location. The **UMLS** (Unified

Medical Language System) acts as the main source for the integration of biomedical terms and diseases. The UMLS provides information from over 100 controlled vocabularies and classification systems to the BGO. Its knowledge covers a wide range of domains from information used in patient records and in the research literature to disease specific knowledge.

Although GO and UMLS provide an appropriate umbrella under which controlled vocabularies can be organised for sharing and reuse, and the databases (described over) complement the knowledge and/or are a source of instances for the ontological model, the Brain-Gene Ontology (BGO) development is both more specific and more broad at the same time. It is more specific in that it is focused on the brain, it is broader in that it is intended to consider other knowledge sources, sources external to GO and UMLS (see Figure 6-3). It should facilitate the integration of information from different disciplinary domains such as neuroscience, bioinformatics, genetics and computer and information sciences (Kasabov, Jain, Gottgtroy, Benevuska, & Joseph, 2007). It is based on new and growing data on the influence of genes upon brain function ("The Nervous System. In: Genes and Disease", 2005).

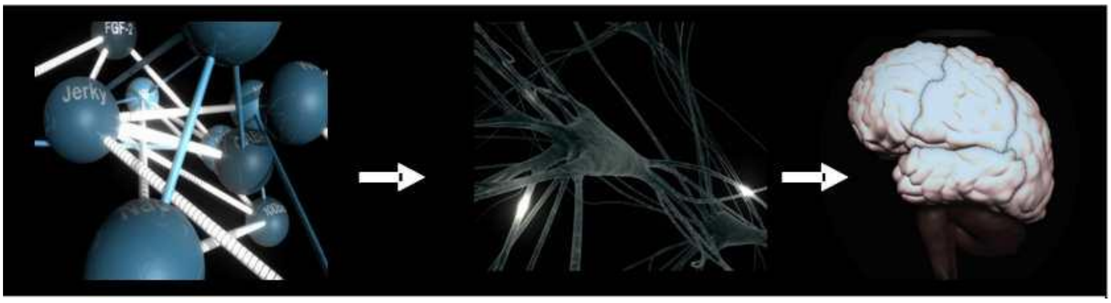


Figure 6-3 – The Brain-Gene Ontology is concerned with complex relationships between genes, their influence upon neurons and the brain

A large amount of information therefore was also extracted from distributed **databases**, including:

- NCBI (National Center of Biotechnology Information) – NCBI is one of the most well-known molecular biology information centres. It provides entries for databases of both genome sequencing data and biomedical research literature. NCBI focuses on the understanding of molecular processes affecting human health and disease, and develops sequence search engine and software tools for analysing genome data and biomedical information.

- Gene Cards (Rebhan, Chalifa-Caspi, Prilusky, & Lancet, 1997) – This is one of the main knowledge sources for BGO development. It integrates knowledge of human genes and the related annotations in a comprehensive topic range. It also provides links about specific genes to more than 50 instances in other databases. Basically, most information known about a human gene such as automatically-mined genomic, proteomic and disease relationships can be found with Gene Cards. The search engine of Gene Cards supports both simple and advanced searches.

- Swiss-Prot (Bairoch, Boeckmann, Ferro, & Gasteiger, 2004) – This provides gene information from other organisms. The accuracy of this database was one of the main reasons for its adoption. The annotations that Swiss-Prot provides for sequences are the result of a labour-intensive process that includes assessment of information from published articles along with use of a variety of programs and algorithms. Swiss-Prot is not only an annotated protein sequence database, but also presents other valuable biochemical information.

- PubMed (NBCI, 2006) - This is the source of most of the literature annotation in BGO. PubMed is a biomedical journal database of the U.S. National Library of Medicine. It was built in the 1950s and up to now, has included over 16 million biomedical citations and other life science journals. Through PubMed query, researchers can find full text articles and other related resources.

The selection of these knowledge sources followed the prioritisation method as defined by the Ontology Driven Knowledge Discovery methodology (Chapter 4). The domain experts were responsible for the selection of the most representative biological databases/ontologies from a compilation of several biomedical informatics knowledge repositories while the ontology engineer added a feasibility dimension to the quadrant analysis. As a result, the above databases and ontologies were selected as main sources following the ODKD methodology.

A list of resources used in the BGO, are introduced briefly in the source table below:

Table 6-1 - Candidate biological knowledge bases for data collection

Name	URL Address	Type of the Knowledgebase
Gene Card	http://www.genecards.org/	An integrated database of human genes
Genbank	http://www.ncbi.nlm.nih.gov/Genbank/index.html	A genetic sequence database

Genes and Disease	http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=gnd.chapter.75	A collection of articles that discuss genes and the diseases to which they are connected.
KEGG	http://www.genome.jp/kegg/	An encyclopaedia of genes and genomes
Allen Brain Atlas	http://www.brain-map.org/	An interactive, genome-wide image database of gene expression in the mouse brain
Swiss-Prot	http://us.expasy.org/sprot/	A collated protein sequence database
Ensembl	http://www.ensembl.org/	A project which produces and maintains automatic annotation on eukaryotic genomes
Gene Ontology	http://www.geneontology.org/	A controlled vocabulary to describe gene and gene product attributes in any organism
GNF	http://expression.gnf.org/cgi-bin/index.cgi	An RNA expression database
GeneLoc	http://bioinfo2.weizmann.ac.il/geneloc/index.shtml	An integrated map for each human chromosome
GeneNote	http://bioinfo2.weizmann.ac.il/cgi-bin/genenote/home_page.pl	A database of human genes and their expression profiles in healthy tissue

Data regarding 93 significant brain genes, 70 proteins and 14 protein complexes with a number of attributes were collected from these online knowledge bases to form the current BGO ontological model. This knowledge is the basis for the interpretation of the simulation results detailed later in this chapter.

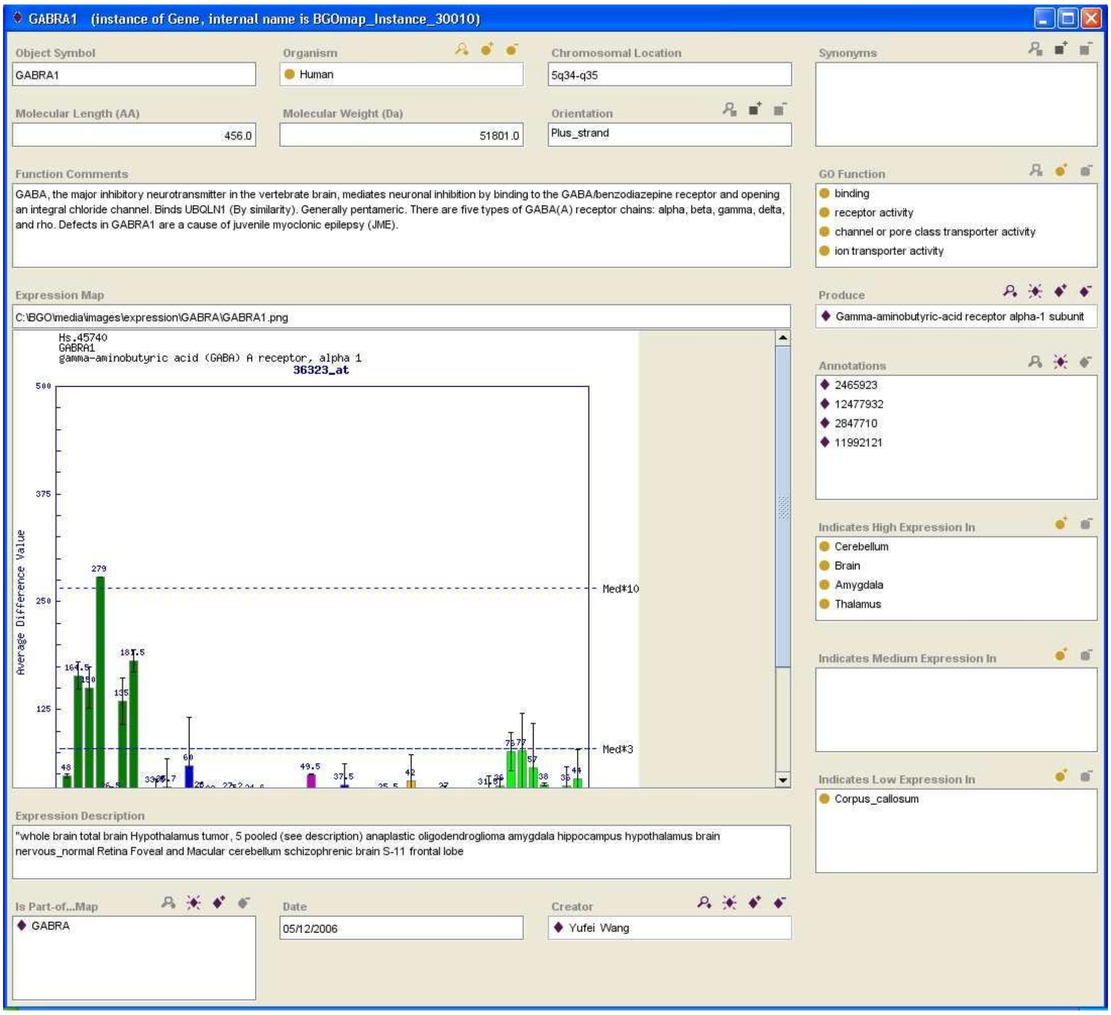
Table 6-2 and Figure 6-4 show a sample of gene information gathered and integrated in the BGO. It shows information related to subunits of the AMPAR (amino-methylisoxazole-propionic acid) receptor.

Table 6-2 - Gene and protein sample data for AMPAR receptor (Wang, 2007) .

Protein Name	Gene Name	Function Comments	Molecular Weight/Length	References at NCBI

Glutamate receptor 1	GRIA1	L-glutamate acts as an excitatory neurotransmitter at many synapses in the central nervous system. The postsynaptic actions of Glu are mediated by a variety of receptors that are named according to their selective agonists.	101536Da;
			906 AA
				PubMed=1311100 [NCBI]
				PubMed=1320959 [NCBI]
				PubMed=1652753 [NCBI]

Glutamate receptor 2	GRIA2	Receptor for glutamate. L-glutamate acts as an excitatory neurotransmitter at many synapses in the central nervous system. The postsynaptic actions of Glu are mediated by a variety of receptors that are named according to their selective agonists. This receptor binds AMPA(quisqualate) > glutamate > kainate. Interacts with PRKCABP, GRIP1 and GRIP2 (By similarity).	98821Da;
			883 AA
				PubMed=8003671 [NCBI]
				PubMed=12477932 [NCBI]
				PubMed=7523595 [NCBI]



GABRA1 (instance of Gene, internal name is BGOmap_Instance_30010)

Object Symbol: GABRA1

Organism: ● Human

Chromosomal Location: 5q34-q35

Synonyms:

Molecular Length (AA): 456.0

Molecular Weight (Da): 51801.0

Orientation: Plus_strand

Function Comments: GABA, the major inhibitory neurotransmitter in the vertebrate brain, mediates neuronal inhibition by binding to the GABA/benzodiazepine receptor and opening an integral chloride channel. Binds UBQLN1 (By similarity). Generally pentameric. There are five types of GABA(A) receptor chains: alpha, beta, gamma, delta, and rho. Defects in GABRA1 are a cause of juvenile myoclonic epilepsy (JME).

GO Function:
- ● binding
- ● receptor activity
- ● channel or pore class transporter activity
- ● ion transporter activity

Expression Map: C:\BGO\media\images\expression\GABRA\GABRA1.png

Produce: ◆ Gamma-aminobutyric-acid receptor alpha-1 subunit

Annotations:
- ◆ 2465923
- ◆ 12477932
- ◆ 2847710
- ◆ 11992121

Indicates High Expression In:
- ● Cerebellum
- ● Brain
- ● Amygdala
- ● Thalamus

Indicates Medium Expression In:

Indicates Low Expression In:
- ● Corpus_callosum

Expression Description: "whole brain total brain Hypothalamus tumor, 5 pooled (see description) anaplastic oligodendroglioma amygdala hippocampus hypothalamus brain nervous_normal Retina Foveal and Macular cerebellum schizophrenic brain S-11 frontal lobe

Is Part-of...Map: ◆ GABRA

Date: 05/12/2006

Creator: ◆ Yufei Wang

Figure 6-4 - Gene data sample within the Brain Gene Ontology.

6.3.2. Brain Gene Ontology Design Criteria

The Brain-Gene ontology has been developed as part of the BGO project at the Knowledge Engineering and Discovery Research Institute (KEDRI) at the Auckland University of Technology. The project is divided into three main phases: proof-of-concept, disease centric release, and integrated release. The first two versions are presented in this chapter while the integrated version, which widely extends the current brain gene knowledge as well as including other domains such as nutritional knowledge, is discussed in Chapter 8 – Contribution / Future research.

The design process followed the guidelines of the ODKD methodology and the design criteria described in this section. The proof-of-concept scope was defined by the domain experts (as described in the ODKD methodology). The following set of questions was selected:

➢ What is the knowledge that BGO will cover?

– BGO represents information about brain genes, proteins, diseases and the relationships between them.

➢ What is the BGO going to be used for?

– BGO is to be used in the study of brain genes and diseases, emphasizing which genes are highly expressed in a brain disease.

➢ What types of questions should the BGO provide answers for?

– Concepts related with brain diseases such as molecular functions and molecular length, identified by current ontologies and databases and collated by the Institute’s domain experts.

➢ Who will use and maintain the BGO?

– Bioinformatics students and researchers are the potential users and maintainers of the BGO.

The BGO was developed to provide sufficient breadth of information and certain specific levels of detailed information about brain genes, proteins and diseases as accorded by the domain experts and the ontology engineer.

The *proof-of-concept* version is the basis for the *disease centric version* which extends the initial BGO version by including some specific disease concepts as well as incorporating instances into the ontological model.

The Brain Gene Ontology Criteria for Scope

A set of **Competency Questions** (Gruninger & Fox, 1995) was defined in addition to the quadrant analysis previously executed in order to refine the scope. The following questions were initially identified by the domain experts as necessary to be answered by the BGO:

- What is the name of the gene/protein related to brain disease?
- Is that a protein complex?
- What is the chromosomal location of the gene?
- What is the molecular length of the gene/protein?
- What is the molecular weight of the gene/protein?
- Which protein does this gene produce?

- What are the reference articles for this gene/protein in PubMed?
- What is the function of the gene/protein?
- Are there any other comments?
- How is the gene expressed in its expression map?
- What is the orientation of the protein/gene?
- What is the source of the gene/protein? (Human/animal)

Although these questions are not exhaustive, and have been constantly extended through the development of BGO, the initial set helped to identify extra knowledge sources and instances related to the brain gene disease domain. Table 6-1 shows a list of information sources utilized to address the questions covered in the case study.

The Criteria for Ontology design

The design of the ontology obeyed the basic principles that have been proven valuable in previous system development as well as those reported as best practice in the ontology engineering research field(Aspirez, Gomez-Perez, Lozano, & Pinto, 1998). These design principles can also be seen as objective criteria for guiding good ontology design and are explained as follows:

The **standardization of names** (Aspirez, Gomez-Perez, Lozano, & Pinto, 1998): Besides the naming rules in Protégé, the naming conventions of BGO are the same for all the name related terms in order to make the BGO easily understood. For example: *EO_Creator_Scheme* and *EO_Annotation_Scheme* follow the same naming conventions while; *EO_Creator_Scheme* and *Annotation_SchemeInEO* do not.

Minimization of the syntactic distance between sibling concepts (Aspirez, Gomez-Perez, Lozano, & Pinto, 1998): The creation of sibling concepts in BGO is based on the same pattern; the representation of these sibling concepts uses the same primitives if possible. This should improve the comprehensibility and reusability of the BGO.

The **representation of disjointed and exhaustive knowledge** (Aspirez, Gomez-Perez, Lozano, & Pinto, 1998): Generally, subclasses are disjointed if they do not have any common instances. Two subclasses in BGO are defined as disjointed decomposition when using the Protégé system if they do not have common instances.

Clarity (Gruber, 1993a): All the definitions (classes) in BGO are objective. They are stated and documented with natural language in BGO.

Extendibility (Gruber, 1993a): Future users of the BGO are able to define new terms for extended usage, for example, the four classes in the red box in Figure 6-5 can be created for the extension of the BGO. This extended content is based on existing vocabulary and did not require revision of existing definitions.

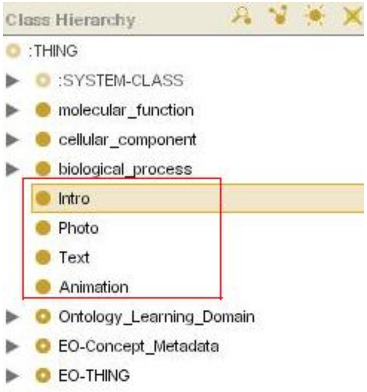


Figure 6-5 - Brain-Gene Ontology extended with learning object concepts used later in the project.

Coherence (Gruber, 1993a): The slots and facets of the definitions in BGO are limited to a reasonable range based on axioms. Anything that causes a definition or instance given to contradict the axioms is fixed.

Minimal ontological commitments (Gruber, 1993b): To support knowledge sharing, BGO specifies the weakest theory and defines only the terms that are crucial to the communication of knowledge consistent with the theory. This minimizes ontological commitments and increases the reusability of the definitions in BGO in other systems.

The Criteria for Reusing Existing Sources

There are two main ways to build an ontology: by reusing a developed ontology or from scratch. The ODKD methodology defines a process for the identification and prioritisation of knowledge sources from both domain expert and ontology engineer perspectives (Chapter 4 - section 4.3.1)

The prioritisation process identified two main ontology sources in the biomedical informatics domain related to brain gene diseases: the Gene Ontology and the Unified Medical Language System. Information about specific genes and proteins was also extracted from well-known bioinformatics databases.

The prioritisation process helped to define the granularity and depth of the knowledge. Although the Gene Ontology was used in its full length at the beginning of this research, its evolution and several extensions made it extremely comprehensive which extrapolated the necessary brain gene knowledge for the BGO project. A *GO Slim* version was then selected as the basis for the first version of the Brain Gene Ontology extended in subsequent research.

GO Slims are cut-down versions of the GO ontologies containing a subset of the terms in the whole GO. They give a broad overview of the ontology content without detail of the specific fine grain terms. *GO Slims* are particularly useful for giving a summary of the results of genome GO annotation, microarray, or cDNA collection when broad classification of gene product function is required or when the needs may be specific to species or to particular areas of the ontology such as in the initial version of the BGO project.

Some criteria were also defined to guide the importing process:

Comparative selection: The knowledge model of BGO used the experiences of both GO and UMLS along with other small ontologies and domain expert knowledge.

Almost all of the data is annotated by multiple sources from different genetic databases. These were marked and linked to the annotation part of the BGO. The different sources of the same unit were also recorded in the BGO through the evolving ontologies metadata (Chapter 3).

The adoption of Plug-ins: Importing and comparison were conducted using Protégé plug-ins. Three plug-ins were mainly used in the importing process: PROMPT Tab, UMLSTab, and OBO Tab. UMLSTab was adopted to access the UMLS knowledge base. The PROMPT Tab was used to compare and merge the imported parts with BGO and track any changes that happened during the whole process. The OBO plug-in was used to import the GO-Slim.

Database evaluation: A list of biological sources covering the whole domain was evaluated by domain experts. The brain gene data and information were collected from large-scale, stable, credible databases. There is also a knowledge acquisition feature in the biomedical knowledge discovery tool, developed in this research, where domain experts can share and annotate information in the field. Each piece of data included was from more than one source forming the BGO. Different results were stored as different categories by source.

Criteria for BGO Structure Evaluation and Biological Data Verification

Domain experts were available to supervise the BGO structure evaluation and biological data verification in the BGO project throughout the development life cycle.

This joint and iterative development process avoided the inclusion of a number of biological and medical resources which, in spite of their availability on the World Wide Web, are not credible, comprehensive or fully related to the specific brain gene disease domain. The domain expert engagement followed the guidelines of the ODKD methodology.

The next section describes the computational model utilised in the case study to simulate gene regulatory networks and inform the results with ontological knowledge.

6.4. A generic computational neurogenetic (CNG) model

Neuroscience, along with the information and mathematical sciences, has developed a variety of theoretical and computational models to enable complex brain functions to be represented and analysed (Kasabov & Benuskova, 2004b). Computational Neuro-Genetic Modelling (Kasabov, Benuskova, & Wysoski, 2004) is a novel computational approach which uses this information to create models that integrate dynamic gene networks with a neural network model.

The CNGM is biologically-based on the fact that a specific gene from the genome relates to the activity of a neuronal cell by means of a specific protein as well as the complex interaction between genes and proteins within the gene/protein regulatory network (GPRN), and this defines the functioning of each neuron and a whole neural network in its turn. It also acknowledges that even in the presence of a mutated gene in the genome, that is known to cause a brain disease, the neurons can still function normally provided a certain pattern of interaction between genes is maintained (Marcus, 2004). On the other hand, if there is no mutated gene in the genome, certain abnormalities in brain function can be observed as defined by a certain state of interaction between genes (Marcus, 2004).

In the Computational Neuro-Genetic Model, interaction of genes regulates the activity of neurons that consequently affects the dynamics of the whole neural network. It has been shown that by tuning the interaction between genes and the initial gene/protein expression levels, different states of neural network operation can be achieved. The main goal of Neuro-Genetic models is thus to simulate the activity of certain parts of the brain using biologically plausible neural networks and to link these activities and their outcomes to the genetic level, in an attempt to enable discoveries of yet unknown dynamic relationships between genes and states of brain activity.

The CNGM approach is adopted in this research as a data mining technique. The model was developed in the context of a joint research project between several researchers including the author of this thesis. The work developed in this thesis contributed to the simulation by identifying a series of biologically plausible parameters, mainly related to genes highly expressed in the brain, which were used to set up the simulations.

In general, the model considers two sets of essential genes – a set $\mathbf{G}_{\text{gen}}$ that defines generic neuronal functions (e.g. general cell life functions) and a set $\mathbf{G}_{\text{spec}}$ that defines specific neuronal functions (e.g. receptors, ion channels). The two sets form together a set $\mathbf{G} =\{ G_1, G_2, \ldots, G_n\}$. The model does not know the absolute values of gene expression levels therefore it works with the relative changes in their expression.

A change in expression level of each gene $g_j(t+\Delta t) \in (-\infty, \infty)$ is a function of the changes in gene expression levels of the rest of the genes in **G**(t). As the first simple model, it is assumed that this function is a linear function, i.e.:

$$g_j(t+\Delta t) = \sum_{k=1}^{n} w_{jk} g_k(t) \qquad (1)$$

The square matrix of gene connection weights **W** represents the GN, $w_{ij} \in$ (1, 1) (see Figure 6-6). The model assumes that when a gene is upregulated (i.e., $g_j(t)>0$) more protein defined by this gene will be produced in the neuron, and vice versa, i.e. when the gene is down regulated (i.e., $g_j(t)<0$) less protein coded by this gene will be produced.

Neuronal functions (neuron's parameters) **P** = $\{P_1, P_2, \ldots, P_m\}$ from a neural network model are related to particular proteins, so that each parameter P_j is a function of the expression of several (or in partial case – one) genes. For simplicity in the model it is assumed that one parameter P_j depends only on one gene such that:

$$P_j(t+\Delta t) = P_j(0) g'_j(t+\Delta t) \qquad (2)$$

where $g'_j(t+\Delta t) \in [0, \infty)$ is a relative change in the protein concentration against its initial concentration at time 0, based on the change in its gene expression level. It is calculated as

$$g'_j(t) = s\left(g(t)\right) \tag{3}$$

where *s* is a squashing function (see e.g. Figure 6-6) and *g(t)* is determined by equation (1).

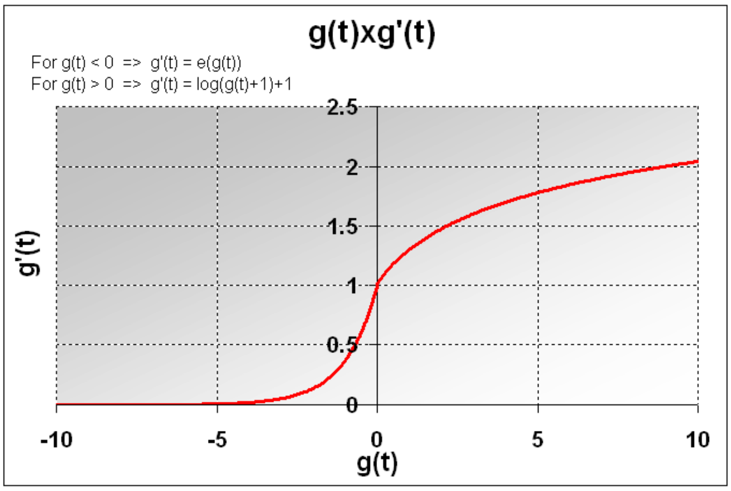


Figure 6-6 - An example of a nonlinear function to obtain relative changes in parameters values (equation (3)) according to the current changes in gene expression levels g(t).

This generic CNG model can be run step by step over time in the following way:

1.Define the initial changes in expression values of the genes in the neuron, **g**(t = 0), and the matrix **W** of the GN if that is possible. Set the initial values of SNN parameters, **P**(t = 0).

2.Update the GN and define the next state of the gene vector **g**(t+Δt) using equation (1).

3.Derive the values of the parameters **P** from the gene state **g**(t+Δt) using equation (2).

4.Evaluate the spiking activity of neuron(s) (taking into account all external inputs to the neural network).

5.Go to step 2.

The model makes several simplifying assumptions:

1. Each neuron has the same GN – in terms of same genes and same network matrix **W**.
2. Each GN starts from the same initial value of the gene expressions.
3. Each individual GN is synchronized with others, i.e. Δt is equal for all genes and for all GNs.
4. There is no feedback from neuronal activity to gene expression level.

6.4.1. Determination of the GN transition matrix *W*

The biggest challenge of the approach followed and the key to the predictions of the model is the construction of the GN transition matrix **W**, which determines the dynamics of GN and consequently the dynamics of the NN. There are several ways to obtain **W**:

(1) Ideally, the values of gene interaction coefficients w_{ij} are obtained from real

measurements through reverse engineering performed on the microarray data. At present, there are very limited experimental data available for the brain.

(2) The values of **W** elements are iteratively optimised from initial random values, for instance with the use of genetic algorithms, to obtain the desired behaviour of the NN. The desired behaviour of the NN can simulate certain brain states like epilepsy, schizophrenic hypofrontality, learning and memory disorders (after incorporation of synaptic learning). This behaviour would be used as a "fitness criterion" in the genetic algorithm to stop the search process for an optimal interaction matrix **W**.

(3) The matrix **W** is constructed heuristically based on some assumptions and insights into what result is intended to obtain and why. For instance, the model can use the theory of discrete dynamic systems to obtain a dynamic system with the fixed point attractor(s), limit cycle attractors or strange attractors.

(4) The matrix **W** is constructed from known facts and literature on gene-protein interaction such as those acquired from the brain-gene ontology.

(5) The matrix **W** is constructed with the use of a mix of the above methods.

6.4.2. Model defined GN corresponding to the desired NN behaviour

The above generic model allows the investigation and discovery of relationships between different GNs and NN states. A procedure to obtain this relationship can read:

1. For an initial GN state, generate a GN matrix **W**;
2. For the matrix **W** run the NN model over a period of time T and record the activation of the neurons in the NN;
3. Evaluate characteristics of the NN behaviour (e.g. total activation, spectral characteristics);
4. Compare the NN characteristics to the characteristics of the desired NN state (e.g. epilepsy);
5. Repeat steps (1) to (4) until a desired GN and NN model behaviour is obtained;
6. Analyse the GN and the NN parameters for significant gene patterns that cause the NN model behaviour.

The generic model above is illustrated in the next sections on a simple CNG model of a spiking neural network (SNN) with an optimised GN.

6.5. A CNG model of a spiking neural network (SNN) – model description

The model is designed as a small network of spiking neurons, N = 120. Inside of each neuron a small gene network (GN) affects the values of neuron parameters in a dynamic fashion (Figure 6-7). Each neuron parameter is in reality linked to a particular protein (receptor, ion channel, enzyme, etc.) the concentration of which is determined by the expression level of the corresponding gene(s).

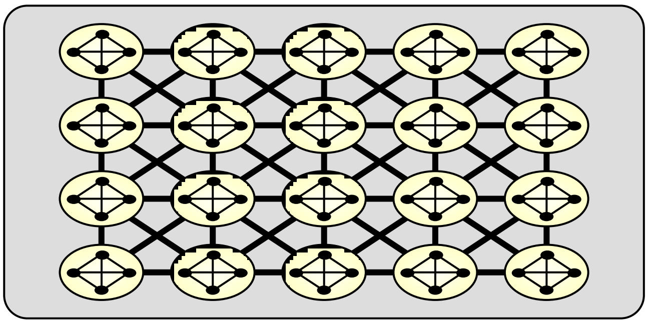


Figure 6-7 - Neurons of the model neural network have a gene network operating within them that affects the values of their parameters in a dynamic fashion.

The neuron spiking model is derived from the spike response model (SRM). The total somatic postsynaptic potential (PSP) of a neuron i is $u_i(t)$. When $u_i(t)$ reaches the firing threshold $\vartheta_i(t)$, neuron i fires, i.e. emits a spike (see Figure 6-8). The moment of $\vartheta_i(t)$ crossing defines the firing time t_i of an output spike. The value of $u_i(t)$ is the weighted sum of all synaptic PSPs, $\varepsilon_{ij}(t-t_j-\Delta_{ij}^{ax})$, such that:

$$u_i(t) = \sum_{j\in\Gamma_i}\sum_{t_j\in F_j} J_{ij}\varepsilon_{ij}(t-t_j-\Delta_{ij}^{ax}) \tag{4}$$

The weight of synaptic connection from neuron *j* to neuron *i* is denoted by J_{ij}. It takes positive (negative) values for excitatory (inhibitory) connections, respectively. Δ_{ij}^{ax} is an axonal delay between neurons *i* and *j*, which linearly increases with Euclidean distance between neurons. The positive kernel expressing an individual postsynaptic potential (PSP) evoked on neuron *i* when a presynaptic neuron *j* from the pool Γ_i fires at time t_j has a double exponential form, i.e.

$$\varepsilon_{ij}^{synapse}(s) = A^{synapse}\left(\exp\left(-\frac{s}{\tau_{decay}^{synapse}}\right) - \exp\left(-\frac{s}{\tau_{rise}^{synapse}}\right)\right) \tag{5}$$

where $\tau_{decay/rise}^{synapse}$ are time constants of the fall and rise of an individual PSP, respectively, *A* is the PSP's amplitude, and *synapse* denotes one of the following: *fast_excitation, fast_inhibition, slow_excitation, and slow_inhibition*. These types of PSPs are based on neurobiological data. Immediately after firing the output spike at t_i, neuron's firing threshold $\vartheta_i(t)$ increases *k*-times and then returns to its resting value ϑ_0 in an exponential fashion:

$$\vartheta_i(t - t_i) = k \times \vartheta_0 \exp\left(-\frac{t - t_i}{\tau_{decay}^{\vartheta}}\right) \tag{6}$$

where τ_{decay}^{ϑ} is the time constant of the threshold decay. In such a way, absolute and relative refractory periods are modelled. External inputs from the input layer are added to the right hand side of (4) at each time step. Each external input has its own weight $J_i^{ext\_input}$ and $\varepsilon_i^{fast\_excitation}(t)$, i.e.

$$u_i^{ext\_input}(t) = J_i^{ext\_input}\ \varepsilon_i^{fast\_excitation}(t) \qquad (7)$$

It was employed a random input with the average firing frequency of 15 Hz.

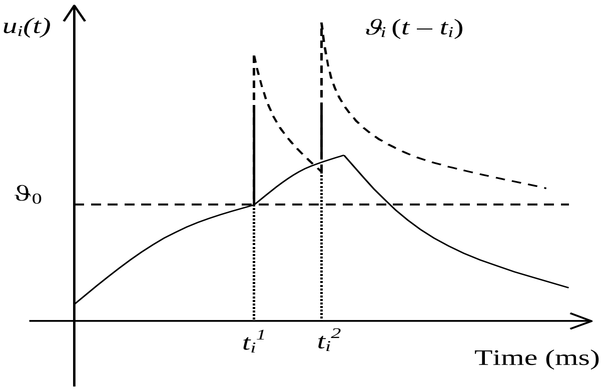


Figure 6-8 - Spiking neuron model. When the state variable ui(t) of a spiking neuron reaches the firing threshold ϑi(t) at time ti, the neuron fires an output spike. Current firing threshold rises after each output spike and decays back to the initial value.

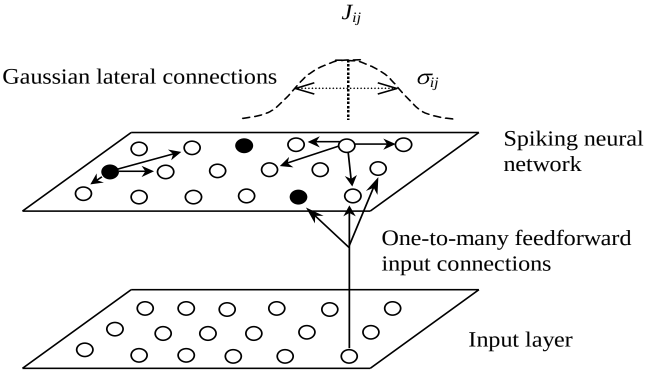

J_{ij}
Gaussian lateral connections
σ_{ij}
Spiking neural network
One-to-many feedforward input connections
Input layer

Figure 6-9 - SNN architecture. Filled circles denote inhibitory neurons.

Figure 6-9 illustrates the architecture of our spiking neural network (SNN). Spiking neurons within the network can be either excitatory or inhibitory. There can be as many as about 10–20% of inhibitory neurons positioned randomly on the rectangular grid of *N* neurons. Lateral connections between neurons have weights that decrease in value with distance from neuron *i* according to a Gaussian formula while the connections between neurons themselves can be established at random. There are one-to-many feed-forward connections from the input layer decreasing in strength according to the Gaussian distribution Table 6-3 and Table 6-4 contain the values of neuron's and SNN parameters, respectively that were used in our preliminary simulations.

Table 6-3 – Neuron's Parameters

GENE	**Neuron's parameter**	**Value**
G1	Amplitude of fast excitation	4
G2	Fast excitation rise / decay time constants (ms)	2 / 5
G3	Amplitude of slow excitation	1
G4	Slow excitation rise / decay time constants (ms)	20 / 50
G5	Amplitude of fast inhibition	1
G6	Fast inhibition rise / decay time constants (ms)	5 / 10
G7	Amplitude of slow inhibition	3
G8	Slow inhibition rise / decay time constants (ms)	50 / 100
G9	Resting firing threshold	19.5
G10	Decay time constant of the firing threshold (ms)	30
	Number of times the threshold is increased k	2

Table 6-4 – SNN Parameters

SNN parameter	**Value**
Number of neurons	**120**
Proportion of inhibitory neurons	**0.2**
Probability of external input fiber firing	**0.015**
Peak/sigma of external input weight	**5 / 1**
Peak/sigma of lateral excitatory weights	**10 / 4**
Peak/sigma of lateral inhibitory weights	**40 / 6**
Probability of connection	**0.5**

Table 6-3 also contains relations between neural parameters and hypothetical genes in our GN. For instance, amplitudes of PSPs would be related to the concentration of receptor-gated ion channels in the postsynaptic membrane. The time constants of PSPs would be related to the properties of individual receptor-gated ion channels. Concentrations and properties of proteins are determined by coding genes.

6.5.1. Some preliminary experimental results

Figure 6-10 illustrates the field potential of the SNN model and its spectral characteristics for an optimised GN matrix. **W** has been optimised to yield the spectral characteristics of Φ as similar as possible to the spectral characteristics of normal state EEG. The coefficients of **W**, $w_{ij} \in (-1, 1)$ was generated, such that the modulus of the maximal (complex) eigenvalue was equal to 1. That means the stable oscillatory behavior in the Lyapunov sense of the corresponding linear gene dynamic system (see Figure 6-11 c). The initial values of parameters are listed in Table 6-3 and Table 6-4 Parameter changes were applied after each 1000 ms of SNN simulation according to the equation (2).

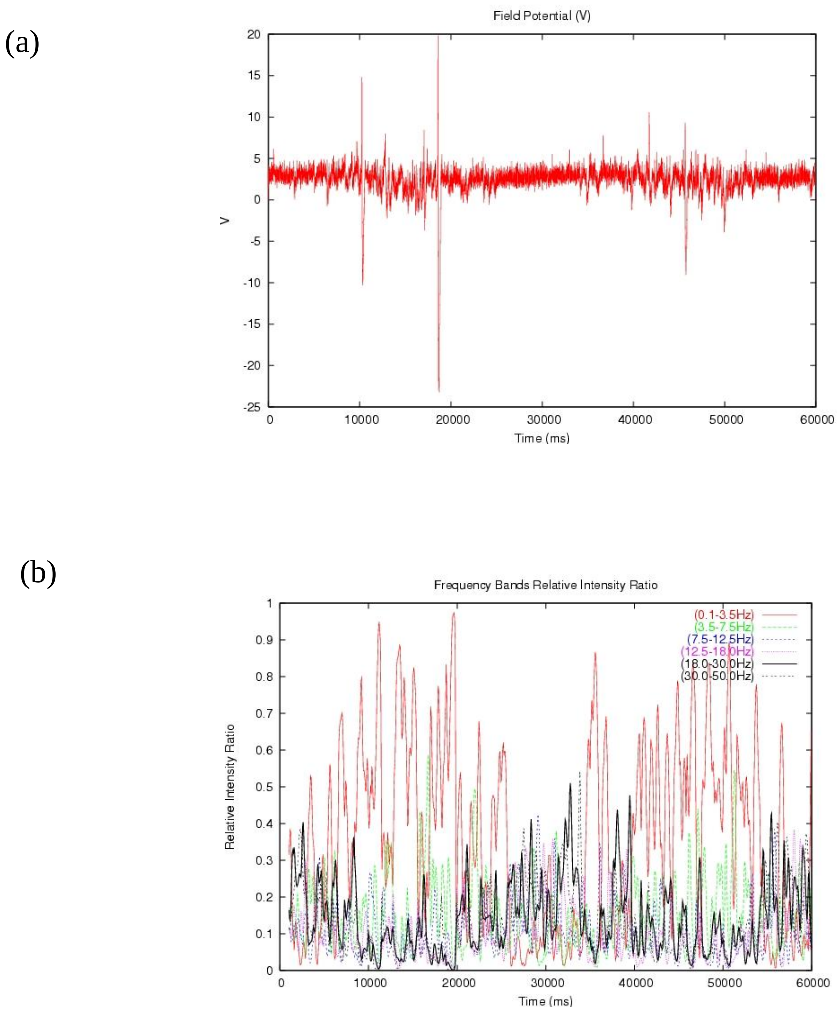


Figure 6-10 - Time evolution of the field potential of the SNN with dynamic parameter values. (a) Field potential. (b) Spectral characterisation. Sampling rate = 1000 Hz, Min/Max frequency = 0.1 / 50 Hz, respectively. The dominant frequency band is delta (0.Ĩ3.5 Hz).

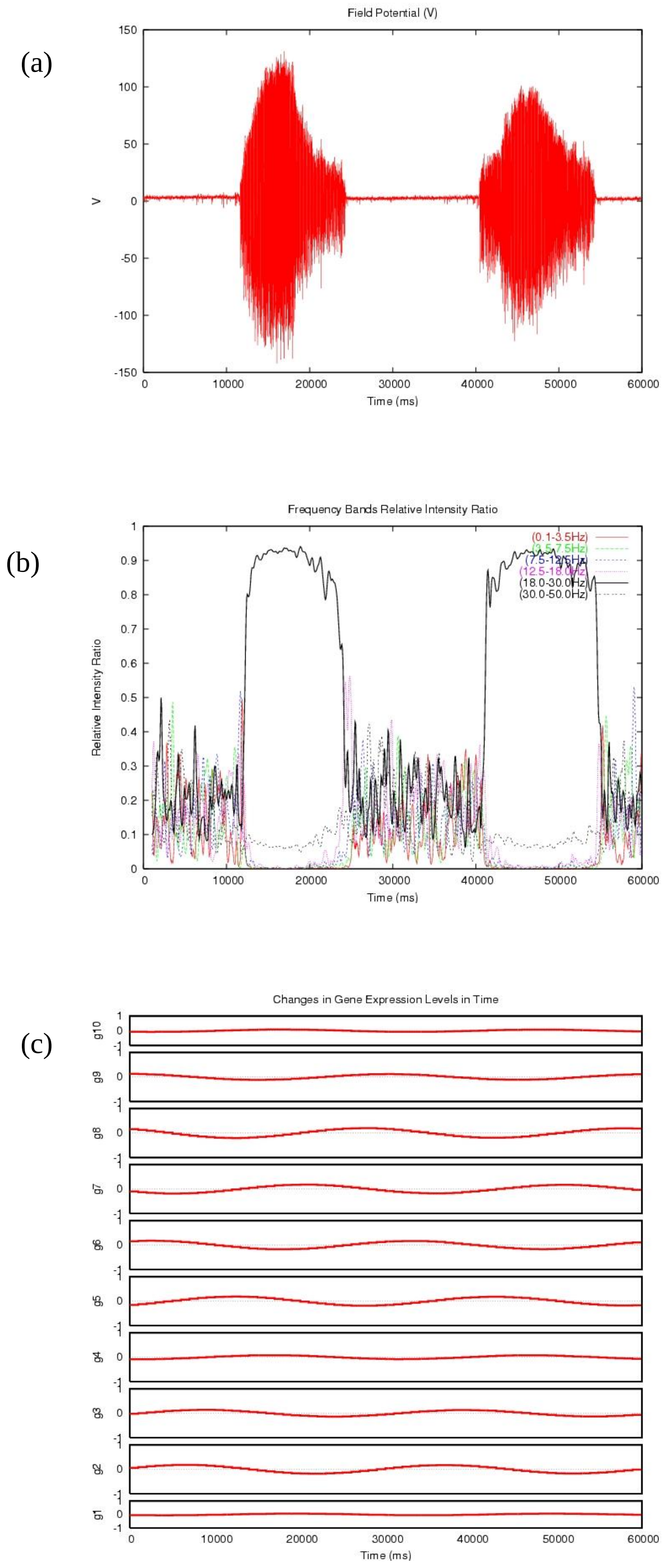

(a)
Field Potential (V)
150
100
50
0
-50
-100
-150
V
0
10000
20000
30000
40000
50000
60000
Time (ms)
(b)
Frequency Bands Relative Intensity Ratio
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Relative Intensity Ratio
(0.1-3.5Hz)
(3.5-7.5Hz)
(7.5-12.5Hz)
(12.5-18.0Hz)
(18.0-30.0Hz)
(30.0-50.0Hz)
0
10000
20000
30000
40000
50000
60000
Time (ms)
(c)
Changes in Gene Expression Levels in Time
g10
g9
g8
g7
g6
g5
g4
g3
g2
g1
1
0
-1
0
10000
20000
30000
40000
50000
60000
Time (ms)

Figure 6-11 - (a, b) Different initial values of parameters can lead to totally different behaviour of SNN. (c) Corresponding relative changes in gene expression levels for a single optimised interaction matrix W, the same as in Figure 6-10. Zero means no change in the expression level.

Then it is kept the same **W** optimised for the normal state EEG and experimented with different initial values of parameters $P_j(0)$ in equation (2). Figure 6-11 illustrates the field potential Φ of the SNN with initial conditions simulating the temporal lobe epilepsy (TLE) with tonic-clonic (TC) seizures. Namely, the slow inhibition was disabled by putting $A^{slow\_inhibition} = 0$ and the fast inhibition was enhanced by putting $A^{fast\_inhibition} = 10$. In Figure 6-11 a, b shows that the behavior of the network has completely changed. Now, the neural dynamics leads to transient global synchronizations with frequency characteristics very similar to the EEG of TC seizures for which the dominant frequency is beta 2 (18–30 Hz) and the EEG amplitude rises to hundreds of μV. It is worth to mention that with the fixed values of parameters, it is not possible to obtain transitions from the normal state to the prolonged periods of global synchronizations and back as seen in Figure 6-11. It seems that the internal dynamics of genes linked to neural parameters is needed to account for observed transitions in neural network states. Analysis of which genes and which parameters are crucial for these state transitions can bring new insights into the etiopathogenesis and treatment of the disease.

While the capability to simulate certain brain states based on the interaction of genes and biologically plausible neural networks is valuable, the parameters of a simulation and its results must be biologically validated in order to contribute to the advance of studies in both neuroscience and bioinformatics.

The integration of the previous knowledge, by means of the Brain Gene Ontology, and newly discovered knowledge, by means of simulation, is then a much-desired feature, which is covered in the next section. It is also important to note that several validation exercises were done previously within the research group (Kasabov, Jain, Gottgtroy, Benuskova, & Joseph, 2007) to both validate the ontological model and the initial results of the simulations. These studies guided the selection of the CNGM as a modelling technique to develop a full ontology driven tool based on the ODKD framework developed in this thesis.

6.6. A Biomedical Knowledge Discovery Tool – BGO System

The biomedical knowledge discovery tool is an extension of the ODKD Framework designed to cope with the requirements of Computational Neuro Genetic Modelling.

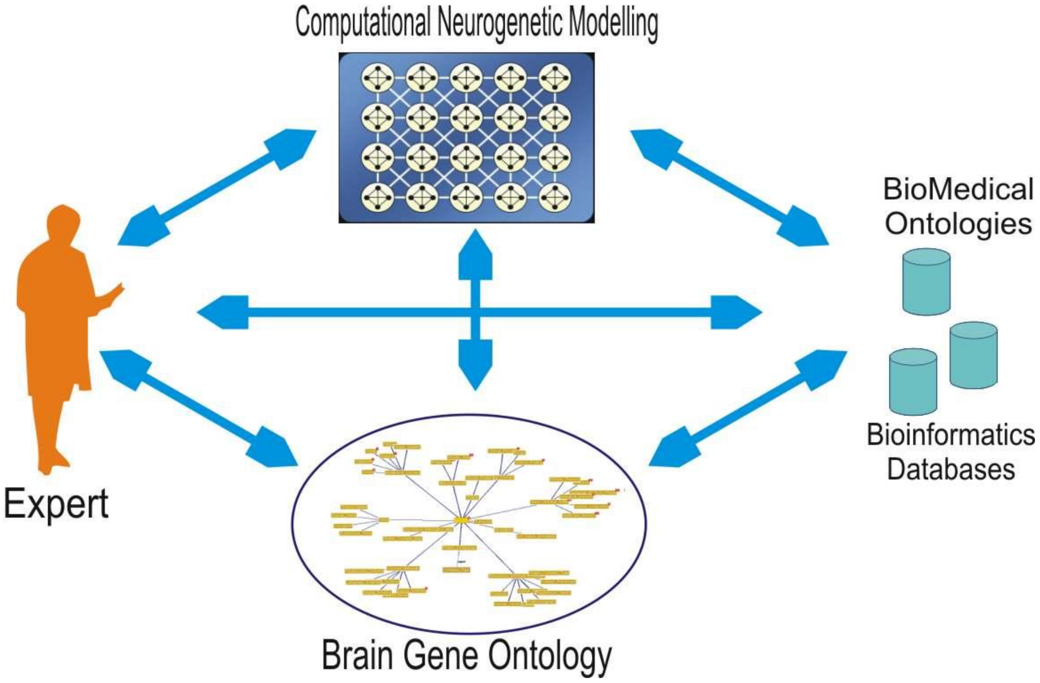


Figure 6-12 – The biomedical knowledge interaction schema.

As shown in Figure 6-12, the ontological model acquires knowledge from experts, biomedical ontologies, bioinformatics databases, and from the CNGM simulator. To this end, it includes a set of plug-ins specially designed and developed in this research to deal with biomedical knowledge sources, gene regulatory networks (GRN), and with the results of a CNGM simulation.

A gene regulatory network or genetic regulatory network (GRN) is a collection of DNA segments in a cell which interact with each other (indirectly through their RNA and protein expression products) and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA.

The biomedical tool is composed of three main components: the Brain Gene Ontology, the Gene Regulatory Network Visualization (GRN Graph Widget) and the gene regulatory network knowledge acquisition tool (Gene Regulatory Network Tab).

The ontology is the central repository of all knowledge related to genes and brain diseases. It was developed following the ODKD process and methodology. The GRN Graph widget is the novel visualization tool developed in this thesis that stores gene regulatory network data on the ontological model. The Gene Regulatory Network tab is responsible for the acquisition/editing/integration of CNGM simulations which was newly developed to incorporate regulatory network data.

Appendix B presents a series of screenshots and more detailed information on each of these components, describing their functionalities and so forth.

6.6.1. CNGM simulation results

The CNGM simulation results are the product of an optimisation of an interaction matrix **W** between genes, initial values of neural parameters, architectural parameters of a spiking neural network model (SNN) (except the total number of neurons, spike delays and probability of establishing a synaptic connection), and input frequency to the SNN.

The main goal of the simulation is to compare the output of the spiking neural network model with targeted real human brain EEG signals and to consequently be able to generate optimal artificial gene regulatory networks for hypothesis checking and validation by experts and for comparison with previous knowledge by means of the Brain Gene Ontology.

The simulation keeps a record of spiking activities of all neurons as well as a record of the local field potential (LFP). This comparison is based on evidence that brain LFPs in principle have the same spectral characteristics as EEG .

A number of gene interaction matrices Ws that show almost a uniform distribution of interaction strengths between genes are presented as an example in Figure 6-13. The small histograms show the percentage distribution of positive and negative interaction weights between genes, which are in this case equally probable. With a uniform generator of numbers from the interval (–5, 5), the probability of generating zero is very small.

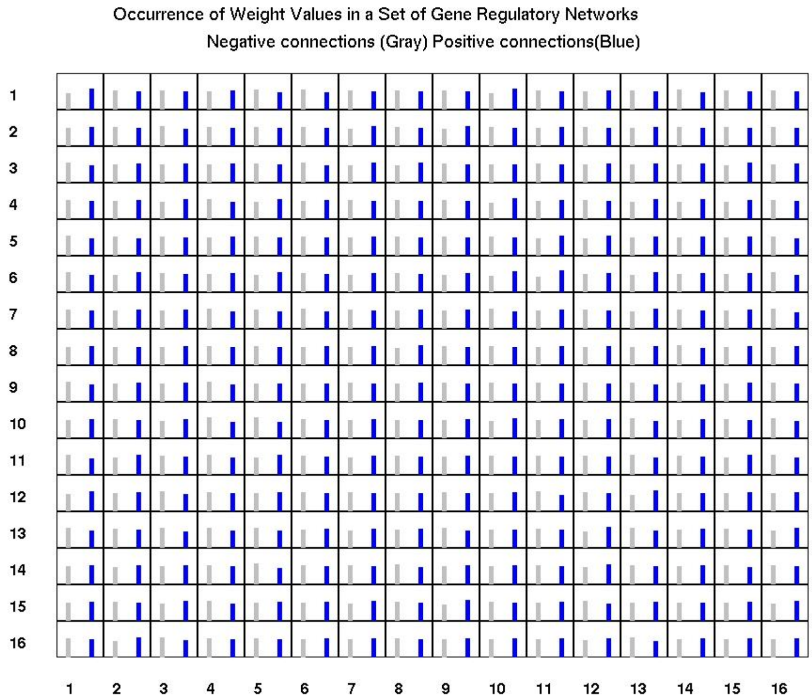


Figure 6-13 – Blue and grey histograms show the percentage of positive and negative gene interactions, respectively, in 400 randomly generated interaction matrices W.

Among the 400 random solutions shown in Figure 6-13, 15 Ws matrices lead to an SNN LFP with spectral characteristics very close to the target EEG signal in terms of Euclidean distance between the characteristic vectors being smaller than 0.1. To discover the knowledge, e.g. to find out what these selected solutions have in common, how many times the interactions between genes become positive and how many times they become negative is calculated (Figure 6-14). A basic frequency statistical analysis test can then be used, for instance the X2-statistic, to make predictions about interactions between real genes in neurons. Figure 6-14 outlines gene groups that are directly related to information-processing parameters of neurons.

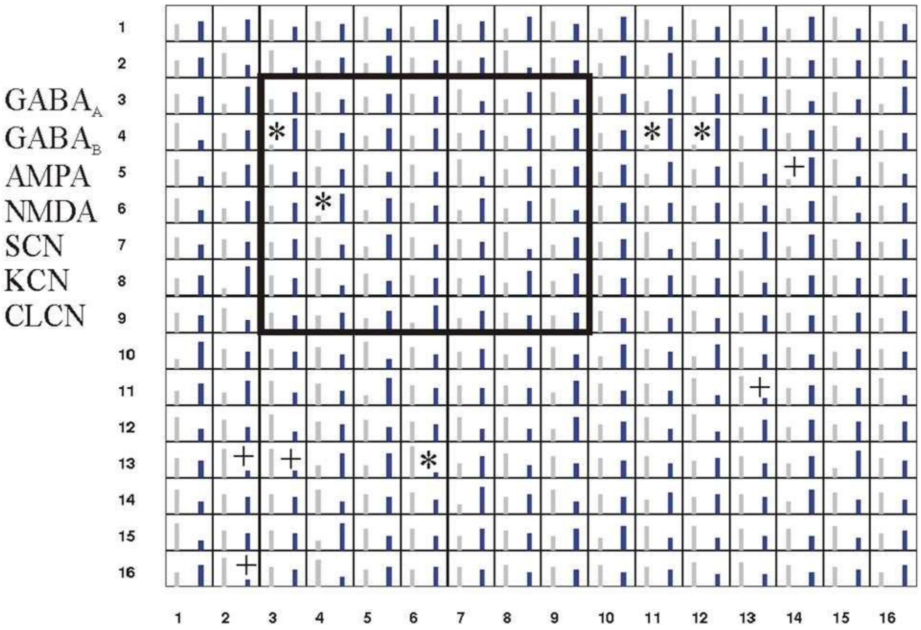
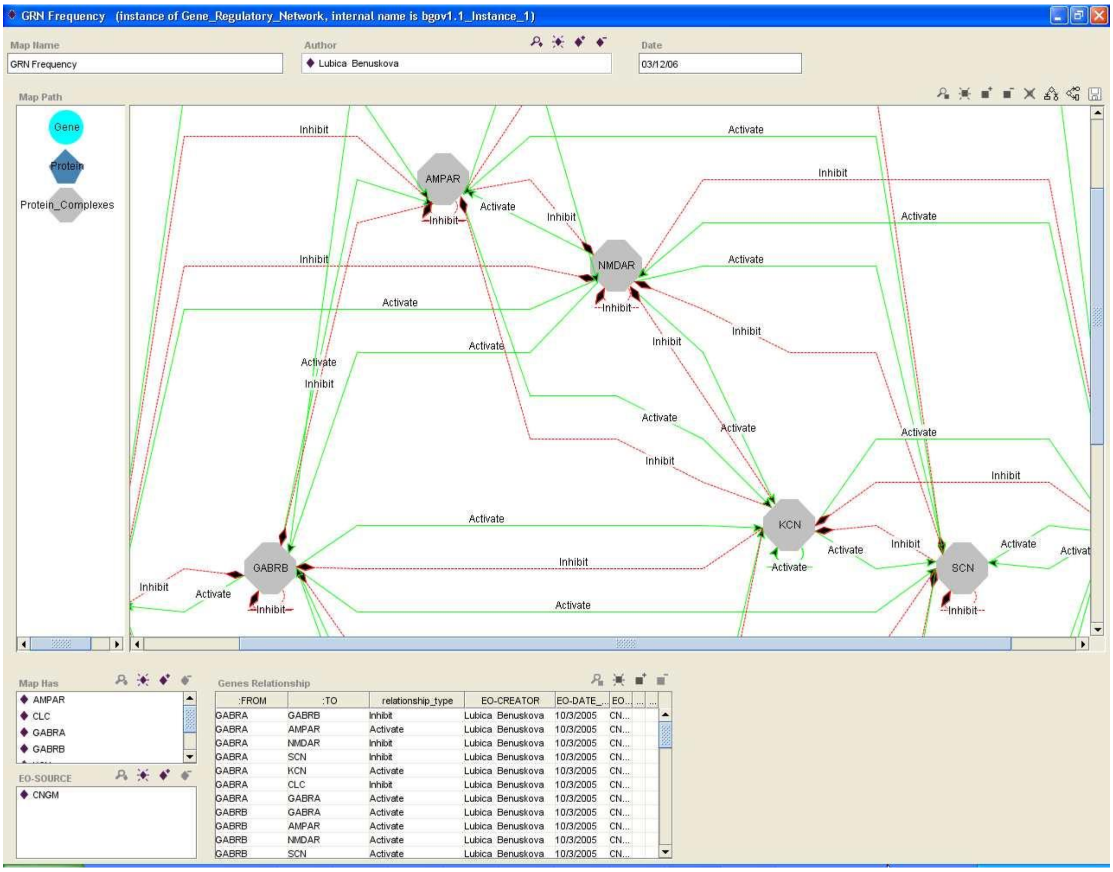


Figure 6-14 – Black and grey histograms show the percentage of positive and negative gene interactions, respectively, in 15 winning interaction matrices after experimental running of CNGM.

This frequency analysis, along with the biological knowledge of the gene groups, which are directly related to information-processing parameters of neurons, are the driving force of the hypothesis about the interactions between real genes within neurons. This information and the associated hypotheses can then be considered and linked to the knowledge represented in the BGO as shown in Figure 6-15.



GRN Frequency (instance of Gene_Regulatory_Network, internal name is bgov1.1_Instance_1)

Map Name: GRN Frequency

Author: ◆ Lubica Benuskova

Date: 03/12/06

Map Path

Map Has
- ◆ AMPAR
- ◆ CLC
- ◆ GABRA
- ◆ GABRB

EO-SOURCE
- ◆ CNGM

Genes Relationship

:FROM	:TO	relationship_type	EO-CREATOR	EO-DATE_...	EO...	...	...
GABRA	GABRB	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	AMPAR	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRA	NMDAR	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	SCN	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	KCN	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRA	CLC	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	GABRA	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	GABRA	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	AMPAR	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	NMDAR	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	SCN	Activate	Lubica Benuskova	10/3/2005	CN...		

Figure 6-15 – BGO System: A screenshot of a GRN graphical representation of the gene regulatory network presented in Figure 6-14 and Table 6-5.

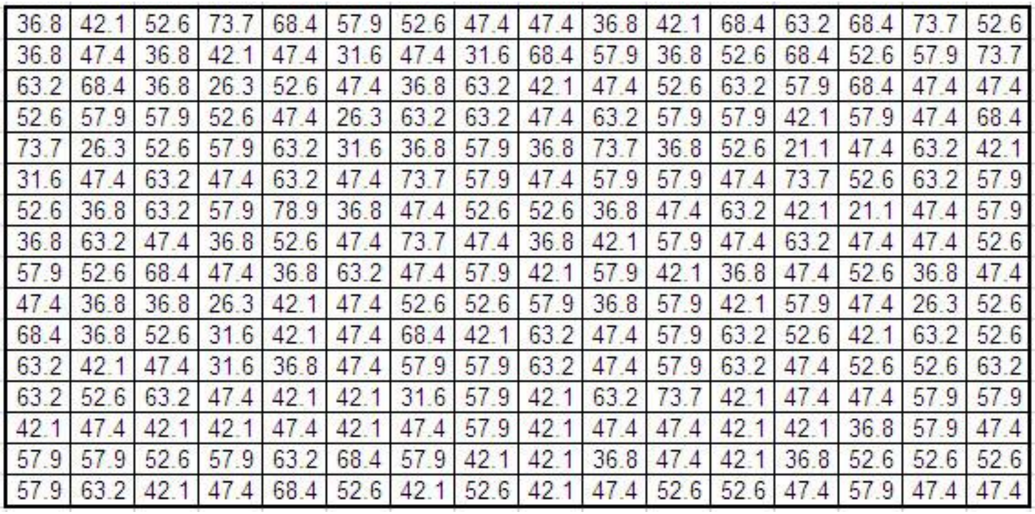
The GRN graph is linked to the ontology by the Gene Regulatory Network concept, which is a sub-class of Bioinformatics Maps within the Brain Gene Ontology. All concepts and relationships allowed are specified in the ontology in accordance with the knowledge needed to represent gene regulatory networks. This ontological representation makes possible the visualization and navigation of all concepts related to the network.

Although designed to cope with the specific requirements of visualizing gene regulatory networks, the *GRN Graph* was designed to support the Protégé user community by maintaining compatibility with the graph widget plug-in (Protege, 2006b). It extends the graph widget by adding link analysis features to it as well as exploring advanced network features such as nodes, shapes, connectors and text configuration (depicted in Appendix B Figure B-8).

6.6.2. Importing Gene Regulatory Network data

The GRN tab is used for the creation and editing of gene regulatory networks. It includes a set of features that are primarily designed to import the results of a computational neuro-genetic simulation such as the example presented in Table 6-5.

Table 6-5 – Gene Expression Matrix acquired from simulation referenced in Figure 6-15



36.8	42.1	52.6	73.7	68.4	57.9	52.6	47.4	47.4	36.8	42.1	68.4	63.2	68.4	73.7	52.6
36.8	47.4	36.8	42.1	47.4	31.6	47.4	31.6	68.4	57.9	36.8	52.6	68.4	52.6	57.9	73.7
63.2	68.4	36.8	26.3	52.6	47.4	36.8	63.2	42.1	47.4	52.6	63.2	57.9	68.4	47.4	47.4
52.6	57.9	57.9	52.6	47.4	26.3	63.2	63.2	47.4	63.2	57.9	57.9	42.1	57.9	47.4	68.4
73.7	26.3	52.6	57.9	63.2	31.6	36.8	57.9	36.8	73.7	36.8	52.6	21.1	47.4	63.2	42.1
31.6	47.4	63.2	47.4	63.2	47.4	73.7	57.9	47.4	57.9	57.9	47.4	73.7	52.6	63.2	57.9
52.6	36.8	63.2	57.9	78.9	36.8	47.4	52.6	52.6	36.8	47.4	63.2	42.1	21.1	47.4	57.9
36.8	63.2	47.4	36.8	52.6	47.4	73.7	47.4	36.8	42.1	57.9	47.4	63.2	47.4	47.4	52.6
57.9	52.6	68.4	47.4	36.8	63.2	47.4	57.9	42.1	57.9	42.1	36.8	47.4	52.6	36.8	47.4
47.4	36.8	36.8	26.3	42.1	47.4	52.6	52.6	57.9	36.8	57.9	42.1	57.9	47.4	26.3	52.6
68.4	36.8	52.6	31.6	42.1	47.4	68.4	42.1	63.2	47.4	57.9	63.2	52.6	42.1	63.2	52.6
63.2	42.1	47.4	31.6	36.8	47.4	57.9	57.9	63.2	47.4	57.9	63.2	47.4	52.6	52.6	63.2
63.2	52.6	63.2	47.4	42.1	42.1	31.6	57.9	42.1	63.2	73.7	42.1	47.4	47.4	57.9	57.9
42.1	47.4	42.1	42.1	47.4	42.1	47.4	57.9	42.1	47.4	47.4	42.1	42.1	36.8	57.9	47.4
57.9	57.9	52.6	57.9	63.2	68.4	57.9	42.1	42.1	36.8	47.4	42.1	36.8	52.6	52.6	52.6
57.9	63.2	42.1	47.4	68.4	52.6	42.1	52.6	42.1	47.4	52.6	52.6	47.4	57.9	47.4	47.4

The next items describe briefly the importing procedure (see Figure 6-16):

1. CNGM generates the GRN matrices **W** with gene expression values;
2. The matrices are imported using the Gene Regulatory Network Tab. Each simulation is imported and respective metadata is annotated using EO-metadata;

3. Each simulation creates a gene regulatory network map instance;
4. Each gene presented in the GRN is linked to its GO-term (Gene Ontology instance) using the unified name or any of the synonyms current available in the ontology. Biomedical data is then automatically attached to the GRN data.
5. A biomedical relationship is created between each pair of genes and its evidence and gene expression is annotated.
6. A link analysis feature is added to the relationship to demonstrate the strength or importance of the relationship. The link analysis then adds biomedical relationship metadata to the newly created relationship.
7. After the biological annotation of each gene and regulatory network data a visualization of the network is created using the GRN visualization widget developed in this research.
8. The visualization and related biological data is then added to a knowledge acquisition tool which is then made available for further analysis and annotation by users such as domain experts and researchers.

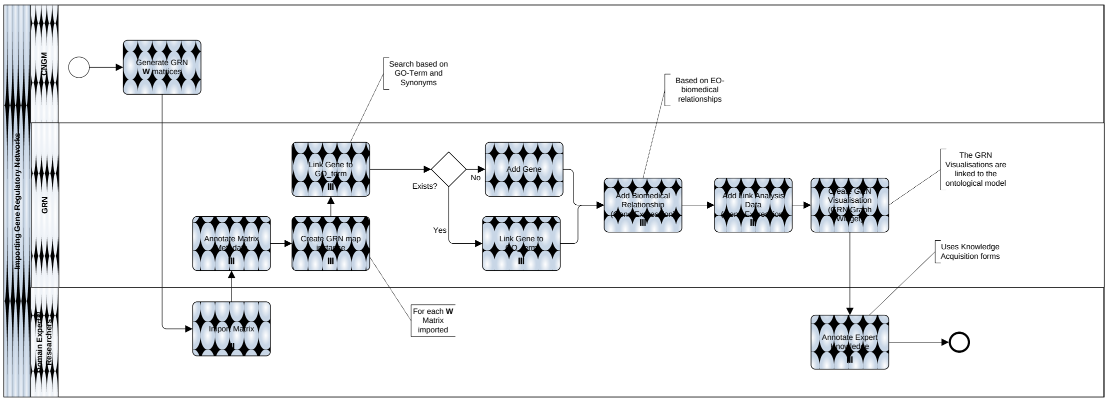

Importing Gene Regulatory Networks
CNGM
GRN
Domain Experts/ Researchers
Generate GRN W matrices
Search based on GO-Term and Synonyms
Based on EO-biomedical relationships
Link Gene to GO_term
Exists?
No
Yes
Add Gene
Link Gene to GO_term
Annotate Matrix Metadata
Create GRN map instance
Add Biomedical Relationship (Gene Expression)
Add Link Analysis Data (Gene Expression)
Create GRN Visualisation (GRN Graph Widget)
The GRN Visualisations are linked to the ontological model
Uses Knowledge Acquisition forms
Import Matrix
For each W Matrix imported
Annotate Expert Knowledge

Figure 6-16 –A process flow to import GRNs.

The GRN tab also enables the editing of a regulatory network by adding new genes, proteins, and protein complexes, or including information about the network such as annotations and uncertainty properties of a relationship. This knowledge acquisition feature is provided through the *GRN Graph Widget* interacting with the knowledge base and all Protégé's editing functionalities. A series of screenshots illustrate the importing procedure is shown in Appendix B.

The table bellow shows a generic algorithm of the process.

```
FOR each Matrix W
    Annotate provenance metadata using EO-Metadata (Creation, EO-Source, etc)
    Create GRN biomedical map instance
    FOR each Gene in the matrix W selected
        IF GO-Term or Synonym exist THEN
            Link Gene to GO-Term
        ELSE
            Add new Gene
        ENDIF
        Add Ontological Knowledge
    ENDFOR
    FOR each gene relationship
        IF gene expression is positive THEN
            Create an activation relationship
        ELSE
```

```
            Create an inhibition relationship
        ENDIF
        Update expression value attribute (link analysis)
    ENDFOR
    Create GRN Visualisation
ENDFOR
```

The matching of a GO-term and/or its synonym is performed using the Protégé API capability by matching strings and the unique internal name attributed to each instance created in the knowledge model. This mechanism then supports the identification of an existent gene or the necessity for the creation of a new one.

6.7. Summary

The challenge of making diverse biomedical knowledge and concepts sharable over applications and reusable for several purposes is both complex and crucial. It is central to enabling comprehensive knowledge-acquisition by medical research communities and molecular biologists involved in biomedical discovery.

This chapter describes the integration of the Brain-Gene Ontology (BGO) with the Computational Neuro-Genetic Modelling simulator (CNGM) in the development of a biomedical knowledge discovery tool. It complements the practical contribution of this thesis and provides verification of the functionality of the system developed as part of the multi methodological research methodology.

The biomedical knowledge discovery tool is based on the Ontology Driven Knowledge Discovery Framework and includes an additional set of plug-ins built to integrate with the Computational Neuro-Genetic Model and simulations as well as with regulatory networks.

The BGO acts as a biological knowledge repository which guides the construction of simulations to test biomedical hypothesis. The simulations generate a series of plausible gene regulatory networks that need to be biologically verified by experts using the knowledge base. The experts can also use other sources of information present in the BGO to complement the knowledge inserted in the ontological model. This hypothesis-driven process enables the creation of an accurate knowledge base of genetic information about brain diseases. All of this information can be re-utilized to create further models of brain function and disease that include models of gene interactions.

The biomedical knowledge discovery tool allows users to navigate through the rich information space of brain functions and brain diseases, brain related genes and their activities in certain parts of the brain and their relation to brain diseases; to run simulations; to download data that can be used in a software machine learning environment; to visualize relationship information; and to add new information to the knowledge base.

The tool has the capability to facilitate active learning and research in the areas of bioinformatics, neuroinformatics, information engineering, and knowledge management.

The biomedical knowledge discovery tool also respects all criteria established early in the research. Each ODKD Framework criterion was respected in the development of the tool achieving the following results:

- The system supports all phases of the Ontology Driven Knowledge Discovery process and is based on the widely-accepted and used Protégé ontology editor;

- The system reuses the evolving ontology meta-data and meta-classes to build the concepts related to the gene regulatory network;

- The system extends the ODKD Framework and maintains a flexible architecture which copes specifically with the biomedical knowledge discovery domain, enabling, for example, the use of the plug-ins in other contexts such as the educational environment (Kasabov, Jain, Gottgtroy, Benuskova, & F. Joseph, 2007);

- The GRN tab has an interface that enables the engineer/user to navigate in and edit the ontological model;

- All the tools developed were published as independent and reusable modules in the Protégé open source library.

The creation of the BGO was facilitated by both the ODKD methodology and process, and the set of tools especially developed to incorporate gene regulatory data into the Ontology. The phased approach followed in the research methodology was also facilitated by the three-fold strategy adopted in the development of the case study: *development of tools*, *enhancement of the body of knowledge* and *simulations*, which in consequence reduced the complexity of developing the case study.

Investigation of the results in light of domain knowledge triggered the necessity of sourcing new data, which in consequence tested the framework's capacity to acquire the data from both a modelling perspective (increasing the maturity level of the conceptual model) and supporting tools (adding more requirements for the development of new tools). The knowledge acquired then defined new parameters to be simulated by the CGNM tool. In consequence a clear requirement set was defined to support the incorporation of the simulations results, finally closing the loop of the analysis.

Chapter 7

Evaluation of Research

This chapter evaluates the contribution of the research taking into consideration the *research methodology* and the *system requirements* established previously in Chapter 1 (Research Design). It verifies the satisfaction of all requirements by reviewing the Design Science and the System Development methodologies adopted in this research.

7.1. Introduction

This research aimed to advance core knowledge in the ontology engineering and knowledge discovery in databases fields by designing, implementing and evaluating an ontology-driven knowledge discovery framework which encompasses a methodology, a meta-knowledge model and software engineering tools to semantically support knowledge discovery in databases.

As *applied research,* rather than basic research, this thesis has two main goals, one theoretical and one practical (section 1.3): "*to understand why things happen in a particular context*", and "*to improve practice by conducting research that will ultimately yield usefulness and improve processes in a particular field*" (Adams & Courtney, 2004). The achievement of these goals is evaluated in the following two sections: Design Science and System Development.

The Design Science section evaluates the methodology objectives (Figure 7-1) by reviewing the conceptual framework and life cycle developed in this research. The System Development section links Nunamaker's software development methodology and the specified system requirements with the research conducted in order to demonstrate *process improvement* and *usefulness* by the systems built from a requirement analysis perspective.

Experiments are summarized at the end of each chapter and published outcomes are listed in Appendix D in order to demonstrate the iterative development process as well as the maturity of the research.

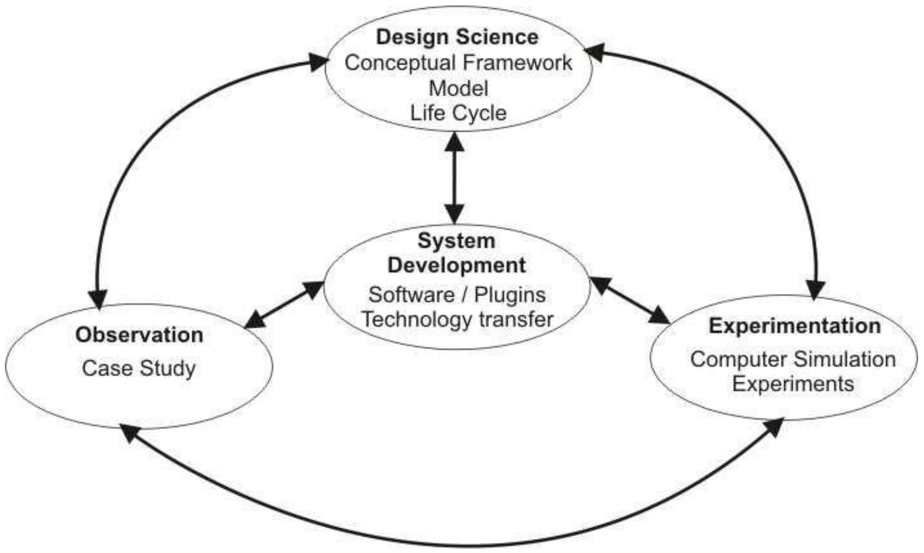


Figure 7-1 – The multi-methodological approach used in this thesis(adapted from Nunamaker et al. and Adams et al.)

7.2. Design Science

As stated in Chapter 1, Design Science is highly suited to applied research as it aims “*to solve problems by introducing into the environment new artefacts”*. Design science is used in this work as part of the multi-methodological research methodology (see Figure 7-1). Design Science is directed at the creation of models (Chapter 3) and processes (Chapter 4) as shown in Figure 0-2. It encompasses the *research proposal* and the *conceptual contribution* of this research. This section mainly covers the conceptual contribution of the novel Evolving Ontology meta-knowledge model (EO) and the definition of the Ontology Driven Knowledge Discovery process (ODKD).

The development of the Evolving Ontology meta-knowledge model, in the context of Knowledge Discovery in Databases (KDD), contributes theoretically to both ontology and KDD research by taking into consideration the need for “knowledge evolution” and the notion of *partial shared conceptualization*.

Partial here means incomplete knowledge, being constructed, dynamic, changing, and evolving. *Shared* means consensual knowledge, maintained through knowledge management. The *Evolving Ontology* is then able to capture the intrinsic consensual conceptual structure of a domain while considering the notions of incompleteness, uncertainty, high dimensionality and other specific perspectives presented in the cyclic knowledge discovery in databases process.

The Evolving Ontology (EO) increases knowledge (theory) by creating a model able to represent evolution in terms of *change management.* It contributes to the understanding of *change* in the context of the dynamics of the hybrid knowledge discovery process embedded in the ODKD.

The EO *quarantine structure* also adds a time dimension to the ontology knowledge acquisition problem. It extends the theoretical discussion by creating a meta-class able to annotate knowledge acquired manually, semi-automatically, and/or automatically and allows its evaluation by experts and/or inference mechanism. It then contributes directly to knowledge validation.

The *change management* and *temporal* requirements discussed and supported by the Evolving Ontology meta-knowledge model also reflect the philosophical aspects of knowledge evolution (section 3.3). This philosophical discussion, in the context of the Ontology Driven Knowledge Discovery process (ODKD), contributes to the ontology engineering field by bringing a novel dimension and proposal to ontology evolution and evaluation research.

The Ontology Driven Knowledge Discovery process contributes to the knowledge engineering field by creating a hybrid methodology and process model which maps KDD processes to an ontology building process. The proposed ontology-driven life cycle considers and leverages the best practices of industry and academia.

The *conceptual contribution*, enveloping the Evolving Ontology Meta-knowledge model and the Ontology Driven Knowledge Discovery Process, is the basis for the development and implementation of the ODKD Framework set of tools. The *Conceptual Model* (EO and ODKD) is functionally able to enhance the process of knowledge discovery from data by adding a high level of abstraction to the process which may enable better reuse of previous knowledge as well as the creation and evaluation of new knowledge.

Conceptualization or the establishment of the theoretical grounding of system requirements is suggested in (Adams & Courtney, 2004) as "*the focal point of the research effort*". This research's conceptual contribution has led to increased knowledge around the integration of ontology engineering and knowledge discovery in databases as well as the notion of ontology evolution.

The introduction of the novel Evolving Ontology model and ODKD then helped to generate an increased understanding of the ontology change problem in the context of KDD as well as improved practice through the creation of a methodology and process model that improved processes in the knowledge discovery field. It also, as suggested in the previous paragraph, established the focal point of the research by defining the requirements for the development of the Ontology Driven Knowledge Discovery Framework discussed in the following System Development section.

7.3. System Development

System Development concerns *theory testing*, and allows a realistic technological evaluation of the product developed and its potential for acceptance. This research followed Nunamaker's systems development process, Figure 7-1, as a foundation for the evaluation of the "*usefulness"* and "*interestingness"* of the Ontology Driven Knowledge Discovery Framework as indicated by the applied research objective defined previously (section 1.3.1).

The following evaluates the research methodology reviewing Nunamaker's principles (section 1.3.1) in relation to the chapters of the thesis:

- "*Design is the most important part of a system development process*."- The system design involved the understanding of the studied domain (Chapters

1) and the application of relevant scientific and technical knowledge for creation of a solution (Chapters 3, 4, and 5).

- "*A good system architecture provides a road map for the systems building process. It puts the system components into the correct perspective, specifies the system functionalities, and defines the structural relationships and dynamic interactions among system components*." – The system architecture is defined in Chapter 5, section 5.2.1. It constitutes the basis for the development of the ODKD implementation framework.

- "*Researchers must identify the constraints imposed by the environment, state the objectives of the development efforts (i.e. the focus of the research), and define the functionalities of the resulting system to achieve the stated objectives.*" - Defined in Chapter 1, section 1.4. This is expanded upon and evaluated in the next sub-section under *System Requirements*.

- "*Building a prototype system always helps to study and to understand a research domain*"- Chapters 6 describes a prototype of a biomedical knowledge discovery tool developed in this thesis, along with the insights that were generated using that tool.

- "*The process of implementing a working system can provide researchers with insights into the advantages and disadvantages of the concepts, the frameworks, and the chosen design alternatives.*" - An iterative development process was followed in this thesis. Early development such as the Brain Gene Ontology, for example, revealed new requirements and suggested modifications in both the EO meta-knowledge model and ontology plug-ins. This iterative process helped to refine the design of the ODKD Framework as well as the meta-knowledge model.

- "*Depending on the focus of the research, one might emphasize the new functionalities or innovative user interface features of the proposed new system rather than the throughput or the response time of the system.*"- Chapter 6 addresses all new functionalities and interfaces developed based on the requirements of the biomedical knowledge discovery domain.

- "*Implementation of a system is used to demonstrate the feasibility of the design and the usability of the functionalities of a system development research project.*"– Chapters 5 and 6 describe the set of tools developed so as to demonstrate the feasibility of the design. The iterative development process has helped in the refinement of the tools and design of the prototype.

- "*Once the system is built, researchers can test its performance and usability as stated in the requirement definition phase, as well as observe its impacts on individuals, groups, or organizations.*"- Chapters 6, 7 and 8 (the Future research section) illustrate the prototype in use in a case study and in further research using and extending the methodologies and tools developed in this thesis.

- "*The test results should be interpreted and evaluated based on the conceptual framework and the requirements of the system defined at the earlier stages.*" – The definition of system requirements (Chapter 1) and their extension by the iterative development process is the basis for the evaluation described in the next sub-section.

The next sub-sections list all requirements defined in Chapter 1 and outline the characteristics of the models, processes and systems developed that meet the requirements.

7.3.1. Knowledge Representation

As indicated in section 1.4.2, the Evolving Ontology Meta-knowledge model should be able to:

- *Define different sources of information* – the ontological model is composed of an evolving meta-knowledge able to integrate multiple knowledge sources, as shown in the case study undertaken (Chapter 6).

- *Maintain a traceable link to original sources* – Each concept in the ontology model keeps provenance of at least source, date and creation (Chapter 3).

- *Represent uncertainty* - meta-class relations can implement different uncertainty measures, such as evidence by literature or quality of source (Chapter 3).

- *Acquire new knowledge while establishing degrees of knowledge acceptance based on external measures* – every concept/instance is annotated in the ontology. The gene expression acquired from the gene regulatory network is used as external measures when acquiring data from the CNGM simulations. Further measures, such as the number and quality of annotations can also be used to define levels of acceptance if necessary (Chapters 3 and 6).

- *Annotate any concept with domain specific and general meta-models* – every concept can be annotated by various sources, such as experts, external databases, and so on (Chapters 3 and 6).

7.3.2. Conceptual Framework

As indicated previously in section 1.4.2, the conceptual framework should be able to integrate both ontology engineering and knowledge discovery in databases processes. The framework should be able to:

- *Reuse knowledge* - the current brain gene ontology exemplifies this requirement by reusing widely used knowledge such as the Gene Ontology (Chapter 6).

- *Support different data mining tasks* - the framework is based on the CRISP-DM process which covers all tasks within a data mining task (Chapter 5). Chapter 6 describes the case study which followed the ODKD methodology.

- *Learn ontology using data mining techniques* – the prototype was tested in closed-loop fashion using Computational Neuro-Genetic Modelling. The system is also able to export ontology instances to WEKA and Neucom systems, and to virtually any system which accepts flat files as input (Chapters 5 and 6).

- *Define a hybrid life cycle* – the ODKD process integrates the best practices of the ontology engineering and KDD fields (Chapter 5).

- *Use a navigation, visualisation and query language to support the hybrid life cycle* – The framework supports all three navigation and search methods (Chapters 5 and 6).

7.3.3. Ontology Driven Methodology

As indicated previously, the methodology should establish a reusable process as well as be able to reuse and integrate different tools. The methodology should be able to:

- *Integrate different databases* – there are several import features available, such as the OBO tab (Chapter 5 and 6).

- *Use manual, automatic and semi-automatic knowledge acquisition tools* – the environment allows for manual knowledge acquisition. The biomedical tool has both automatic and manual knowledge acquisition features (Chapter 6).

- *Follow a hybrid ontology-driven KDD life cycle* – ODKD is based on the CRISP-DM and best academic practices (Chapter 5).

- *Suggest a conceptual modelling technique* - the ODKD is both a process and methodology that can adopt different conceptual modelling techniques. The brain gene ontology design, for example, was initially based on Significant Conceptual Modelling (Gottgtroy, 2000) and later extended as a Masters thesis (Wang, 2007) using a hybrid ontology building modelling technique.

7.3.4. Ontology Engineering Tool

As specified in the related section in Chapter 5, the ontology engineering tool developed as part of the ODKD Framework should be able to:

- *Support the whole hybrid life cycle* – a set of newly developed, extended and existing Protégé plug-ins is included in the framework to support the entire ODKD life cycle (Chapter 5).

- *Include navigation, visualization and query support* – all three features are supported (Chapter 5 and 6).

- *Integrate different data mining workbenches* - the novel Instance Selection Tab is able to export instances of interest to several data mining workbenches (Chapters 5 and 6).

- *Be extensible* – the system is based on the extensible Protégé architecture.

- *Be reusable* - the system is based on an object oriented approach and on the open Protégé architecture which enables reusability.

- *Be published in the public domain* – the plug-ins are open source and will be widely published after the examination of the thesis.

- *Integrate and reuse the best tools and practices in the ontology engineering field* – the system is based on the current, most-used, freely-available, ontology environment – Protégé.

7.4. Additional Assessment

The aim of the research has always centred on the development of a system and processes to yield "usefulness" and "process improvement" in the ontology engineering field (section 1.3). Such an aim is aligned with the research philosophy of Adams & Courtney (2004) and was adopted in both the research design and in its evaluation following Nunamaker's systems development process. That said, the solution can also be assessed against other criteria commonly utilised in system evaluations - performance and usability - and against other candidate solutions via comparative analysis. These assessments now follow.

7.4.1. Performance Assessment

There is no performance restrictions in regard to the meta-knowledge model, methodology and processes developed in this research. The methodology and model are independent of performance issues related to knowledge base size. The meta-knowledge model may be impacted by the knowledge representation formalism, for example, source annotations are perhaps easier to implement using RDF or OWL (ontology web language) as representation formalisms, but this in itself is not a constraint on performance.

However it should be noted that the performance of the ontology environment may be impacted by the memory restriction imposed by the java virtual machine when considering the time to load and query a knowledge base. These restrictions were avoided in this research by tuning the prototyping environment to reach the best performance in terms of memory and reading access. In general, there is a 2 GB limitation to run the java virtual machine and load data into memory (1.6 GB on Windows and 2GB on UNIX machines). This limitation is perceived when reading knowledge bases from text-based storage. This is also applied to the user interface when loading and changing a knowledge base. However, this can be easily fixed by tuning the memory configuration, selecting a relational database as ontology backend storage and accessing the knowledge base directly through the API.

This performance issue has not impacted on the biomedical discovery tool developed since the amount of data used and the data mining technique employed did not stress the system. The Protégé community has worked with knowledge bases comprising more than 100k frames with acceptable performance. Although this reported threshold is far from the current knowledge base tested and deployed in this research (approximately 1K frames and instances), the next version of the system, described in the future research section, may be deployed using a relational database backend or a RDF store based on the Oracle 10g RDF data store which has proved to achieve excellent performance and is under constant development by Oracle and semantic technology partners.

7.4.2. Usability Assessment

There are two main points when considering usability assessment in regards to the prototype: ODKD coverage and user interface.

ODKD coverage is related to the system’s ability to support all phases of the ODKD process model. The coverage is explained in detail in Chapters 5 and 6 as well as in the Appendices through extra screenshots of the tools. Therefore it is not covered further in this section. However coverage can also be considered from the point of view of extensibility and applicability; in other words the ability to be used in different domains, tasks and scenarios.

Even though the design and implementation of the functionalities were directed to the biomedical domain and more specifically targeted to the brain gene case study, from a system point of view, the tools developed or extended in this research are all generic implementations which can be assembled or used in different problem domains. Even the features specifically designed for biomedical informatics, such as the GRN widget, were also made available as generic link analysis tools able to be used in different scenarios (such as the educational scenario described in the next chapter).

With respect to the user interface and its usability, this was not formally evaluated in light of the fact that, in keeping with the stated requirements of the research (Chapter 1 and 5), the Protege environment was selected as the context for implementation. The research therefore adopted the open source development best practices enabled by Protégé's core functionalities, and deployed them in a manner consistent with Protégé's user experience. The newly developed and adapted components all employed standard metaphors/mechanisms e.g. trees, graphs, grids/tables and tabs, so the cognitive burden on the user of the tools should not be any greater than that experienced in 'regular' use of Protégé.

Improvements in *usefulness* were made by identifying the need for extension of some core functionalities and plug-ins, as described previously. Enhancements and improved user experience were provided through the development of novel functionalities and plug-ins. For example, the Instance selection tab covers more than one phase of the ODKD and at the same time simplifies the user experience by adopting an interface that aggregates several functionalities while still following Protege's user experience premises. The introduction of new features required by the

Protege community for some time, such as the introduction of link analysis features to the graph widget, might be considered as an example of novelty that does not compromise usability.

7.4.3. Comparative Analysis

The integration of knowledge discovery and ontology has been investigated by different fields from several perspectives, such as in the use of ontology to guide clustering algorithms in data mining tasks, and in the use of ontology to model social entities in social network analysis.

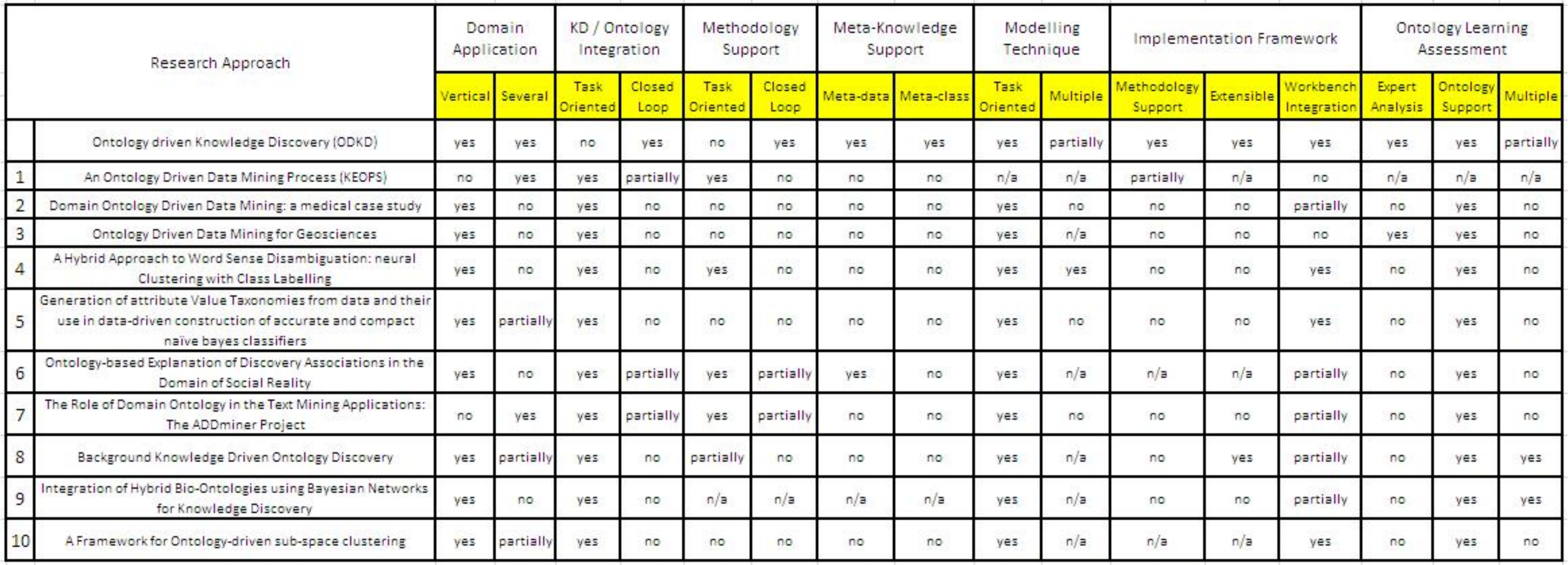
The knowledge discovery in databases community in particular has investigated the use of ontology with three different emphases: Ontology for KDD, KDD for ontology and Closed loop integration. There are numerous research efforts that fall in the first two categories as reported in the literature and presented in Chapter 4. However few research endeavours can be classified as investigating the closed loop integration of ontology and knowledge discovery in databases.

This section therefore uses a framework analysis, which considers several characteristics relevant to the closed loop integration of knowledge discovery in databases and ontology engineering, to compare some of these different research efforts. This comparative analysis is not exhaustive and cannot be claimed to be complete.. It is, however, indicative of the state of capability in regard to the functionalities required in closed loop integration. This section then should be considered as providing a brief map between some of the approaches followed in the literature and the research undertaken in this thesis.

The analysis is based primarily on research published in the first and second International Workshop on Knowledge Discovery and Ontologies (KDO 2004 and 2005), the International Workshop on Domain Driven Data Mining (DDDM '07) and papers referenced as related to ontology driven knowledge discovery. These workshops are part of the major conferences in the field such as the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining and the European Conference on Machine Learning and the European Conference on Principles and Practice of Knowledge Discovery in Databases.

The first five approaches (Brisson & Collard, 2006; Kang, Silvescu, Zhang, & Honavar, 2004; Legrand & J., 2004; Tadepalli, Sinha, & Ramakrishnan, 2004; Yen-Ting, Andrew, Liz, & Kathy, 2007) (1-5 in Table 7-1) are referenced in the literature as related to the integration of ontologies and KDD. The next two (Garcia, Ferraz, & Pinto, 2006; Svatek, Rauch, & Flek, 2005) (6 and 7), are not intended to cover the full ontology engineering and KDD processes, but they recognise the need for integration in their publications. The last three (Chen, Alahakoon, & Indrawan, 2005; Jinze, Wei, & Jiong, 2004; McGarry & Wermter, 2007) (8, 9 and 10), are good examples of the specific application of ontology in KDD tasks.

Table 7-1 – Comparative Analysis



	Research Approach	Domain Application		KD / Ontology Integration		Methodology Support		Meta-Knowledge Support		Modelling Technique		Implementation Framework			Ontology Learning Assessment		
		Vertical	Several	Task Oriented	Closed Loop	Task Oriented	Closed Loop	Meta-data	Meta-class	Task Oriented	Multiple	Methodology Support	Extensible	Workbench Integration	Expert Analysis	Ontology Support	Multiple
	Ontology driven Knowledge Discovery (ODKD)	yes	yes	no	yes	no	yes	yes	yes	yes	partially	yes	yes	yes	yes	yes	partially
1	An Ontology Driven Data Mining Process (KEOPS)	no	yes	yes	partially	yes	no	no	no	n/a	n/a	partially	n/a	no	n/a	n/a	n/a
2	Domain Ontology Driven Data Mining: a medical case study	yes	no	yes	no	no	no	no	no	yes	no	no	no	partially	no	yes	no
3	Ontology Driven Data Mining for Geosciences	yes	no	yes	no	no	no	no	no	yes	n/a	no	no	no	yes	yes	no
4	A Hybrid Approach to Word Sense Disambiguation: neural Clustering with Class Labelling	yes	no	yes	no	yes	no	no	no	yes	yes	no	no	yes	no	yes	no
5	Generation of attribute Value Taxonomies from data and their use in data-driven construction of accurate and compact naïve bayes classifiers	yes	partially	yes	no	no	no	no	no	yes	no	no	no	yes	no	yes	no
6	Ontology-based Explanation of Discovery Associations in the Domain of Social Reality	yes	no	yes	partially	yes	partially	yes	no	yes	n/a	n/a	n/a	partially	no	yes	no
7	The Role of Domain Ontology in the Text Mining Applications: The ADDminer Project	no	yes	yes	partially	yes	partially	no	no	yes	no	no	no	partially	no	yes	no
8	Background Knowledge Driven Ontology Discovery	yes	partially	yes	no	partially	no	no	no	yes	n/a	no	yes	partially	no	yes	yes
9	Integration of Hybrid Bio-Ontologies using Bayesian Networks for Knowledge Discovery	yes	no	yes	no	n/a	n/a	n/a	n/a	yes	n/a	no	no	partially	no	yes	yes
10	A Framework for Ontology-driven sub-space clustering	yes	partially	yes	no	no	no	no	no	yes	n/a	n/a	n/a	yes	no	yes	no

n/a – not applicable.

The comparison above can be interpreted from different perspectives. The paragraphs that follow present two such perspectives covering two sets of features: Domain Application/Metadata Support and Implementation Framework/Ontology Learning Assessment.

The first comparison of Domain Application/Metadata Support addresses the following questions: Does the approach focus on a specific domain? As a consequence, does it determine, or constrain, the coverage of the application? Paper number 4 (A Hybrid Approach to Word Sense Disambiguation: Neural Clustering with Class Labelling) focuses on word disambiguation. Therefore its research is concentrated on taxonomy building. The approach taken in paper number 7 (The Role of Domain Ontology in the Text Mining Applications: The ADDminer Project) focuses on the role of domain knowledge. It then considers the taxonomy aspect of the support, as in paper 4, plus all other aspects related to domain ontologies application. However, due to limitations on their scope, neither study explores the role of other metadata support for KDD.

Further approaches have also been investigated more recently, for example, "Ontology management and evolution for business intelligence" published in the *International Journal of Information Management* (2010) by Mikroyannidis and colleagues from Manchester Business School. In this case they consider a series of ontologies and layer them to address some of the wider application of ontologies for information management. Since their main interest is on data sharing, however, they do not explicitly consider the application of wider meta-knowledge. Therefore metadata support for decision-making and closed loop integration is not explored. Their work is therefore useful for a particular aspect of the problem space but does not deal with the ‘bigger picture’ issues that this research has worked towards.

Another example of a particular comparison may be drawn when considering Implementation Framework and Ontology Learning Assessment together.

The research reported in paper number 9 (Integration of Hybrid Bio-ontologies Using Bayesian Networks for Knowledge Discovery) presents a very effective task-oriented approach. The authors have developed a workbench tightly coupled to a particular data mining technique but also use domain and application ontologies to extract networks from text and linking these to the Gene Ontology. While the approach lacks support for a hybrid and closed loop methodology it could be easily integrated with the ODKD.

These few examples illustrate again that the domain area and research focus can and do play a major role in the development of specific solution steps for the integration of ontology engineering and KDD.

The outcomes of this brief analysis are not unexpected as closed loop integration of ontology engineering and KDD is still a young and evolving field of research. The work described in this thesis was focused on the methodology, meta-knowledge and foundation implementation of such integration. Other approaches, such as that followed in the KEOPS implementation, focus on knowledge extraction. Any deep comparison between these approaches may not even be appropriate and would lack consistency as the number of open problems is still great. However, it seems reasonable to say that the ODKD approach makes a novel contribution. Furthermore, these efforts may converge through research collaboration and through the use of these methods and tools by a wide community. This should bring together some of these techniques in novel ways in the future, such as those described in the next chapter.

7.4.4. Related Work

This section presents some examples of current work using the methodologies and tools developed in this research. A list of works referencing this research is also briefly presented.

Several researchers have utilized the work developed in this thesis at different levels. For example, the paper "Considering Application Domain Ontologies for Data Mining" by Pinto et al (2009) published in *Transactions on Information Science and Applications*, reports how thay have taken the framework and meta knowledge and extended it using OWL and SWRL. This research focuses on the use of business rules and descriptive logics to support ontology integration in KDD. It is part of a wider approach for database marketing. This approach could even be further extended by using some of the techniques developed subsequently in our research.

Another related research effort has focused on the use of the toolset to acquire and analyse biomedical knowledge. Vishal Jain developed his PhD thesis (Integrative approaches to modelling and knowledge discovery of molecular interactions in bioinformatics) utilizing the toolset developed in this research to enable the acquisition and analysis of molecular interactions. The Kedri Brain Gene Ontology which is derived from this research is also utilized in both teaching and research at AUT University.

Several other related research endeavours that have cited the work reported here are listed in Appendix D.

7.5. Conclusion

This chapter evaluated the work reported in this thesis from a research design perspective. The requirements defined in Chapter 1 were compared to the achievement of research outcomes. Development of the thesis followed fully the research design and covered all requirements specified previously.

Although outside the core scope of this research, the ontology environment was briefly analysed from a performance, usability and comparative analysis perspective. Further comments on this aspect of the system are made in the future research section of the next Chapter.

The next chapter addresses the general contribution of the thesis and more specific topics such as technology transfer and future research.

Chapter 8

Contributions / Future Research

This chapter concludes the thesis by giving a brief summary of the research, stating the key features of the Ontology Driven Knowledge Discovery framework and emphasising the research contributions. It then considers the limitations of the research imposed by scope, the need for generality and technology. The final section presents ongoing and future research based on the outcomes of this thesis that has been conducted in the Knowledge Engineering and Discovery Research Institute and through technology transfer to industry.

8.1. Introduction

This thesis was motivated by the Conceptual Biology and Theoretical Biology challenges presented in *Nature* by Blagoskolonny and Perdee (2002), and Bray (2001) respectively. The authors argue in these publications that the explosion of biomedical data and the growing number of disparate data sources present opportunities to build a shared knowledge repository capable of transforming the current data collection era into one of hypothesis–driven and experimental research by using models and tools to design and test small genetic circuits in theory in order to understand living cell interactions.

This research, based on these challenges, and on the ongoing capabilities in all aspects of computer science, aimed to advance core knowledge in the integration of the ontology engineering and knowledge discovery in databases fields by designing, implementing and evaluating an ontology driven knowledge discovery framework (ODKD) able to support the above requirements.

This novel ontology driven knowledge discovery framework encompasses a methodology, a meta-knowledge model and software engineering tools based on the best practices in industry and academia in both ontology engineering and KDD fields.

In the next sections key features and contributions of this ontology driven framework are presented. Limitations to the work are then acknowledged. A comparative analysis of this work and that of others is then provided, in order to highlight the positioning of this research in the body of knowledge. This is followed by a description of future research and technology transfer.

8.1.1. Key features and Contributions

Brachman & Anand (1996) define knowledge discovery in databases (KDD) as a "*knowledge intensive iterative task consisting complex interactions, protracted over time, between human and a large database, possibly supported by a set of heterogeneous suite of tools"*. Being consistent with this concept of KDD, the ontology driven knowledge discovery framework is a conceptual model, a methodology and process and a 'suite of tools' that incorporate both iterative and interactive characteristics of KDD by means of an ontology driven KDD process.

The Evolving Ontology meta-knowledge model enables the representation of evolving knowledge and the tracking of changes in a shared repository. It enables the construction and integration of different sources of data and is capable of incorporating knowledge acquired by humans and by machine learning algorithms. It also creates a new semantic layer for data by adding a powerful metadata server to associated "data mining" exercises.

The methodology and process integrate both KDD and ontology engineering processes into a novel ontology driven knowledge discovery process. This new process goes towards the new generation of KDD tools proposed by Fayyad, Piatetsky-Shapiro, and Uthurusamy (2003) which combines the current vertical data mining solutions for different problem domains by reusing their findings in different knowledge discovery tasks.

The ODKD suite of tools implements a framework able to integrate the evolving ontology meta-knowledge and methodology to provide a more holistic view of the KDD process. It enables the engineer/user to select and combine several sources of information and apply the most suitable techniques for domain-specific problems in order to generate new hypotheses and build links across domains

The extensible system architecture enables the adoption of new knowledge, industry standards and tools and provides a solution architecture framework to deploy the ontology driven knowledge discovery process in different domains and software platforms, such as those described in the future research and technology transfer section later in this chapter.

The ontology driven knowledge discovery is novel in that it integrates both ontology engineering and KDD processes into one framework and a supporting methodology. It creates a new semantic structure, channel and process able to combine several sources of information and data mining tools within a shared knowledge repository. As such, it addresses the challenge of using computer models in concert with human knowledge to test hypotheses and to validate and integrate knowledge that may be created by different sources with diverse intentions but that when linked can promote the discovery of new knowledge.

The research concludes that the Ontology Driven Knowledge Discovery Framework is a flexible and versatile host for the implementation of a new generation of “data mining” workbench. The evaluation of the prototype and current use of extended versions of the ODKD framework developed in this thesis indicate that key novel and valuable functionalities have been produced.

The use of case studies in different stages of the research has helped to solidify the research and to engage knowledge experts throughout the process. This engagement was especially important since this thesis investigated different research fields and dealt with specific issues related to biology, philosophy, information systems, KDD and software engineering.

The ODKD framework thus provides a means of describing and representing evolving knowledge, managing shared knowledge, integrating data mining tools and algorithms, and enabling semantically rich knowledge discovery. It can be used in the design and implementation of ontology driven decision support systems, based on sound conceptual foundations in knowledge representation and knowledge discovery. The outcomes of this research are of potential value and importance and to a variety of different users and systems. The following sections summarise some important opportunities for use:

Ontology Engineers

The ODKD framework has been implemented to allow the ontology builder to:

- Create ontologies for a variety of modeling paradigms and for different domains;
- Modify ontologies while keeping track of change;
- Use existing ontologies and instances in the creation of more complex ontologies;
- Select instances by visualization, query and navigation to export in different formats and to several data mining tools.
- Acquire knowledge manually, semi-automatically and automatically from multiple sources and data mining tools.

Knowledge Experts

The framework provides a set of knowledge acquisition tools designed to enable a knowledge expert to enter and validate knowledge acquired by data mining tools and other knowledge experts. The knowledge expert can also explore the shared knowledge base by different means to link knowledge which was not known before. The meta-knowledge model allows the creation of knowledge maps which highlight similarities and differences in understanding. Future work may be used for case based reasoning and inference engine triggering.

Decision Makers

Decision makers can use the framework to make informed decisions based on the semantic repository and systems built on top of the knowledge base. The framework has been implemented to provide decisions makers with an environment that:

- Has strong knowledge management capability, dealing with provenance of source, creation, and so on;
- Can accommodate different modelling techniques;
- Implements industry standard methods and processes to support knowledge discovery tasks;
- Acts as a semantic integration organizational facility.

Developers

The framework provides an open source architecture that allows developers to:

- Extend the functionalities of the framework, such as adding a new export format or import different regulatory network formats;
- Reuse code;
- Maintain consistency with the original Protégé program interface;
- Recompile the code for future versions of the ontology environment;
- Integrate new data mining algorithms.

General Users

The framework allows general users to navigate complex knowledge structures by different means. Users can select different visualization techniques developed in this research or available from the Protégé community to suit their purposes.

The framework has been implemented as a general Protégé tab which limits the need for any further learning by Protégé users. It also uses the well established documentation provided by Protégé to engage new users and users outside the ontology research/community.

8.2. Research Limitations

This research concentrated on the development of a hybrid life-cycle and a suite of tools to support this cycle. It was focused on the development of a strategic and integrative view to enable the effective use of semantic technologies in the context of KDD.

Many specific topics considered in the Framework have been investigated by other researchers and are reported in the literature. All of those related specific topics were clearly referenced in the text, such as the several ontology merging techniques available in the literature. Others were further investigated as part of other studies aligned to this research, as referenced by the literature review in each chapter.

The biomedical application is a single (albeit comprehensive) case study. While it is believed that the framework is sufficiently generic to be applicable to other problems in other domains, this will need to be verified in independent research. (That said, it has since been used in other aligned research as described in the next section.)

Finally, the implementation framework presented in this thesis is fully usable and has been used by different researchers; however it is not intended to be a final system; a more complete version may be built in order to accommodate specific problem and domain requirements. Limitations in terms of quantity of techniques available, size of knowledge bases, and performance measures, out of the scope of the current research, would then need to be given greater consideration.

Even though Protégé is widely used in academia and in some research applications it cannot be considered as an industry-ready tool. The Protégé building tool has the capability to support the creation of infrastructural applications such as the ODKD. However, Protégé’s complex user interfaces, especially when considering the potential variety and number of end-users, may limit its ability to be used widely as an application platform.

These limitations did not impact on the development and evaluation of ODKD as reported here. However, industry tools such as TopBraid composer (an “extended” version of Protégé) could be used by others to overcome some of these limitations.

8.3. Future Research

The biomedical knowledge discovery tool developed as the case study of this thesis is the basis of ongoing research in the Knowledge Engineering and Discovery Research Institute. This demonstrates the importance, validity and viability of the ODKD framework and tool itself. Details of this ongoing work are as follows.

The Global Ontology Knowledge Bases for Biomedical and Bioinformatics Decision Support, for example, aims to extend the brain gene ontology case study developed in this thesis by adding different dimensions and diseases to the biomedical ontology and to utilise its complexity and richness to provide efficient profiling, prognosis, diagnosis and decision support for every individual person who needs it. This task will require both adaptive, evolving knowledge repository systems and methods for local and personalised modelling through tool/method integration and dynamic interaction.

The ontology will therefore need to represent brain knowledge in its multiple aspects of functioning and disease, at different levels. The processes that occur at each of these levels are very complex and difficult to understand, but much more difficult to understand is the interaction between the different levels, e.g. gene-brain function-disease. It may be that understanding the interaction through its modeling would be a key to understanding each level of processing and perhaps the brain as a whole (Benuskova & Kasabov, 2007).

In order to reason appropriately with these data, local and personalised models need to be developed and applied to the prediction of a person's risk or the likely outcome of a disease. The integration of these models will be based on the ontology driven knowledge discovery framework. The Machine inference module, that includes local and personalised techniques, will be integrated with NeuCom and other data mining workbenches as specified and developed in the implementation framework. Further data mining techniques will be integrated in the current system extending its capability.

The availability of new methods such as the integrative connectionist learning systems (ICOS) (Kasabov, 2008) that integrate in their structure and learning algorithms principles from different hierarchical levels of information processing in the brain including neuronal, genetic and quantum will open new opportunities to the integration of those methods in the ontology driven framework as well as bring new requirements for the representation of such complex knowledge.

A further major challenge currently being investigated is how to use newly discovered knowledge to further enhance existing ontologies. The current quarantine structure repository developed as part of the evolving ontology meta-knowledge model during the course of the research for this thesis comprises all metadata about newly acquired knowledge and current ontological knowledge. This knowledge supports inference and rank aggregation and will also be utilized to reason with data and will trigger relevant systems and/or tools.

Even though the current state of the quarantine structure suits the proposal of this research, it is evident that a rules engine could add great value to the framework by introducing a capability to infer new knowledge and also implement domain-specific heuristics.

Another application of a rules engine would be to manage the conflicts between knowledge discovered by the machine learner and facts stored in an ontology. Therefore the development of a new scoring engine to compare and assess knowledge generated by different algorithms and experts presents a great opportunity for research in this area which may also include the application of multi-agent systems.

The biomedical knowledge discovery tool will then be extended and different knowledge representation formalisms will be tested along with specific backend storage facilities such as RDF and Oracle 10g Spatial RDF store.

This research envisages that new requirements will create a strong foundation for the extension of the current tool in a fully integrated ontology driven decision support tool.

8.4. Knowledge Transfer

The technology transfer criteria established in chapter 5 and specified in the research design stated that the outcomes should:

- *Be published as an open source project* – the code is open source and will be fully published after the thesis publication;

- *Emphasize new ODKD functionalities* – the new functionalities are the basis for new projects, and of a new tool design described later in this section;

- *Be an ontology engineering platform for data mining research and projects* – the previous and next section, and several research publications, demonstrate its use in different research projects;

- *Leverage specific ontology engineering research* – several parts of the framework have been further developed as ontology engineering topics such as the inclusion of rule based engines to evaluate knowledge acquisition and integration of statistical algorithms with rule based systems;

- *Be able to be published in modules in order to be applied to different domains* – the next section describes a new design for ontology driven Intelligence systems.

8.4.1. Ontology Driven Intelligence Systems

Although the technology and knowledge acquired in this research have been actively used in research projects, technology transfer to industry was also established as one of the desired outcomes of this work. It seems fitting to close this thesis with a description of an ongoing project that is intended to deliver such an outcome. This section therefore briefly describes a general architecture which has been recently designed based on the ODKD framework. The design is under current development for an Intelligence knowledge base system (The system and industry will be omitted in order to respect the confidential nature of the project).

Risk and Intelligence Framework

This framework has been conceived to enable the creation of an ontology driven knowledge discovery system that will integrate different sources of data in a service oriented architecture to identify risk, monitor fraudulent activities and acquire intelligence in order to maintain compliance with the legal requirements of this industry.

The framework integrates several technologies and vendors (such as Oracle and Microsoft) and specialized identity resolution systems in an architecture able to collect, analyse and publish intelligence based on data and specialised knowledge from users.

Facilities to manage access control and provenance of change – crucial given the domain at hand - are founded on an ontology prototype based on the evolving ontology meta-knowledge model. This meta-knowledge will also be migrated to a RDF representation and extended with specific industry standards.

Solution Architecture

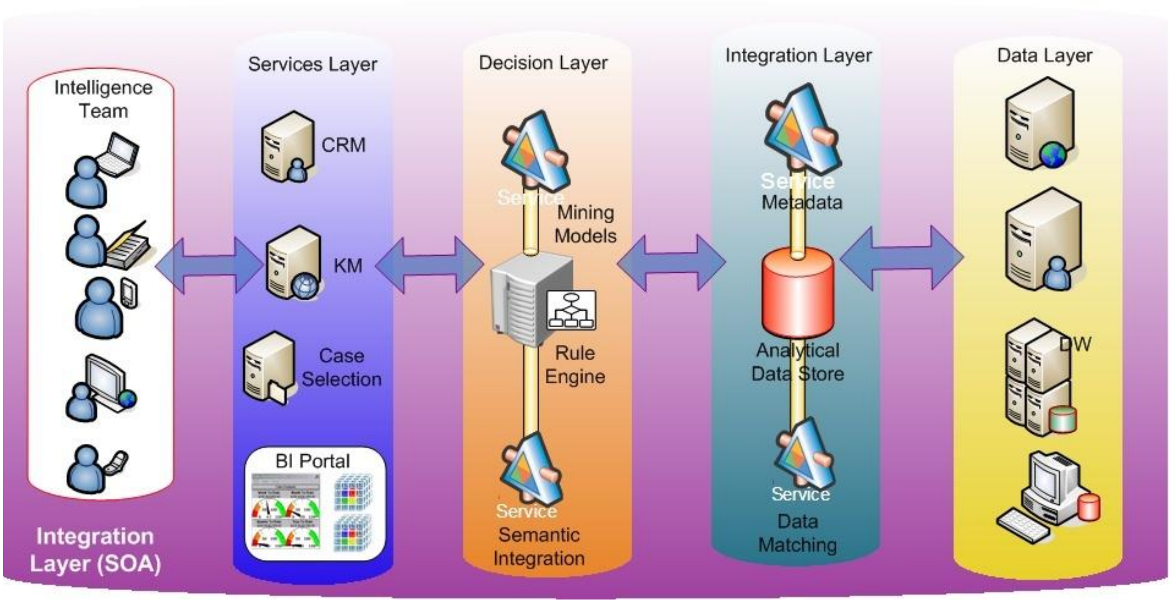


Figure 8-1– risk and intelligence solution architecture.

The solution (as shown in Figure 8-1) integrates different sources of information such as customer data acquired from call centres, internal data warehouses, web and internal systems. This data is exposed to the integration layer as services and annotated in the meta-data server based on the evolving ontology meta-knowledge model. Individual information based on the EO-Identification_Scheme (section 3.4.1) is used by a data matching system which scores the identity of individuals considering different identification attributes.

The analytical data store, based on an ontology, is then exposed to the decision layer through an API for further data analysis and intelligence analysis. A semantic relational query mechanism enables the user to query and select data from the ontology.

Rules are built to monitor the Intelligence knowledge base and alert when any suspicious activity is detected. Mining models are built to classify individuals and predict risks. The rules are integrated with the inference engine for processing, monitoring and model updating.

The system interfaces with the production systems to alert and expose controlled information to specific stakeholders/systems and the Intelligence community.

It should be clearly evident that this design is based on the ontology driven knowledge discovery process described in this thesis. It has undergone prototyping for further requirement gathering. The backend is Oracle 10g Spatial with both database and RDF store. The prototype front-end is Protégé frames. Protégé's Oracle plug-in has been tested to interface and build the RDF knowledge base.

The design has been approved and is under the evaluation of an IT strategic architecture special group which is assessing the impact and integration of the solution with the current IT infrastructure.

This research has therefore contributed not only to the development of the emerging field of research integrating ontologies and knowledge discovery processes and tools, but has also transferred technology to industry by providing a foundation for the development of a new semantic driven solution. Future work is expected to further enrich both the ODKD Framework and its application in a growing range of domains.

References

ACM. (1998). *ACM Computing Classification System (1998).* Retrieved 01/07/2006, from http://www.informatik.uni-stuttgart.de/zd/buecherei/ifibib_hilfe_cr.html

Adams, L., & Courtney, J. (2004). Achieving relevance in IS research via the DAGS framework. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 8*, Hawaii, IEEE.

Alani, H. (2006). *TGVizTab*. Retrieved 01/07/06, from http://users.ecs.soton.ac.uk/ha/TGVizTabTGVizTab.htm

Ashburner, M., & Ball, C. A. (2000). Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1), 25-29.

Aspirez, J., Gomez-Perez, A., Lozano, A., & Pinto, S. (1998). *(onto)2agent: An ontology-based www broker to select ontologies*. Paper presented at the Workshop on Applications of Ontologies and Problem-Solving Methods, Brighton, England.

Avery, J., & Yearwood, J. (2004). *Supporting Evolving Ontologies by Capturing the Semantics of Change*. Paper presented at The 10th Australian World Wide Web Conference. Retrieved 01/07/2006, from http://ausweb.scu.edu.au/aw04/papers/refereed/avery/

Bairoch, A., Boeckmann, B., Ferro, S., & Gasteiger, E. (2004). Swiss-Prot: Juggling between evolution and stability . *Brief. Bioinform*, 5, 39-55.

Barnes, J. (2002). Conceptual biology: a semantic issue and more. *Nature*, 417(6889), 587-588.

Benuskova, L., Jain, V., Wysoski, S. G., & Kasabov, N. (2006). Computational Neurogenetic Modelling: A pathway to new discoveries in Genetic Neuroscience. *Intl. Journal of Neural Systems*, 16(3), 215-226.

Benuskova, L., & Kasabov, N. (2007). *Computational Neurogenetic Modeling* (Vol. XII). New York: Springer Verlag.

Biowisdom. (2003). *DiscoveryInsight*. Retrieved 01/07/2006, from http://www.biowisdom.com/

Blagosklonny, M., & Pardee, A. (2002). Conceptual biology: unearthing the gems. *Nature*, 416(6879), 373.

Bodenreider, O. (2001). Medical Ontology Research [Electronic version]: *National Center for Biomedical Communications,* Retrieved 01/07/2006, from http://lhncbc.nlm.nih.gov/lhc/servlet/Turbine/template/research,langproc,MedicalOntology.vm.

Bodenreider, O., Mitchell, J., & McCray, A. (2002). *Evaluation of the UMLS as a terminology and knowledge resource for biomedical informatics*. Paper presented *at the ACL'2002 Workshop Natural Language Processing in the Biomedical Domain*, Philadelphia, USA.

Bodenreider, O., Mitchell, J., & McCray, A. (2003). *Biomedical ontologies*. Paper presented at the Pacific Symposium on Biocomputing 2003: World Scientific, Hawaii, USA.

Brachman, R., & Anand, T. (1996). The Process of Knowledge Discovery in Databases: A Human-Centered Approach. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy (Eds.), *Advances In Knowledge Discovery And Data Mining* (pp. 37-57): AAAI Press/The MIT Press.

Bray, D. (2001). Reasoning for results. *Nature*, 412(6850), 863.

Brisson, L., & Collard, M. (2006). *An Ontology Driven Data Mining Process*. Retrieved 26/10/2006, from www.i3s.unice.fr/~mcollard/KEOPS.pdf

Buckminster, R. (2006). *Buckminster Fuller Institute*. Retrieved 26/07/2006, from http://www.bfi.org/designsc.htm

Burgun, A., Botti, G., Fieschi, M., & Le Beux, P. (1999). *Sharing knowledge in medicine: semantic and ontologic facets of medical concepts*. Paper presented at the IEEE SMC '99 - International Conference on Systems, Man, and Cybernetics, Tokyo, Japan.

Carroll, J., Bizer, C., Hayes, P., & Stickler, P. (2005). *Named graphs, provenance and trust*. Paper presented at the 14th international conference on World Wide Web, Chiba, Japan.

Catton, C., & Shotton, D. (2004). The use of Named Graphs to enable ontology evolution. [Electronic Version]. *W3C Workshop on the Semantic Web for Life Sciences*. Retrieved 01/07/2005 from http://www.bioimage.org/pub/paradigm.htm.

Cespivova H., J., R., V., S., M., K., & M, T. (2004). *Roles of Medical Ontology in Association Mining CRISP-DM Cycle*. Paper presented at the ECML/PKDD04 Workshop on Knowledge Discovery and Ontologies, Pisa, Italy.

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies, and why do we need them? *Intelligent Systems, IEEE* 14(1), 20-26.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., et al. (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. Retrieved 01/07/2007, from http://www.crisp-dm.org/Process/index.htm

Cheah, Y.-N., & Abidi, S. S. R. (2001). *Augmenting knowledge-based medical systems with tacit healthcare expertise: towards an intelligent tacit knowledge acquisition info-structure*. Paper presented at the 14th IEEE Symposium on Computer-Based Medical Systems (CBMS 2001), Maryland, USA.

Chen, S., Alahakoon, D., & Indrawan, M. (2005). *Background knowledge driven ontology discovery*. Paper presented at the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, 2005. EEE '05, Hong Kong, China.

D'Hollosy, W., De Vries Robbe, P. F., Mars, N. J. I., Witjes, W. P. J., Debruyne, F. M. J., & Wijkstra, H. (1996). *Representation of domain knowledge needed to define relevant study variables for clinical trials in urology*. Paper presented at the Bridging Disciplines for Biomedicine. 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Amsterdam, Netherlands.

DCMI. (2007). *The Dublin Core Metadata Initiative*. Retrieved 01/08/2007, from http://dublincore.org/

Denny, M. (2004). *Ontology Tools Survey, Revisited.* [Electronic Version]. XML review, Editing Ontologies. Retrieved 01/07/2007 from http://www.xml.com/pub/a/2004/07/14/onto.html.

Devlin, K. (2001). *Infosense: Turning Information into Knowledge*. New York: W. H. Freeman.

Domingos, P. (1999). The Role of Occam's Razor in Knowledge Discovery. *Data Mining and Knowledge Discovery*, 3, 409-425.

Duane, D. (2006). *Design, Implementation and Testing of A Common Data Model Supporting Autonomous Vehicle Compatibility And Interoperability*. PhD Thesis. Retrieved 01/07/06, from http://www.stormingmedia.us/62/6207/A620754.html

Euzenat, J., Le Bach, T., Barrasa, J., Bouquet, P., de Bo, J., Dieng, R., et al. (2004). *D2.2.3: State of the Art on Ontology Alignment*. Retrieved 01/07/2005, from www.starlab.vub.ac.be/research/projects/knowledgeweb/kweb-223.pdf

Falconer, S. M., Noy, N., & Storey, M.-A. (2006). *Towards understanding the needs of cognitive support for ontology mapping*. Paper presented at the Ontology Matching Workshop. Athens, Georgia.

Fayyad, U. (1996). Data mining and knowledge discovery: making sense out of data. *IEEE Intelligent Systems and Their Applications*, 11(5), 20-25.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.). (1996). *Advances in Knowledge Discovery and Data Mining*. Menlo Park: AAAI Press/The MIT Press.

Fayyad, U., Piatetsky-Shapiro, G., & Uthurusamy, R. (2003). Summary from the KDD-03 panel: data mining: the next 10 years. *SIGKDD Explor. Newsl.*, 5(2), 191-196.

Fayyad, U., & Uthurusamy, R. (2002). Evolving data into mining solutions for insights. *Commun. ACM*, 45(8), 28-31.

Fensel, D., Horrocks, I., Van Harmelen, F., Decker, S., Erdmann, M., & Klein, M. (2000). *OIL in a nutshell*. Paper presented at the 12th International Conference on Knowledge Engineering and Knowledge Management, Juan-les-Pins, France.

Fernandez, M., Gomez-Perez, A., & Juristo, N. (1997). *METHONTOLOGY: From Ontological Art Towards Ontological Engineering*. Paper presented at the AAAI-97 Spring Symposium on Ontological Engineering, Stanford University, Palo Alto, USA.

Flouris, G., & Plexousakis, D. (2005). *Handling Ontology Change: Survey and Proposal for a Future Research Direction. (No. TR-362)*. Crete: Institute of Computer Science, FO.R.T.H. Retrieved 01/07/2007, from http://www.ics.forth.gr/isl/publications/
paperlink

Foster, I. (2006). 2020 computing: A two-way street to science's future. [Electronic Version]. *Nature*, 440, 419-440 from http://www.nature.com/nature/focus/futurecomputing/index.html.

Futschik, M., Reeve, A. & Kasabov, N. (2003). Evolving connectionist systems for knowledge discovery from gene expression data of cancer tissue. *Artif. Intell. Med.*, vol. 28, pp. 165-189.

Gandon, F. (2006). Ontology Engineering: a Survey and a Return on Experience. (No. RR-4396): INRIA [Electronic version].from http://hal.inria.fr/docs/00/07/21/92/PDF/RR-4396.pdf

Ganter, B., Wille, R., & Stumme, G. (2005). *Formal Concept Analysis*. New York: Springer.

Garcia, A., Ferraz, I., & Pinto, F. (2006). *The Role of Domain Ontology in Text Mining Applications: The ADDMiner Project*. Paper presented at the Sixth IEEE International Conference on Data Mining, Las Vegas, USA.

Gärdenfors, P. (1990). The dynamics of belief systems: Foundations vs. coherence theories. *Revue International de Philosopie* 44, 24--46.

Giarratano, J., & Riley, G. (2004). *Expert Systems: Principles and Programming* (III ed.): Boston: PWS-Kent Publishing Co.

Glass, A., & Karopka, T. (2002). Genomic Data Explosion - The Challenge for Bioinformatics. In P. Perner (Ed.), *Advances in Data Mining - Applications in E-commerce, Medicine, and Knowledge Management* (pp. 80-98). Berlin: Springer.

GO. (2006). *The Gene Ontology Consortium*. Retrieved 01/07/2006, from http://www.geneontology.org.

Gómez-Perez, A. (1999). Ontological engineering: A state of the art. *Expert Update*, 2(3), 33-43.

Gómez-Pérez, A. (2002). *Deliverable 1.3: A survey on ontology tools*. Retrieved 01/02/2003, from http://ontoweb.aifb.uni-karlsruhe.de/

Gómez-Pérez, A. (2003). *Survey of ontology learning methods and techniques*. Retrieved 01/07/2003, from http://ontoweb.aifb.uni-karlsruhe.de/

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). *Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Berlin: Springer-Verlag.

Gottgtroy, M. P. B., & Gottgtroy, P. (2001). *Significant Conceptual Modeling - A health care enterprise case study*. Paper presented at the Database and Expert Systems Applications, Munich, Germany.

Gottgtroy, P. (2003). *Ontology builder for biomedical informatics*. Paper presented at the Neurocomputing and Evolving Intelligence 2003, Auckland, New Zealand.

Gottgtroy, P., Kasabov, N., & MacDonell, S. (2003a). *Building Evolving Ontology Maps for Data Mining and Knowledge Discovery in Biomedical Informatics*. Paper presented at the Brazilian Symposium of Mathematical and Computational Biology, Rio de Janeiro, Brazil.

Gottgtroy, P., Kasabov, N., & MacDonell, S. (2003b). *An ontology engineering approach for Knowledge Discovery from data in evolving domains*. Paper presented at the Data Mining IV, Rio de Janeiro, Brazil.

Gottgtroy, P., Kasabov, N., & MacDonell, S. (2006). Evolving Ontologies for Intelligent Decision Support. In E. Sanchez (Ed.), *Fuzzy Logic and the Semantic Web* (pp. 415-441): Elsevier.

Gottgtroy, P. C. M. (2000). *A Conceptual Model Proposal for Dynamic Systems Requirements Elicitation*. Federal University of Rio Grande do Norte, Natal, Brazil.

Gregory, P.-S. (2000). Knowledge discovery in databases: 10 years after. S*IGKDD Explor. Newsl.*, 1(2), 59-61.

Gruber, T. R. (1993a). *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. Paper presented at the International Workshop on Formal Ontology, Padova, Italy.

Gruber, T. R. (1993b). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2), 199-220.

Gruninger, M., & Fox, M. S. (1995). *Methodology for the design and evaluation of ontologies*. Paper presented at the International Joint Conference on Artificial Intelligence (IJCAI-95). Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada.

Guarino, N. (1998). *Formal Ontology and Information Systems*. Paper presented at the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy.

Guarino, N., & Welty, C. (2000). *Towards a Methodology for Ontology Based Model Engineering*. Paper presented at the ECOOP-2000 Workshop on Model Engineering, Sophia Antipolis, France.

Haase, P., & Stojanovic, L. (2005). *Consistent evolution of OWL ontologies*. Paper presented at the Second European Semantic Web Conference (ESWC '05), Heraklion, Crete.

Hahn, W., & Weinberg, R. A. (2002). A subway map of cancer pathways. *Nature Review* Retrieved 01/07/2006, from http://www.nature.com/nrc/ poster/subpathways/index.html

Hartmanis, J. (1994). Turing Award lecture on computational complexity and the nature of computer science. *Commun. ACM*, 37(10), 37-43.

Hartmanis, J. (1995). On computational complexity and the nature of computer science. *ACM Comput. Surv.*, 27(1), 7-16.

Hayes, P. (2004). *RDF Semantics.* [Electronic Version]. Retrieved 01/07/2005 from http://www.w3.org/TR/rdf-mt/.

Hayes, P., & Menzel, C. (2004). *Simple Common Logic* [Electronic Version]. Retrieved 01/07/2005 from http://www.w3.org/2004/12/rules-ws/paper/103/.

Heflin, J., Hendler, J., & Luke, S. (1999). *Coping with Changing Ontologies in a Distributed Environment*. Paper presented at the AAAI workshop on ontology management, Orlando, Florida, USA.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems. *Research MIS Quarterly* 28(1), 75-105.

Honavar, V., Andorf, C., Caragea, D., Silvescu, A., Reinoso-Castillo, J., & Dobbs, D. (2001). *Ontology-Driven Information Extraction and Knowledge Acquisition from Heterogeneous, Distributed, Autonomous Biological Data Sources*. Paper presented at the International Joint Conference on Artificial Intelligence - IJCAI-01. Workshop on Knowledge Discovery from Heterogeneous, Distributed, Dynamic, Autonomous Data and Knowledge Sources, Seattle, Washington, USA.

Ives, B., Hamilton, S., & Davis, G. (1980). A framework for research in computer-based management information systems. *Management Science*, 26(9), 910-934.

Jinze, L., Wei, W., & Jiong, Y. (2004). *A framework for ontology-driven subspace clustering*. Paper presented at the Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle, WA, USA.

Joline, M., & Joey, F. G. (1995). Exploring the software engineering component in MIS research. *Commun. ACM*, 38(7), 80-91.

Kang, D.-K., Silvescu, A., Zhang, J., & Honavar, V. (2004). Generation of Attribute Value Taxonomies from Data and Their Use in Data-Driven Construction of Accurate and Compact Naive Bayes Classifiers. *Knowledge Discovery and Ontologies (KDO-2004)* Retrieved 26/07/2006, from http://olp.dfki.de/pkdd04/cfp.htm

Kasabov, N., & Benuskova, L. (2004a). Computational neurogenetics. *Journal of Computational and Theoretical Nanoscience,* 1(1), 47-61.

Kasabov, N., & Benuskova, L. (2004b). Neuro-, genetic-, and neurogenetic information processing. In M. Rieth & W.Schommers (Eds.), *Handbook of Computational and Theoretical Nanotechnology*. Los Angeles: American Scientific Publishers.

Kasabov, N., Benuskova, L., & Wysoski, S. G. (2004). *Computational neurogenetic modelling: gene networks within neural networks*. Paper presented at the International Joint Conference on Neural Networks, Budapest, Hungary.

Kasabov, N., Jain, V., Gottgtroy, P., Benuskova, L., & Joseph, F. (2007). Brain-Gene Ontology: Integrating Bioinformatics and Neuroinformatics Data, Information and Knowledge to Enable Discoveries. In *Computational Intelligence in Bioinformatics*: Spring Verlag .

Kasabov, N., Vishal, J., Gottgtroy, P., Benuskova, L., & Joseph, F. (2007). Evolving Brain-Gene Ontology and Simulation System (BGOS): Towards Integrating Bioinformatics and Neuroinformatics Data, Information and Knowledge to Facilitate Discoveries. *Special Issue of Neural Networks*.

Kasabov, N. (2009). Integrative connectionist learning systems inspired by nature: current models, future trends and challenges. *Natural Computing: an international journal* 8, 2, 199-218.

Kauppinen, T., & Hyvönen, E. (2006). *Bridging the Semantic Gap between Ontology Versions* [Electronic Version]. Retrieved 01/07/2006 from http://www.seco.tkk.fi/events/2004/2004-09-02-web-intelligence/papers/changebridges-wis04.pdf.

KDO-2004. (2004). *Knowledge Discovery and Ontologies (KDO-2004).* Retrieved 01/07/2006, from http://olp.dfki.de/pkdd04/cfp.htm

KEDRI. (2006a). *Knowledge Engineering and Discovery Research Institute*. Retrieved 01/07/2006, from http://www.aut.ac.nz/research/research_institutes/kedri/

KEDRI. (2006b). *NeuCom*. Retrieved 01/07/2006, from www.theneucom.com

Klein, M., & Noy, F. (2003). *A Component-Based Framework for Ontology Evolution*. Paper presented at the IJCAI-03 Workshop on Ontologies and Distributed Systems, Acapulco, Mexico.

Knublauch, H. (2003). An AI tool for the real world [Electronic Version]. *JavaWorld.com*. Retrieved 01/07/2007 from http://www.javaworld.com/javaworld/jw-06-2003/jw-0620-protege.html.

Köhler, J., & Schulze-Kremer, S. (2002). The Semantic Metadatabase (SEMEDA): Ontology based integration of federated molecular biological data sources. In *Silico Biology*, 2(0021).

Kuhn, T. S. (1996). *The Structure of Scientific Revolutions.* (3rd edition ed.). Chicago, Illinois: University of Chicago Press.

Lambrix, P., Habbouche, M., & Perez, M. (2003). Evaluation of ontology development tools for bioinformatics. *Bioinformatics*, 19 (12), 1564-1571.

Legrand, S., & J., P. (2004). A Hybrid Approach to Word Sense Disambiguation: Neural Clustering with Class Labeling. *Knowledge Discovery and Ontologies (KDO-2004)* Retrieved 26/07/2006, from http://olp.dfki.de/pkdd04/cfp.htm

Lytras, M. (2004). Tom Gruber in AIS SIGSEMIS Bulletin!!! [Electronic Version]. *Bulletin of AIS Special Interest Group on Semantic Web and Information Systems,* 1, 4-11. Retrieved 01/08/2007 from http://lsdis.cs.uga.edu/library/download/LSDIS-Lab-Report-sigsemis_bulletin_1-3_2004.pdf.

Mannila, H. (1996). *Data mining: machine learning, statistics, and databases*. Paper presented at the Eight International Conference on Scientific and Statistical Database Management, Stockholm, Sweden.

Marchiori, M. (2002). *The platform for privacy preferences*. [Electronic Version] from http://www.w3.org/TR/P3P/.

Marcus, G. (2004). *The Birth of the Mind: How a Tiny Number of Genes Creates the Complexity of the Human Mind*. New York: Basic Books.

MathWorks. (2006). *Matlab*. Retrieved 01/07/06, from http://www.mathworks.com/

McGarry, K., & Wermter, S. (2007). *Integration of Hybrid Bio-Ontologies using Bayesian Networks for Knowledge Discovery*. Paper presented at the Workshop on Neuro-Symbolic Learning and Reasoning, International Joint Conference on Artificial Intelligence (IJCAI-07), Hydrabad, India.

Mihael, A. (2002). Report on the SIGKDD-2002 panel the perfect data mining tool: interactive or automated? *SIGKDD Explor. Newsl.*, 4(2), 110-111.

Minsky, M. (1974). *A Framework for Representing Knowledge*. [Electronic Version]. Retrieved 01/07/2005 from http://web.media.mit.edu/~minsky/papers/Frames/frames.html.

Mizoguchi, R., & Ikeda, M. (2006). Towards Ontology Engineering (No. AI-TR-96-1) [Electronic Version] Retrieved 01/07/2007, from http://www.ei.sanken.osaka-u.ac.jp/pub/miz/miz-onteng.pdf

Nardi, D., & Brachman, R. J. (2002). An Introduction to Description Logics. In *The Description Logic Handbook*. (pp. 5-44): Cambridge University Press.

Natalya, F. N., & Mark, A. M. (2003). The PROMPT suite: interactive tools for ontology merging and mapping. *Int. J. Hum.-Comput. Stud.*, 59(6), 983-1024.

NCBI. (2006). *PubMed.* Retrieved 01/07/2006, from http://www.ncbi.nlm.nih.gov/sites/entrez?db=pubmed

NCI. (2003). *Unified Medical Language System*. Retrieved 01/07/2003, from www.nlm.nih.gov/research/umls/.

NCBI (2005).The Nervous System. In: *Genes and Disease. National Centre for Biotechnology Information (NCBI)*, from http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=gnd.chapter.75

Newell, A., & Simon, H. (1976). Computer science as empirical inquiry: symbols and search. *Commun. ACM*, 19(3), 113-126.

Noe, K. (2002). The Structure of Scientific Discovery: From a Philosophical Point of View. In S. Arikawa & A. Shinohara (Eds.), *Progress in Discovery Science* (pp. 31-39). Berlin Heidelberg: Springer-Verlag.

8.5. Noy, N., Fergerson, R., & Musen, M. (2000). The Knowledge Model of Protege-2000: Combining Interoperability and Flexibility. Paper presented at the 12th European Workshop on Knowledge Acquisition, Modeling and Management, Juan-les-Pins, France.

Noy, N., & Klein, M. (2004). Ontology Evolution: Not the Same as Schema Evolution. *Knowledge and Information Systems*, 6(4), 428-440.

Nunamaker, J. F., Chen, M., & Titus, P. (1990). *Systems development in information systems research*. Paper presented at the Twenty-Third Annual Hawaii International Conference on System Sciences, Hawaii, USA.

Nunamaker, J. F., Chen, M., & Titus, P. (1991). Systems development in information systems research. *J. Management Information Systems*, 7(3), 89-106.

Ontospace. (2006). *OntoBase plug-in for Protégé*. Retrieved 01/07/06, from http://www.ontospace.net/

Ontoviz. (2006). *Ontoviz*. Retrieved 01/07/2006, from http://protege.cim3.net/cgi-bin/wiki.pl?OntoViz

Parikshit, B., Pears, R., & Gottgtroy, P. (2006). *Use of Association Rule Mining to form a Digital Library*. Paper presented at the New Zealand Ontology Workshop, Auckland, NZ.

Pat, L., & Herbert, A. (1995). Applications of machine learning and rule induction. *Commun. ACM*, 38(11), 54-64.

Pechenizkiy, M., Puuronen, S., & Tsymbal, A. (2006). On the Use of Information Systems Research Methods in Data mining. In O. Vasilecas, W. Wojtkowski & J. Zupančič (Eds.), *Information Systems Development Advances in Theory*, Practice, and Education (pp. 487-499): Springer.

Peter, M., & Hans, A. (2004). Towards a new synthesis of ontology technology and knowledge management. *Knowl. Eng. Rev.*, 19(4), 317-345.

Peter, P., Olga, T., & Sven, C. (2007). Understanding ontology evolution: A change detection approach. *Web Semant*., 5(1), 39-49.

Phillips, J., & Buchanan, B. G. (2001). *Ontology-guided knowledge discovery in databases*. Paper presented at the Proceedings of the International Conference on Knowledge Capture, Victoria, British Columbia, Canada.

Protege. (2006a). *Protege 2000*. Retrieved 01/07/2006, from http://www.protege. stanford.edu/.

Protege. (2006b). *Protege Plugins Library*. Retrieved 01/07/06, from http://protege.cim3.net/
cgi-bin/wiki.pl?ProtegePluginsLibraryByTopic

Quan, T. T., Hui, S. C., & Cao, T. H. (2004). *FOGA: A Fuzzy Ontology Generation Framework for Scholarly Semantic Web*. Paper presented at the Knowledge Discovery and Ontologies (KDO-2004), Pisa, Italy.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann Publishers Inc.

Rahm, E., & DO, H. (2000). Data Cleaning: Problems and Current Approaches [Electronic Version]. *Bulletin of the Technical Committee on Data Engineering*, 23. Retrieved 01/07/2007 from http://wwwiti.cs.uni-magdeburg.de/iti_db/lehre/dw/paper/
data_cleaning.pdf.

Ramoni, M., Stefanelli, M., Magnani, L., & Barosi, G. (1992). An epistemological framework for medical knowledge-based systems. *IEEE Transactions on Systems, Man and Cybernetics*, 22(6), 1361-1375.

Rebhan, M., Chalifa-Caspi, V., Prilusky, J., & Lancet, D. (1997). *GeneCards: encyclopedia for genes, proteins and diseases*: Weizmann Institute of Science, Bioinformatics Unit and Genome Center, http://www.genecards.org/.

Reinberger, M.-L., Spyns, P., Pretorius, J., & Daelemans, W. (2004). *Automatic Initiation of an Ontology*. Paper presented at the ODBase'04, Ayia Napa, Cyprus.

Rosse, C., & Mejino, J. L. V. (2003). A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *J Biomed Inform.* , 36, 478-500.

Rüping, S. (2007). Advancing Clinico-Genomic Clinical Trials on Cancer: Open Grid Services for improving Medical Knowledge Discovery. Report: Consolidated

Requirement Analysis for Data Mining, Analysis and the Visualization Environment. Retrieved 01/03/07, from http://www.eu-acgt.org/documents/public-deliverables.html

Sanchez, E. (2006). *Fuzzy Logic and the Semantic Web.* Amsterdan: Elsevier.

Sarabjot, S., David, A., & John, G. (1995). *The role of domain knowledge in data mining*. Paper presented at the Proceedings of the Fourth International Conference on Information and Knowledge Management, Baltimore, Maryland, USA.

Schreiber, G., Akkermans, H., Anjewierden, A., Hoog, R., Shadbolt, N., Van de Velde, W., et al. (1999). *Knowledge Engineering and Management: The CommonKADS Methodology*. Boston: The MIT Press.

Schulze-Kremer, S. (2002). Ontologies for molecular biology and bioinformatics. In *Silico Biology*, 2(0017).

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, 5(4), 13-22.

Sheth, A., & Ramakrishnan, C. (2000). Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis [Electronic Version]. *Bulletin of the Technical Committee on Data*

Engineering, 23. Retrieved 01/07/2007 from http://knoesis.wright.edu/library/download/SR03-BW.pdf.

Silberfein, A., & Gennari, J. (2006). *OBO Tab*. Retrieved 01/07/06, from http://faculty.
washington.edu/gennari/Protege-plugins/OBO-import/

Sintek, M. (2006). *XML Tab*. Retrieved 01/07/06, from http://protege.cim3.net/cgi-bin/wiki.pl?XMLTab

Sofia Pinto, H., Gómez-Perez, A., & Martins, J. P. (1999). *Some Issues on Ontology Integration*. Paper presented at the Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends, Stockholm, Sweden.

Sowa, J. F. (1984). *Information Processing in Mind and Machine*. Reading: Addison-Wesley.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove: Cole Publishing Co.

Sowa, J. F. (2002). *Ontology, Metadata, and Semiotics* [Electronic Version]. Retrieved 02/12/2006 from http://users.bestweb.net/~sowa/peirce/ ontometa.htm.

Sowa, J. F. (2005). *Building, Sharing, and Merging Ontologies* [Electronic Version]. Retrieved 02/12/2006 from http://www.jfsowa.com/ontology/ontoshar.htm.

Sprague, R. H., & Carlson, E. D. (1982). *Building Effective Decision Support Systems*. Englewood Cliffs, N.J: Prentice Hall College Div.

Spyns, P., Meersman, R., & Jarrar, M. (2002). Data modelling versus ontology engineering. *SIGMOD Record Special Issue on Semantic Web, Database Management and Information Systems*, 31(4).

Stojanovic, L., Maedche, A., Stojanovic, N., & Studer, R. (2003). *Ontology Evolution as Reconfiguration-Design Problem Solving*. Paper presented at the International Conference on Knowledge Capture (K-CAP-03), Florida, USA.

Straccia, U. (2006). *A Fuzzy Description Logic for the Semantic Web*. In E. Sanchez (Ed.), Fuzzy Logic and the Semantic Web (pp. 73-90): Elsevier.

Sure, Y. (2004). *Ontology Evolution*. Retrieved 01/07/2005, from www.sdkcluster.org/
presentations/2004-04-06%20SDK-SEKT-Evolution.ppt

Sure, Y., Staab, S., & Studer, R. (2002). Methodology for development and employment of ontology based knowledge management applications. *SIGMOD Rec.*, 31(4), 18-23.

Svatek, V., Rauch, J., & Flek, M. (2005). Ontology-Based Explanation of Discovered Associations in the Domain of Social Reality. *Knowledge Discovery and Ontologies (KDO-2005)* Retrieved 26/07/2006, from webhosting.vse.cz/svatek/KDO05/
paper10.pdf

Tadepalli, S., Sinha, A. K., & Ramakrishnan, N. (2004). *Ontology Driven Data Mining for Geosiences*. Paper presented at the Geographical Society of America - Geosience in a changing world, Denver, Colorado.

TouchGraph. (2006). *TouchGraph*. Retrieved 01/07/06, from http://www.touchgraph.com/
index.html

TWO, R. (2006). *GO-KDS*. Retrieved 01/07/2006, from www.go-kds.com/

UMLS. (2006). *UMLS Tab*. Retrieved 01/07/06, from http://protege.cim3.net/cgi-bin/wiki.pl?ProtegePluginsLibraryByTopic#nid3Q1

Uschold, M., & Gruinger, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2), 93-155.

Vasant, D., & Roger, S. (1997). *Seven methods for transforming corporate data into business intelligence*. Englewood Cliffs, N.J., Prentice-Hall: Prentice-Hall.

Verma, A., Gottgtroy, P., Havukkala, I., & Kasabov, N. (2006). *Developing "Evolving Ontologies" for Nutritional Advice for Type-2 diabetes Patients*. . Paper presented at the Molecular Biology Meeting Queenstown, New Zealand.

Verma, A., Song, Q., & Kasabov, N. (2006). *Developing "Evolving Ontology" for Personalised Risk Evaluation for Type-2 Diabetes Patients*. Paper presented at the 6th International Conference on Hybrid Intelligent Systems (HIS'06), Auckland, New Zealand.

Walker, M. G. (1986). How Feasible is Automated Discovery. *IEEE Expert*, 2(1), 70-82.

Wang, L., & Gottgtroy, P. (2006). *Ontology Visualization: A Biomedical Informatics case study*. Paper presented at the New Zealand Ontology Workshop, Auckland, NZ.

Wang, Y. (2007). *Ontology Engineering: The Brain Gene Ontology Case Study*. Auckland University of Technology, Auckland.

Watts, M. J. 2009. A decade of Kasabov's evolving connectionist systems: a review. *Trans. Sys. Man Cyber Part C* 39, 3 (May. 2009), 253-269.

Witten, I., & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. (2nd ed.). San Francisco: Morgan Kaufmann.

Yen-Ting, K., Andrew, L., Liz, S., & Kathy, P. (2007). *Domain ontology driven data mining: a medical case study*. Paper presented at the Proceedings of the 2007 international workshop on Domain driven data mining, San Jose, California,USA.

Yoon., S.-C., Henschen, L. J., Park, E. K., & Makki, S. (1999). *Using Domain Knowledge in Knowledge Discovery*. Paper presented at the Conference on Information and Knowledge Management. Kansas City, MO, USA.

Zadeh, L. A. (2006). From Search Engines to Question Answering Systems - The Problems of World Knowledge, Relevance, Deduction and Precisiation. In E. Sanchez (Ed.), *Fuzzy Logic and the Semantic Web* (pp. 163-210): Elsevier.

Appendix A

1. Ontology On-line Analytical Pre-processing (OOLAP)

On-line analysis has been widely used in data warehousing in order to explore huge amounts of multidimensional data. The use of visualization techniques in conjunction with simple tabular data enable users, mainly decision makers, to work with business intelligence tools that are familiar to them in the form of spreadsheets and graphs. The success of these business intelligence tools is constrained, however, by the limited effectiveness of the data warehouse model, driven, as it is, by the static characteristics of its underlying data model. Any query, investigation or even insight desired that is out of the range of the multi-dimensional model is difficult and expensive to obtain. The ontology driven knowledge discovery approach avoids this restriction by adding a flexible semantic layer (evolving ontologies) to the data warehouse schema.

Ontology On-line Analytical Pre-processing (OOLAP) enables detailed, human-driven (ad-hoc) analysis of the domain. It converts any highly dimensional ontology space into a simple tabular representation that can be translated via semantic mapping into a multi-dimensional format to be used by current business intelligence tools as well as to be further explored by data mining workbenches. Additionally, it interacts with available Online Analytical Processing (OLAP) and visualization tools to facilitate ontology exploration and analysis in order to refine the ontology model and improve data and domain understanding. In particular, OOLAP encompasses the ontology visualization, population and instance selection activities of the ontology driven knowledge discovery process.

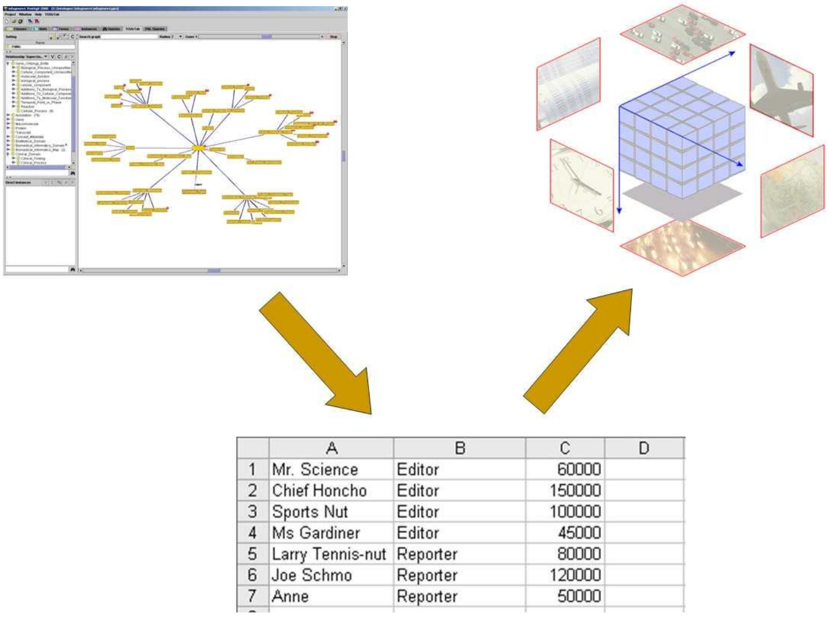
The OOLAP process is currently implemented using a variety of tools for ontology engineering, data warehousing, OLAP, and data mining, namely: Protégé, MS-Analysis Service, Excel spreadsheets. These tools were selected so as to avoid any extra learning requirement by both experts and users in the experiments executed in this research. However the principles behind of this alternative 'pipeline' are not specific to any platform or vendor and might be implemented using any online analytical tool or server.

1.1. OOLAP alternative pipeline

OOLAP encompasses three tasks of the ontology driven knowledge discovery process: Ontology Visualization, Ontology Population and Instance Selection. It can be considered an alternative process that is akin to the CRISP-DM process. OOLAP adds an ontology analysis capability as well as an ontology integration tool able to integrate data warehouse models.

As described previously in the Ontology Visualization task, this pipeline utilizes different visualization techniques developed in the Protégé environment, such as Jambalaya (Falconer, Noy, & Storey, 2006) and Ontoviz (Ontoviz, 2006), in order to gain an understanding of the domain and to select the best concepts for analysis. The different ontology visualizations support user investigation and enable exploration of the knowledge base from different perspectives. Furthermore, as described earlier, the environment incorporates a query language that can speed up any data search or concept understanding effort.

Additionally, through the instance selection tool, described in chapter 5, it explores Protégé's knowledge acquisition feature to enable the user to update the knowledge base 'on the fly' where necessary. Visualization can facilitate the identification of important missing concepts or relationships that should be further explored in the data understanding phase.



	A	B	C	D
1	Mr. Science	Editor	60000	
2	Chief Honcho	Editor	150000	
3	Sports Nut	Editor	100000	
4	Ms Gardiner	Editor	45000	
5	Larry Tennis-nut	Reporter	80000	
6	Joe Schmo	Reporter	120000	
7	Anne	Reporter	50000	

Figure A-1– A newspaper sample Ontology/cube mapping.

The next stage is concerned with the translation of the ontology format into a simple tabular format used in the OLAP cube and thus able to be processed by the OLAP tool (see Figure A-1). Once they have a level of domain understanding, the user can navigate the ontology and select the scope of his/her analysis. The selected data is then mapped and translated to a local cube format.

The current development translates the knowledge base into a text specific format compatible with Microsoft’s local cube specification. However the translation can be done for any cube format depending on the platform and OLAP tool(s) selected. One of the main characteristics of this local cube approach is its total independence of the possibly complex database environments, such as SQL Server, MYSQL or Oracle.

After exploring the multi-dimensional, application-independent and semantically powerful ontology environment and translating it to the cube format, the next step exploits all the advantages of well-developed OLAP techniques to analyse the target problem.

This step introduces human-driven analysis into the data preparation step of the CRISP-DM and ODKD processes. The data selected is made available to the decision maker/analyst through the local cube that enables him/her to use OLAP tools for his/her analysis. MS-Excel and MS- Data Analyser were chiefly used in the experiments.

1.2. OLAP biomedical informatics application

The OOLAP approach was used in the early development of a brain disease ontology, which represents knowledge of genes and proteins that are linked to specific brain-related disorders such as epilepsy and schizophrenia. The analysis was focused on the crucial neuronal parameters that are in some way thought to control the phenomenon of epileptic seizures and/or through their direct or indirect interactions with other genes/proteins, influencing the gene regulatory behaviour in brain diseases.

The OOLAP methodology helped to identify missing concepts, new relationships and select the best candidates for our modelling tasks. This OOLAP biomedical informatics case study was developed as part of a wider ontological research effort in neurogenetic modelling that integrates dynamic gene networks within neurons (Kasabov, Benuskova, & Wysoski, 2004) described in the biomedical application chapter (Chapter 6).

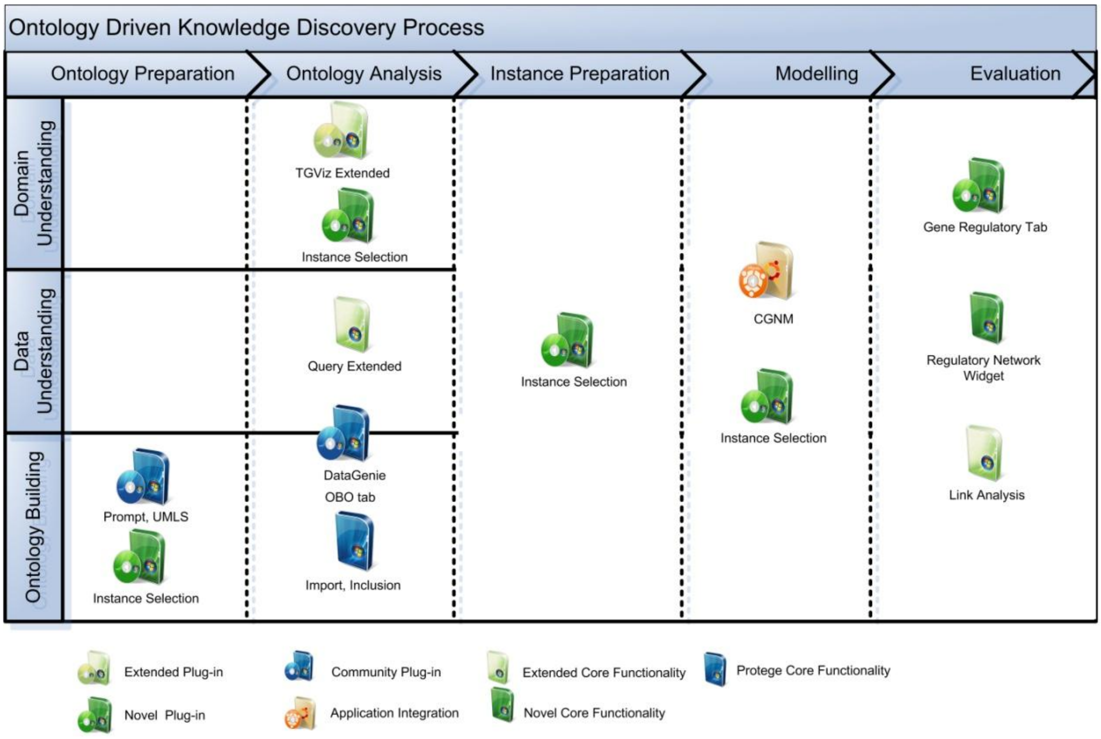
Appendix B

1. Introduction

This appendix shows a series of screenshots taken from the biomedical case study. This set doesn’t intend to cover all features and functionalities developed in this research but may help the reader to visualise the content of the application.

The Table B-1 - Ontology Driven Knowledge Discovery phases and tasks and tools.illustrated the relationship between the ODKD process and main plug-ins utilized in this biomedical case study. Some specific plug-ins were developed to support the case study and are described in the table below.

Table B-1 - Ontology Driven Knowledge Discovery phases and tasks and tools.



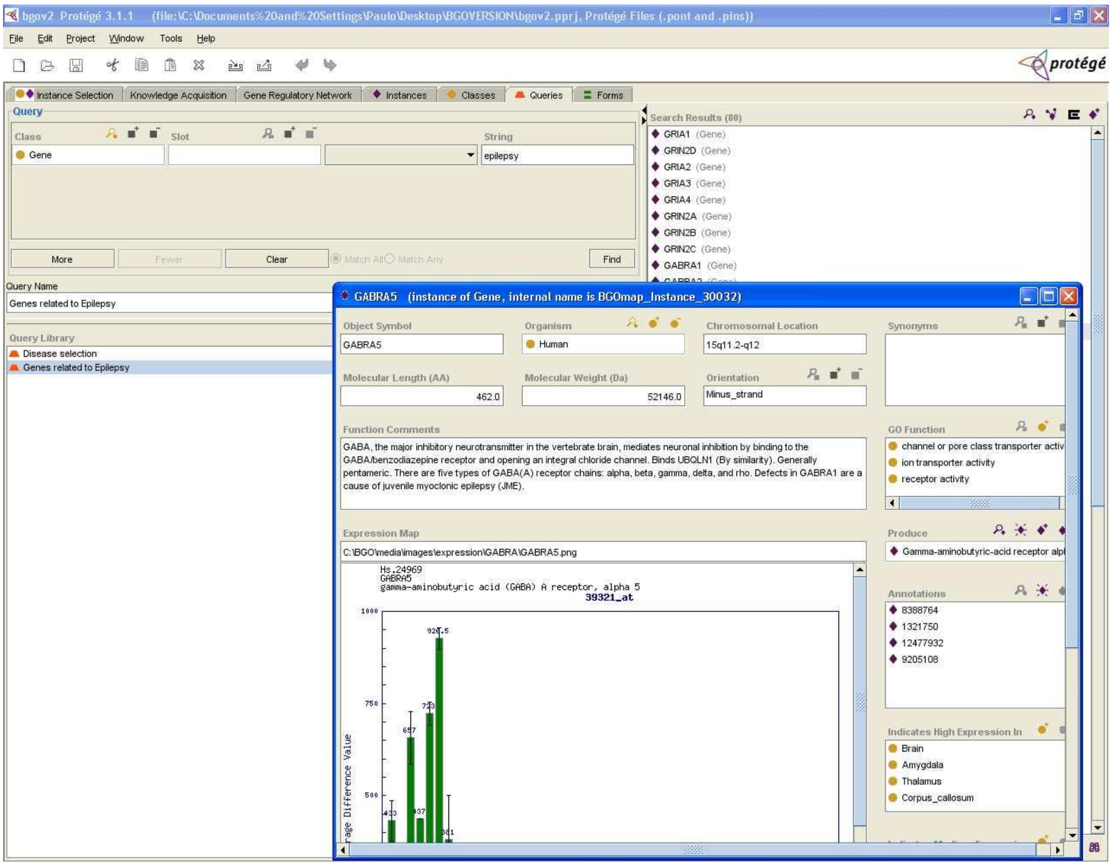
Ontology Driven Knowledge Discovery Process

	Ontology Preparation	Ontology Analysis	Instance Preparation	Modelling	Evaluation
Domain Understanding		TGViz Extended Instance Selection	Instance Selection	CGNM Instance Selection	Gene Regulatory Tab Regulatory Network Widget Link Analysis
Data Understanding		Query Extended			
Ontology Building	Prompt, UMLS Instance Selection	DataGenie OBO tab Import, Inclusion			

Extended Plug-in
Novel Plug-in
Community Plug-in
Application Integration
Extended Core Functionality
Novel Core Functionality
Protege Core Functionality

2. Exploring brain gene knowledge with the BGO system

All ontology instances and concepts are traceable through a query language plug-in that enables the searching for answers to questions such as; 'which genes are related to the occurrence of epilepsy?', by, for example, simply typing the key word epilepsy into the query window and selecting the class gene (see Figure B-1). As a result, the system returns a list of 80 genes potentially related to epilepsy. By selecting any of them we can obtain additional, detailed information about that particular gene, for example, its GO function, chromosomal location, synonyms, brain expression profile, and so on.


bgov2 Protégé 3.1.1 (file:\C:\Documents%20and%20Settings\Paulo\Desktop\BGOVERSION\bgov2.pprj, Protégé Files (.pont and .pins))
File
Edit
Project
Window
Tools
Help
protégé
Instance Selection
Knowledge Acquisition
Gene Regulatory Network
Instances
Classes
Queries
Forms
More
Fewer
Clear
○ Match All ○ Match Any
Find
Search Results (88)
GABRA5 (instance of Gene, internal name is BGOmap_Instance_30032)

Query

Class: Gene

Slot:

String: epilepsy

Query Name: Genes related to Epilepsy

Query Library
- Disease selection
- Genes related to Epilepsy

Search Results (88)
- GRIA1 (Gene)
- GRIN2D (Gene)
- GRIA2 (Gene)
- GRIA3 (Gene)
- GRIA4 (Gene)
- GRIN2A (Gene)
- GRIN2B (Gene)
- GRIN2C (Gene)
- GABRA1 (Gene)

Object Symbol: GABRA5

Organism: Human

Chromosomal Location: 15q11.2-q12

Synonyms:

Molecular Length (AA): 462.0

Molecular Weight (Da): 52146.0

Orientation: Minus_strand

Function Comments: GABA, the major inhibitory neurotransmitter in the vertebrate brain, mediates neuronal inhibition by binding to the GABA/benzodiazepine receptor and opening an integral chloride channel. Binds UBQLN1 (By similarity). Generally pentameric. There are five types of GABA(A) receptor chains: alpha, beta, gamma, delta, and rho. Defects in GABRA1 are a cause of juvenile myoclonic epilepsy (JME).

GO Function
- channel or pore class transporter activ
- ion transporter activity
- receptor activity

Expression Map: C:\BGO\media\images\expression\GABRA\GABRA5.png

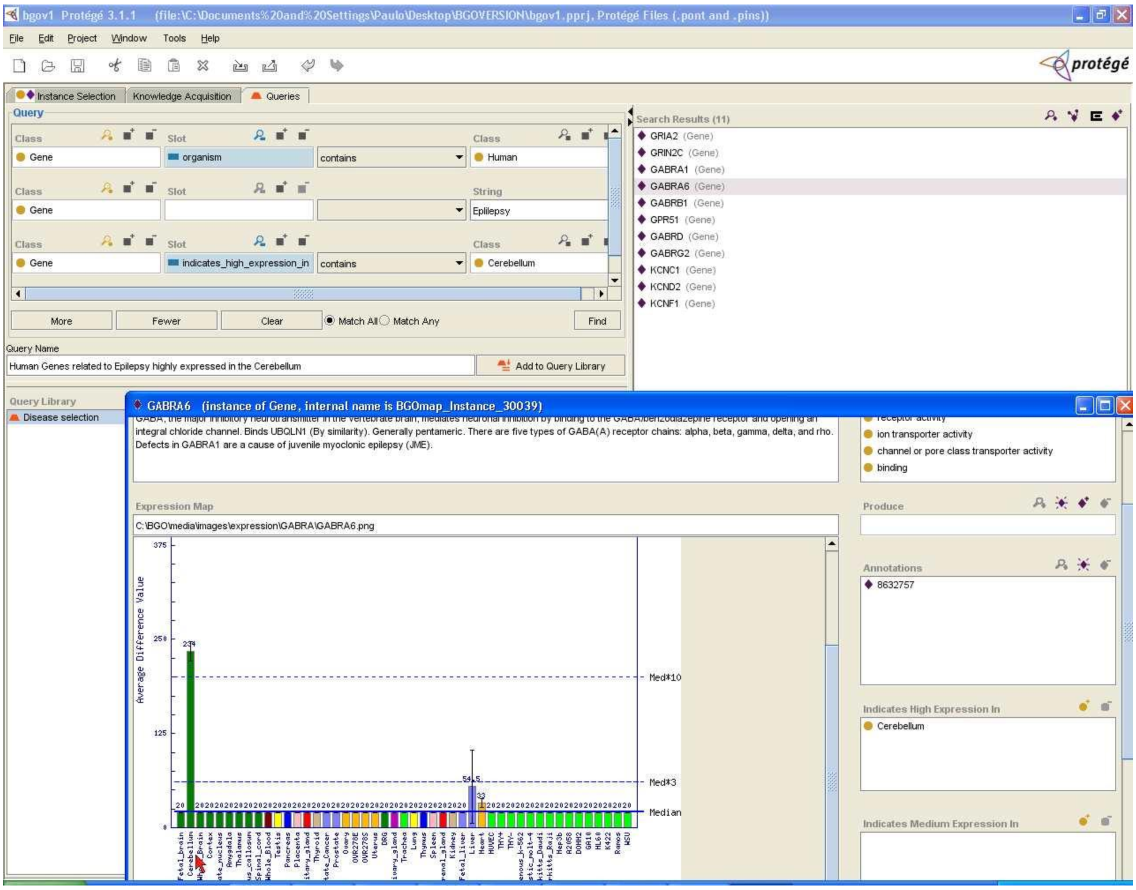
Produce
- Gamma-aminobutyric-acid receptor alpl

Annotations
- 8388764
- 1321750
- 12477932
- 9205108

Indicates High Expression In
- Brain
- Amygdala
- Thalamus
- Corpus_callosum

Figure B-1 – BGO System: Query search system looking for epilepsy and the GABRA5 gene as an example.

The query can be refined by filtering the search and/or creating different conditionals with relevant information as shown in Figure B-2. The query can also be stored in the system for future use.



bgov1 Protégé 3.1.1 (file:\C:\Documents%20and%20Settings\Paulo\Desktop\BGOVERSION\bgov1.pprj, Protégé Files (.pont and .pins))

Class: Gene | Slot: organism | contains | Class: Human

Class: Gene | Slot: | | String: Epilepsy

Class: Gene | Slot: indicates_high_expression_in | contains | Class: Cerebellum

● Match All ○ Match Any

Query Name: Human Genes related to Epilepsy highly expressed in the Cerebellum

Search Results (11)

- GRIA2 (Gene)
- GRIN2C (Gene)
- GABRA1 (Gene)
- GABRA6 (Gene)
- GABRB1 (Gene)
- GPR51 (Gene)
- GABRD (Gene)
- GABRG2 (Gene)
- KCNC1 (Gene)
- KCND2 (Gene)
- KCNF1 (Gene)

GABRA6 (instance of Gene, internal name is BGOmap_Instance_30039)

GABA, the major inhibitory neurotransmitter in the vertebrate brain, mediates neuronal inhibition by binding to the GABA/benzodiazepine receptor and opening an integral chloride channel. Binds UBQLN1 (By similarity). Generally pentameric. There are five types of GABA(A) receptor chains: alpha, beta, gamma, delta, and rho. Defects in GABRA1 are a cause of juvenile myoclonic epilepsy (JME).

receptor activity
ion transporter activity
channel or pore class transporter activity
binding

Expression Map: C:\BGO\media\images\expression\GABRA\GABRA6.png

Produce:

Annotations: 8632757

Indicates High Expression In: Cerebellum

Indicates Medium Expression In:

Figure B-2 - BGO System: Query result for human genes related to epilepsy highly expressed in the cerebellum with the GABRA6 gene as an example.

Additional to the query language, the biomedical tool has a plug-in which enables the selection of gene(s) of interest as well as a visualization of their relationships with other concepts/instances in the BGO. The visualisation plug-in extends the Protégé TGVizTab that, besides normal visualization and search capabilities, also provides an instance selection feature (shown in Figure B-3).

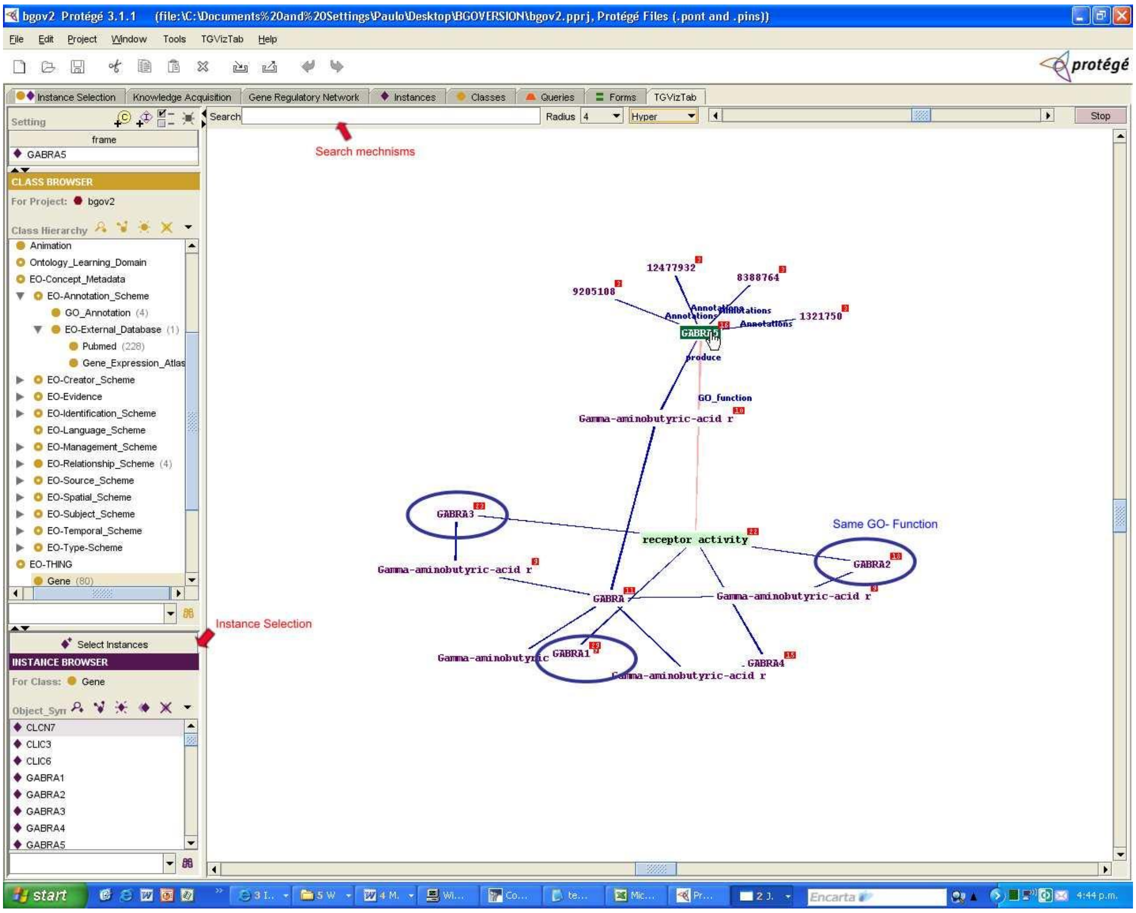
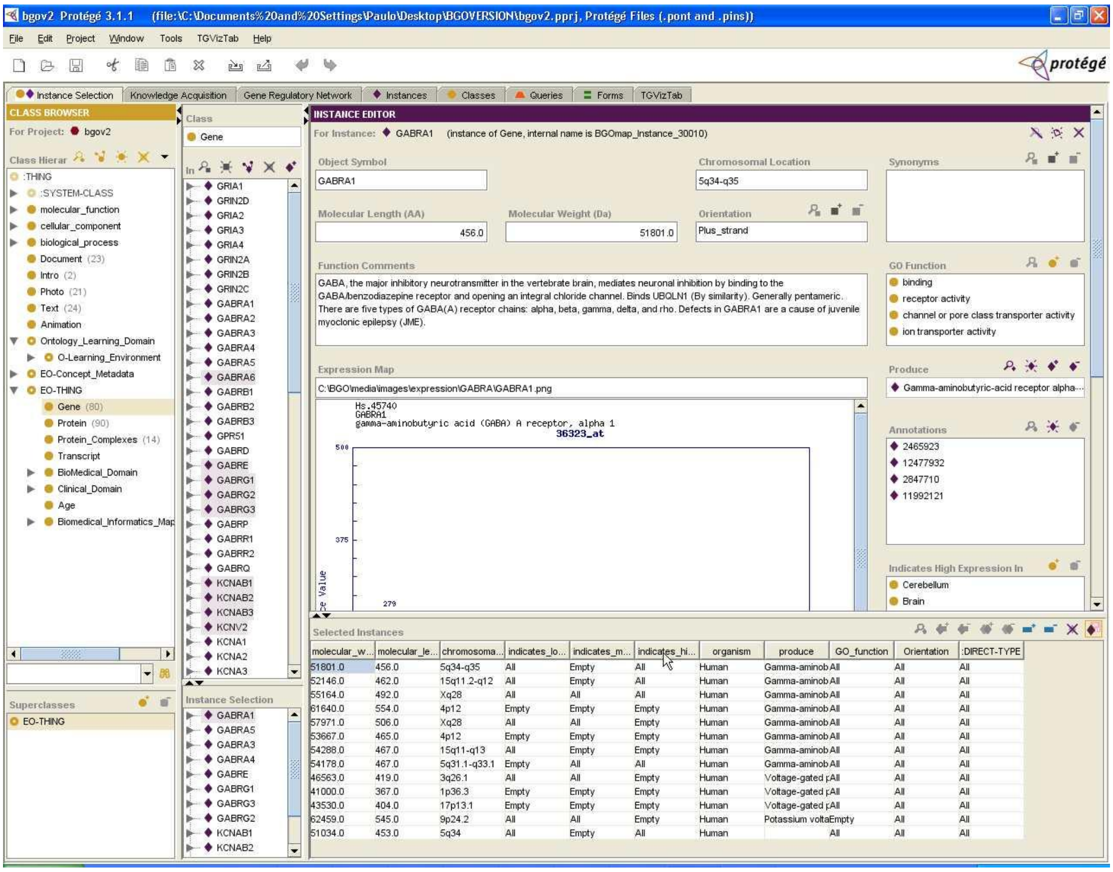


Figure B-3 - BGO System: Snapshot of BGO detail showing relationships between genes, proteins, molecular and neuronal functions for the GABRA5 gene. Each node can be expanded further so that one can identify relationships between molecular weight, chromosomal location, gene product, function in neurons, mutations and related diseases.



bgov2 Protégé 3.1.1 (file:\C:\Documents%20and%20Settings\Paulo\Desktop\BGOVERSION\bgov2.pprj, Protégé Files (.pont and .pins))

For Instance: GABRA1 (instance of Gene, internal name is BGOmap_Instance_30010)

Object Symbol: GABRA1

Chromosomal Location: 5q34-q35

Synonyms:

Molecular Length (AA): 456.0

Molecular Weight (Da): 51801.0

Orientation: Plus_strand

Function Comments: GABA, the major inhibitory neurotransmitter in the vertebrate brain, mediates neuronal inhibition by binding to the GABA/benzodiazepine receptor and opening an integral chloride channel. Binds UBQLN1 (By similarity). Generally pentameric. There are five types of GABA(A) receptor chains: alpha, beta, gamma, delta, and rho. Defects in GABRA1 are a cause of juvenile myoclonic epilepsy (JME).

GO Function:
- binding
- receptor activity
- channel or pore class transporter activity
- ion transporter activity

Expression Map: C:\BGO\media\images\expression\GABRA\GABRA1.png

Produce:
- Gamma-aminobutyric-acid receptor alpha-...

Annotations:
- 2465923
- 12477932
- 2847710
- 11992121

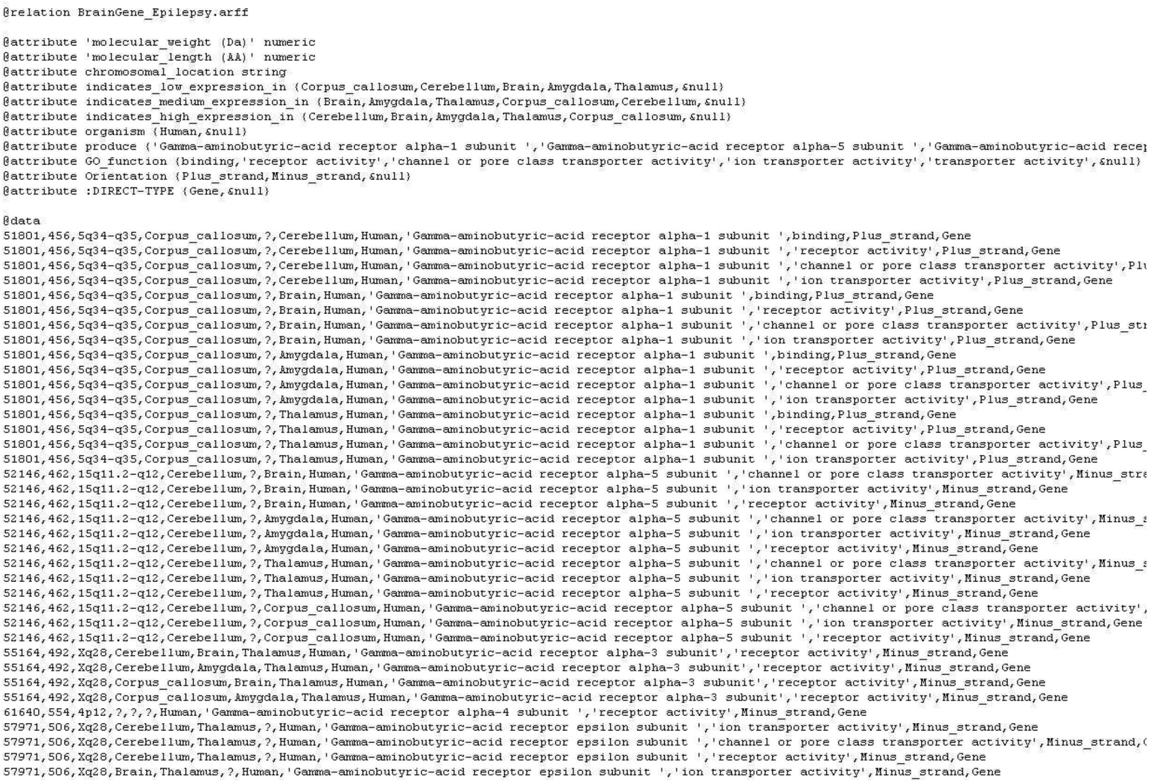
Indicates High Expression In:
- Cerebellum
- Brain

Selected Instances

molecular_w...	molecular_le...	chromosoma...	indicates_lo...	indicates_m...	indicates_hi...	organism	produce	GO_function	Orientation	:DIRECT-TYPE
51801.0	456.0	5q34-q35	All	Empty	All	Human	Gamma-aminob	All	All	All
52146.0	462.0	15q11.2-q12	All	Empty	All	Human	Gamma-aminob	All	All	All
55164.0	492.0	Xq28	All	All	All	Human	Gamma-aminob	All	All	All
61640.0	554.0	4p12	Empty	Empty	Empty	Human	Gamma-aminob	All	All	All
57971.0	506.0	Xq28	All	All	Empty	Human	Gamma-aminob	All	All	All
53667.0	465.0	4p12	Empty	Empty	Empty	Human	Gamma-aminob	All	All	All
54288.0	467.0	15q11-q13	All	Empty	Empty	Human	Gamma-aminob	All	All	All
54178.0	467.0	5q31.1-q33.1	Empty	All	All	Human	Gamma-aminob	All	All	All
46563.0	419.0	3q26.1	All	All	Empty	Human	Voltage-gated p	All	All	All
41000.0	367.0	1p36.3	Empty	Empty	Empty	Human	Voltage-gated p	All	All	All
43530.0	404.0	17p13.1	Empty	Empty	Empty	Human	Voltage-gated p	All	All	All
62459.0	545.0	9p24.2	All	All	Empty	Human	Potassium volta	Empty	All	All
51034.0	453.0	5q34	All	Empty	All	Human		All	All	All

Figure B-4 – BGO System: A set of data prepared for export using the Instance Selection Tab.

The Instance Selection plug-in developed for the ODKD Framework is used to navigate in the ontology as well as to prepare and extract biomedical data (illustrated in Figure B-4). It enables users to select and export specific data of interest such as chromosomal location or molecular sequence length, or expression patterns, which can then be analysed in a machine learning environment, such as WEKA (see export example in Figure B-5) or NeuCom to train prediction or classification models or to visualize relationship information. Such exported data on gene/protein identification numbers can also be analysed in a different manner, by standard bioinformatics software like BLAST and FASTA for revealing homology patterns for those genes/proteins of interest, as described in Benuskova (2006).

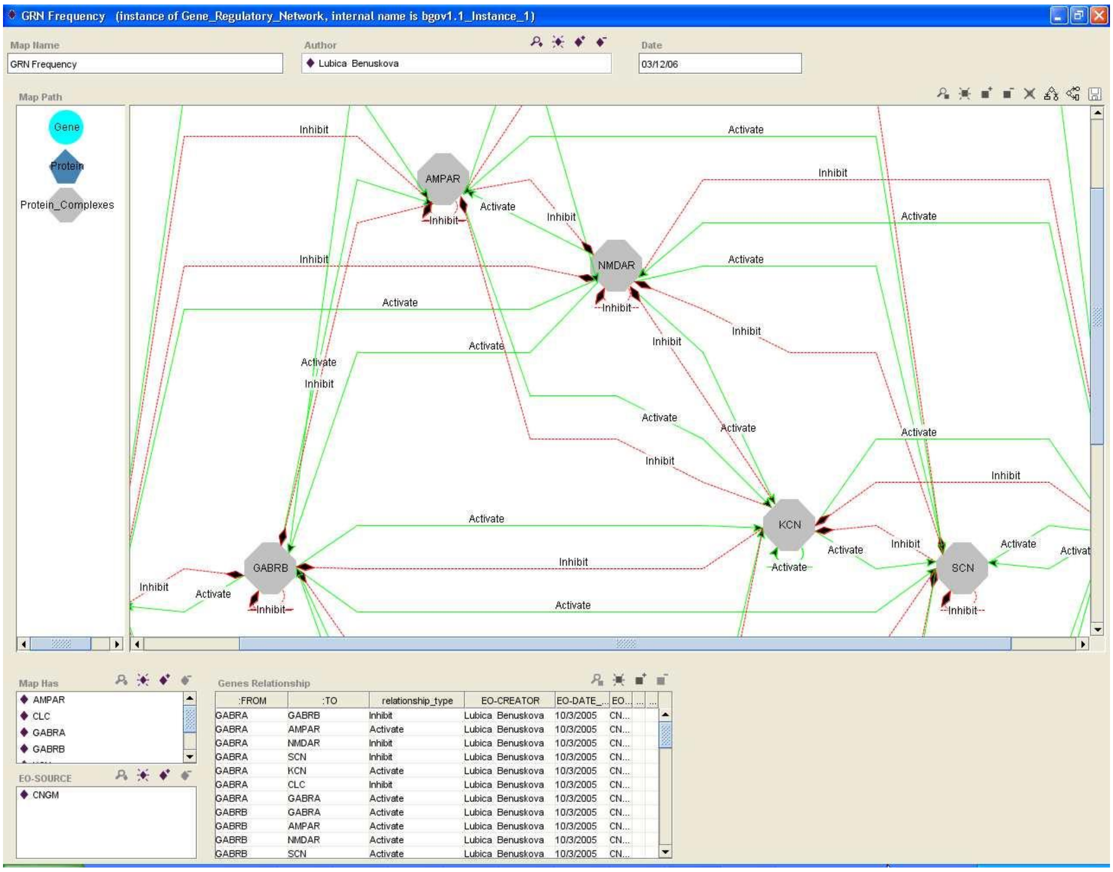


```
@relation BrainGene_Epilepsy.arff

@attribute 'molecular_weight (Da)' numeric
@attribute 'molecular_length (AA)' numeric
@attribute chromosomal_location string
@attribute indicates_low_expression_in {Corpus_callosum,Cerebellum,Brain,Amygdala,Thalamus,&null}
@attribute indicates_medium_expression_in {Brain,Amygdala,Thalamus,Corpus_callosum,Cerebellum,&null}
@attribute indicates_high_expression_in {Cerebellum,Brain,Amygdala,Thalamus,Corpus_callosum,&null}
@attribute organism {Human,&null}
@attribute produce {'Gamma-aminobutyric-acid receptor alpha-1 subunit ','Gamma-aminobutyric-acid receptor alpha-5 subunit ','Gamma-aminobutyric-acid rece
@attribute GO_function {binding,'receptor activity','channel or pore class transporter activity','ion transporter activity','transporter activity',&null}
@attribute Orientation {Plus_strand,Minus_strand,&null}
@attribute :DIRECT-TYPE {Gene,&null}

@data
51801,456,5q34-q35,Corpus_callosum,?,Cerebellum,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ',binding,Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Cerebellum,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','receptor activity',Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Cerebellum,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','channel or pore class transporter activity',Pl
51801,456,5q34-q35,Corpus_callosum,?,Cerebellum,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','ion transporter activity',Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Brain,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ',binding,Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Brain,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','receptor activity',Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Brain,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','channel or pore class transporter activity',Plus_st
51801,456,5q34-q35,Corpus_callosum,?,Brain,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','ion transporter activity',Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Amygdala,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ',binding,Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Amygdala,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','receptor activity',Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Amygdala,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','channel or pore class transporter activity',Plus_
51801,456,5q34-q35,Corpus_callosum,?,Amygdala,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','ion transporter activity',Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ',binding,Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','receptor activity',Plus_strand,Gene
51801,456,5q34-q35,Corpus_callosum,?,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','channel or pore class transporter activity',Plus_
51801,456,5q34-q35,Corpus_callosum,?,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-1 subunit ','ion transporter activity',Plus_strand,Gene
52146,462,15q11.2-q12,Cerebellum,?,Brain,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','channel or pore class transporter activity',Minus_str
52146,462,15q11.2-q12,Cerebellum,?,Brain,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','ion transporter activity',Minus_strand,Gene
52146,462,15q11.2-q12,Cerebellum,?,Brain,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','receptor activity',Minus_strand,Gene
52146,462,15q11.2-q12,Cerebellum,?,Amygdala,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','channel or pore class transporter activity',Minus_
52146,462,15q11.2-q12,Cerebellum,?,Amygdala,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','ion transporter activity',Minus_strand,Gene
52146,462,15q11.2-q12,Cerebellum,?,Amygdala,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','receptor activity',Minus_strand,Gene
52146,462,15q11.2-q12,Cerebellum,?,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','channel or pore class transporter activity',Minus_
52146,462,15q11.2-q12,Cerebellum,?,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','ion transporter activity',Minus_strand,Gene
52146,462,15q11.2-q12,Cerebellum,?,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','receptor activity',Minus_strand,Gene
52146,462,15q11.2-q12,Cerebellum,?,Corpus_callosum,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','channel or pore class transporter activity',
52146,462,15q11.2-q12,Cerebellum,?,Corpus_callosum,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','ion transporter activity',Minus_strand,Gene
52146,462,15q11.2-q12,Cerebellum,?,Corpus_callosum,Human,'Gamma-aminobutyric-acid receptor alpha-5 subunit ','receptor activity',Minus_strand,Gene
55164,492,Xq28,Cerebellum,Brain,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-3 subunit','receptor activity',Minus_strand,Gene
55164,492,Xq28,Cerebellum,Amygdala,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-3 subunit','receptor activity',Minus_strand,Gene
55164,492,Xq28,Corpus_callosum,Brain,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-3 subunit','receptor activity',Minus_strand,Gene
55164,492,Xq28,Corpus_callosum,Amygdala,Thalamus,Human,'Gamma-aminobutyric-acid receptor alpha-3 subunit','receptor activity',Minus_strand,Gene
61640,554,4p12,?,?,?,Human,'Gamma-aminobutyric-acid receptor alpha-4 subunit ','receptor activity',Minus_strand,Gene
57971,506,Xq28,Cerebellum,Thalamus,?,Human,'Gamma-aminobutyric-acid receptor epsilon subunit ','ion transporter activity',Minus_strand,Gene
57971,506,Xq28,Cerebellum,Thalamus,?,Human,'Gamma-aminobutyric-acid receptor epsilon subunit ','channel or pore class transporter activity',Minus_strand,
57971,506,Xq28,Cerebellum,Thalamus,?,Human,'Gamma-aminobutyric-acid receptor epsilon subunit ','receptor activity',Minus_strand,Gene
57971,506,Xq28,Brain,Thalamus,?,Human,'Gamma-aminobutyric-acid receptor epsilon subunit ','ion transporter activity',Minus_strand,Gene
```

Figure B-5 – The selected set of data exported as arff format to be used in WEKA.

2.1. Gene Regulatory Network Visualization – GRN Graph Widget

The GRN graph is a network visualization tool designed to represent gene regulatory networks and link the network to instances and relationships between the instances from the ontological model. It is also used in the biomedical tool as an alternative knowledge acquisition tool to forms and other editing Protégé plug-ins as it enables direct editing from its interface.



GRN Frequency (instance of Gene_Regulatory_Network, internal name is bgov1.1_Instance_1)

Map Name: GRN Frequency

Author: Lubica Benuskova

Date: 03/12/06

Map Has: AMPAR, CLC, GABRA, GABRB

EO-SOURCE: CNGM

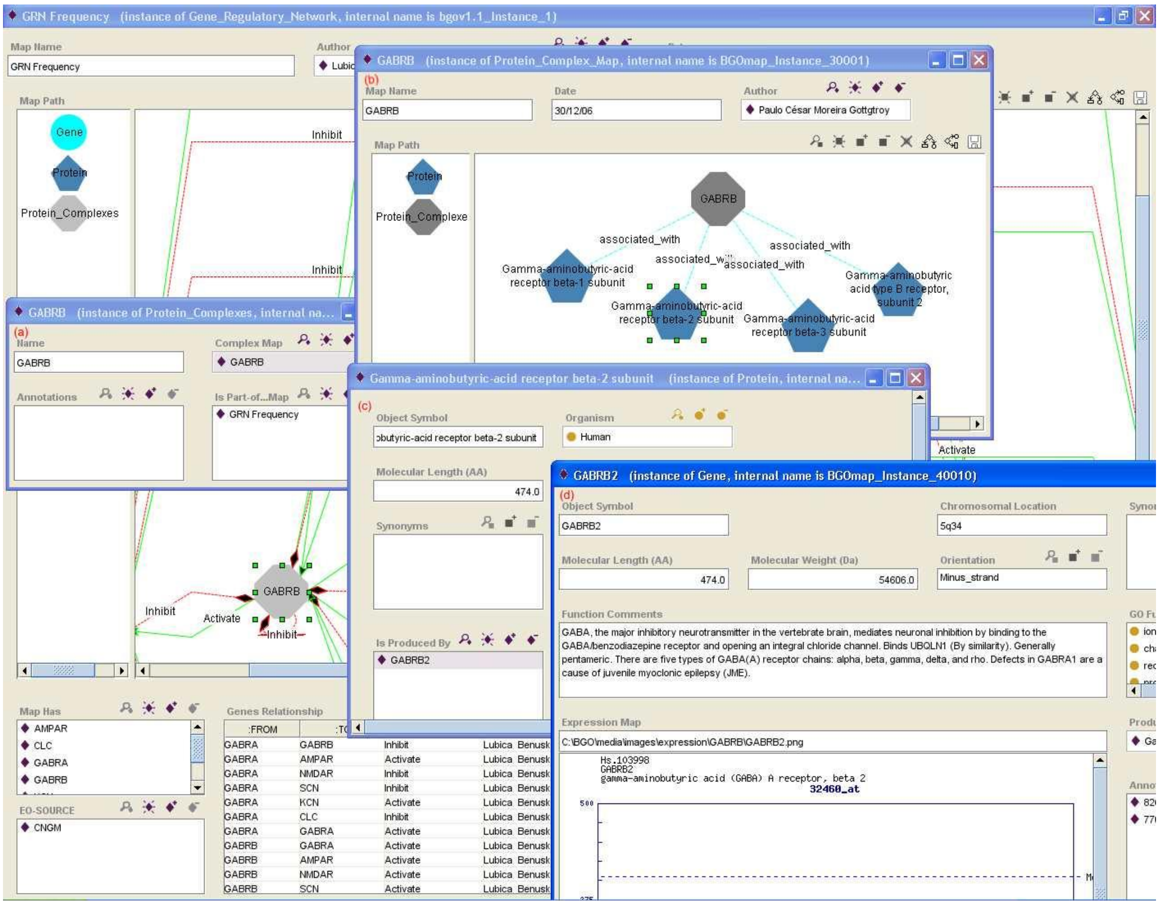
Genes Relationship

:FROM	:TO	relationship_type	EO-CREATOR	EO-DATE_...	EO...	...	...
GABRA	GABRB	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	AMPAR	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRA	NMDAR	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	SCN	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	KCN	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRA	CLC	Inhibit	Lubica Benuskova	10/3/2005	CN...		
GABRA	GABRA	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	GABRA	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	AMPAR	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	NMDAR	Activate	Lubica Benuskova	10/3/2005	CN...		
GABRB	SCN	Activate	Lubica Benuskova	10/3/2005	CN...		

Figure B-6 – BGO System: A screenshot of a GRN graphical representation of the gene regulatory network presented in Figure 6-14.

The GRN graph is linked to the ontology by the Gene Regulatory Network concept, which is a sub-class of Bioinformatics Maps within the Brain Gene Ontology. All concepts and relationships allowed are specified in the ontology in accordance with the knowledge needed to represent gene regulatory networks. This ontological representation makes possible the visualization and navigation of all concepts related to the network. Figure B-7 and Figure B-8, for example, show the relationships of the protein complex GABRB (a) which is composed of other proteins

(b) such as the ‘Gamma-aminobutyric-acid receptor beta-2 subunit’ (c) which is produced by the gene GRABRB2 (d).



GRN Frequency (instance of Gene_Regulatory_Network, internal name is bgov1.1_Instance_1)

Map Name: GRN Frequency

Author: Lubic

GABRB (instance of Protein_Complexes, internal na...

(a)

Name: GABRB

Complex Map: GABRB

Annotations:

Is Part-of...Map: GRN Frequency

GABRB (instance of Protein_Complex_Map, internal name is BGOmap_Instance_30001)

(b)

Map Name: GABRB

Date: 30/12/06

Author: Paulo César Moreira Gottgtroy

Gamma-aminobutyric-acid receptor beta-2 subunit (instance of Protein, internal na...

(c)

Object Symbol: obutyric-acid receptor beta-2 subunit

Organism: Human

Molecular Length (AA): 474.0

Synonyms:

Is Produced By: GABRB2

GABRB2 (instance of Gene, internal name is BGOmap_Instance_40010)

(d)

Object Symbol: GABRB2

Chromosomal Location: 5q34

Molecular Length (AA): 474.0

Molecular Weight (Da): 54606.0

Orientation: Minus_strand

Function Comments: GABA, the major inhibitory neurotransmitter in the vertebrate brain, mediates neuronal inhibition by binding to the GABA/benzodiazepine receptor and opening an integral chloride channel. Binds UBQLN1 (By similarity). Generally pentameric. There are five types of GABA(A) receptor chains: alpha, beta, gamma, delta, and rho. Defects in GABRA1 are a cause of juvenile myoclonic epilepsy (JME).

Expression Map: C:\BGO\media\images\expression\GABRB\GABRB2.png

```
Hs.103998
GABRB2
gamma-aminobutyric acid (GABA) A receptor, beta 2
32460_at
```

Map Has:
- AMPAR
- CLC
- GABRA
- GABRB

EO-SOURCE:
- CNGM

Genes Relationship

:FROM	:T		
GABRA	GABRB	Inhibit	Lubica Benusk
GABRA	AMPAR	Activate	Lubica Benusk
GABRA	NMDAR	Inhibit	Lubica Benusk
GABRA	SCN	Inhibit	Lubica Benusk
GABRA	KCN	Activate	Lubica Benusk
GABRA	CLC	Inhibit	Lubica Benusk
GABRA	GABRA	Activate	Lubica Benusk
GABRB	GABRA	Activate	Lubica Benusk
GABRB	AMPAR	Activate	Lubica Benusk
GABRB	NMDAR	Activate	Lubica Benusk
GABRB	SCN	Activate	Lubica Benusk

Figure B-7 – BGO System: A screenshot of a GRN linked to other knowledge available in the Brain Gene Ontology.

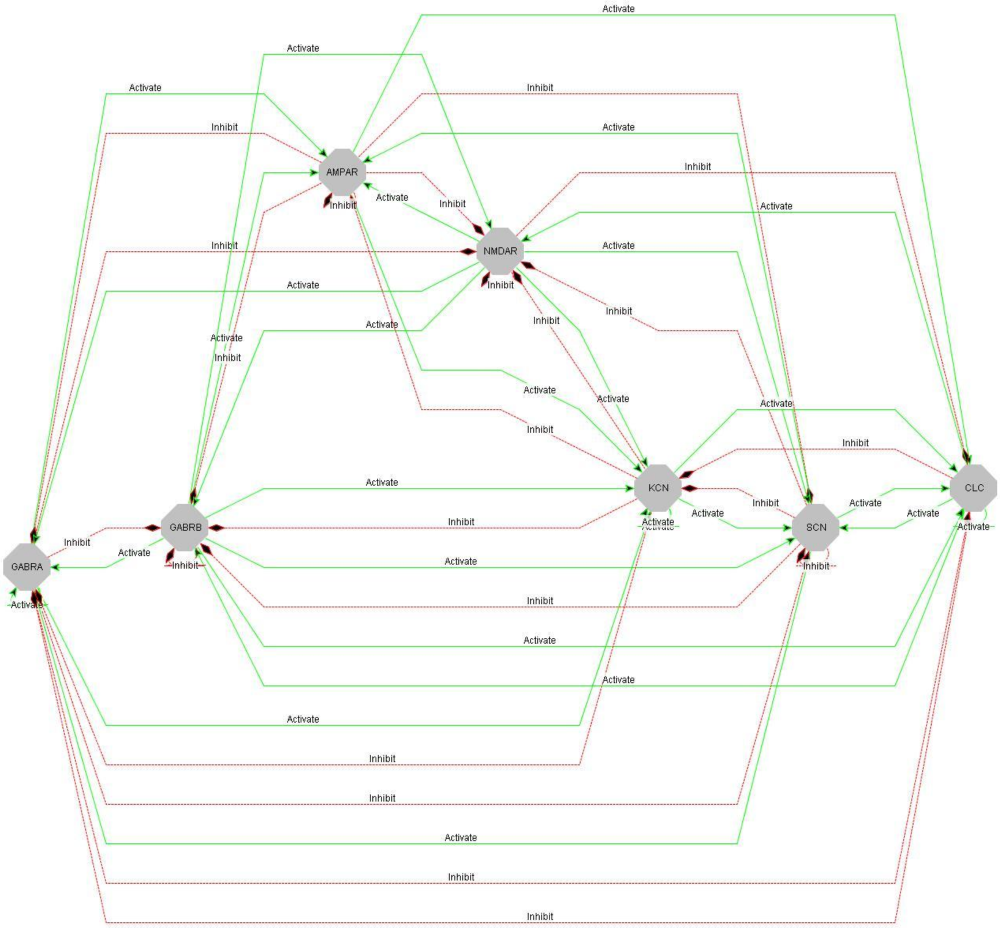
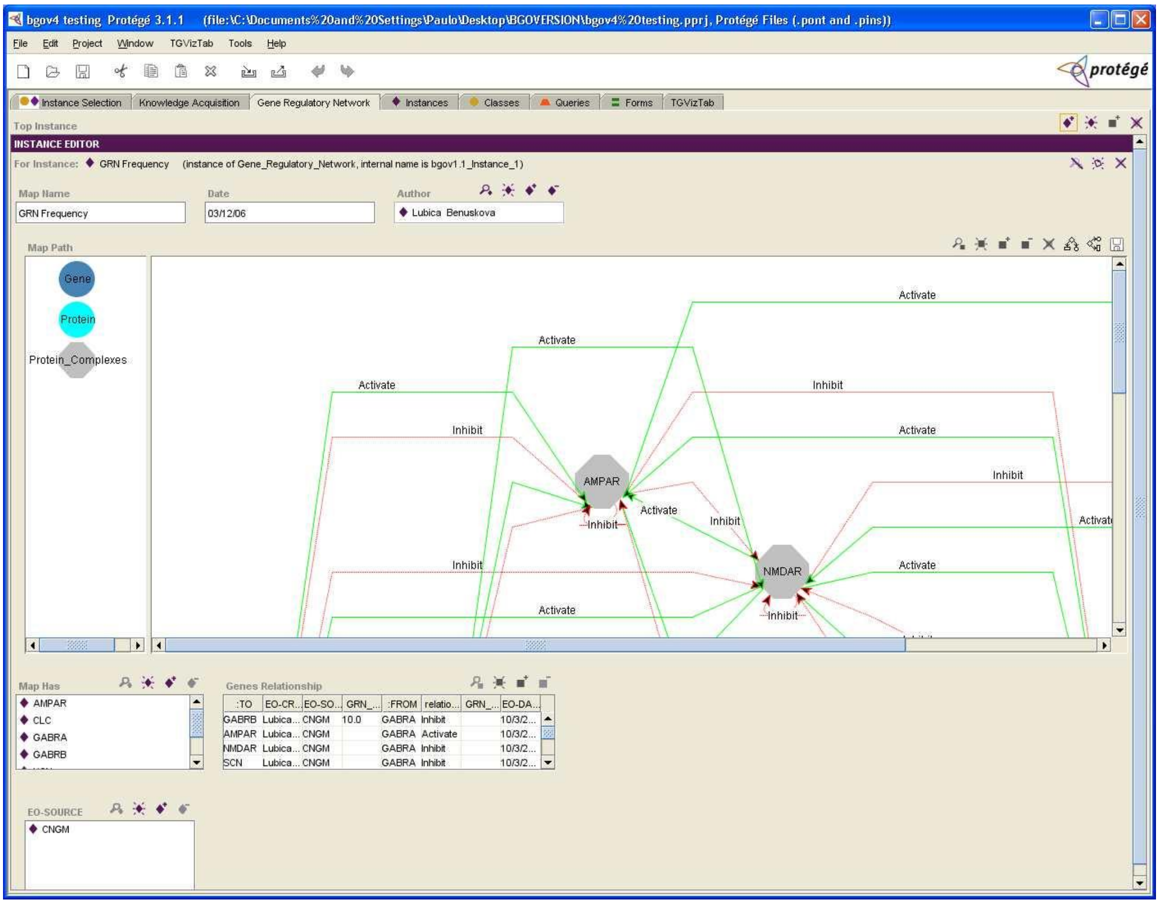


Figure B-8 – A gene regulatory network exported from the GRN Graph widget showing different relationship colours, style and text based on results acquired from a CNGM simulation.

2.2. Gene Regulatory Network Tab – GRN Tab

The GRN tab is responsible for the creation and editing of gene regulatory networks. It includes a set of features that are primarily designed to import the results of a computational neuro-genetic simulation. The tab is built as a novel Protégé plug-in and uses its application program interface to interact with the Brain-Gene Ontology and the GRN Graph widget to display the regulatory networks (as shown in Figure

B-8). The knowledge acquisition feature allows for the creation and editing of new or existent gene regulatory networks (Figure B-9). The figure below also shows some metadata which is used to annotate the ontology such as the slot Author (top middle) and EO-Source (left bottom) which indicates that these GRN was originally acquired from CNGM.



bgov4 testing Protégé 3.1.1 (file:\C:\Documents%20and%20Settings\Paulo\Desktop\BGOVERSION\bgov4%20testing.pprj, Protégé Files (.pont and .pins))

INSTANCE EDITOR

For Instance: ◆ GRN Frequency (instance of Gene_Regulatory_Network, internal name is bgov1.1_Instance_1)

Map Name: GRN Frequency

Date: 03/12/06

Author: ◆ Lubica Benuskova

Map Has:
- ◆ AMPAR
- ◆ CLC
- ◆ GABRA
- ◆ GABRB

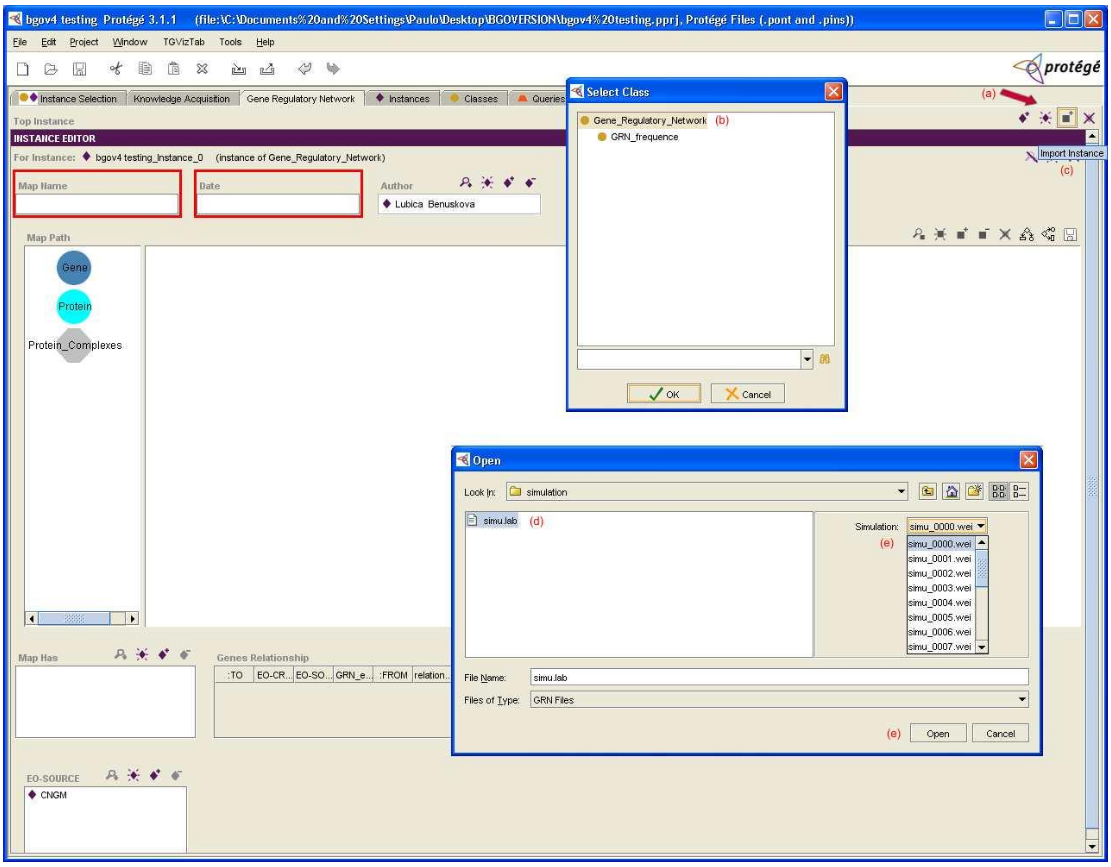
Genes Relationship

:TO	EO-CR...	EO-SO...	GRN_...	:FROM	relatio...	GRN_...	EO-DA...
GABRB	Lubica...	CNGM	10.0	GABRA	Inhibit		10/3/2...
AMPAR	Lubica...	CNGM		GABRA	Activate		10/3/2...
NMDAR	Lubica...	CNGM		GABRA	Inhibit		10/3/2...
SCN	Lubica...	CNGM		GABRA	Inhibit		10/3/2...

EO-SOURCE: ◆ CNGM

Figure B-9 – BGO System: A screenshot of the Gene Regulatory Network tab with data from the CNGM simulation presented in the section 6.6.1.

A CNGM simulation is imported in a sequence of steps as described in the Chapter 6 in the section 6.6.2. The following figure illustrates some of these steps.



bgov4 testing Protégé 3.1.1 (file:\C:\Documents%20and%20Settings\Paulo\Desktop\BGOVERSION\bgov4%20testing.pprj, Protégé Files (.pont and .pins))

For Instance: bgov4 testing_Instance_0 (instance of Gene_Regulatory_Network)

Author: Lubica Benuskova

Look In: simulation

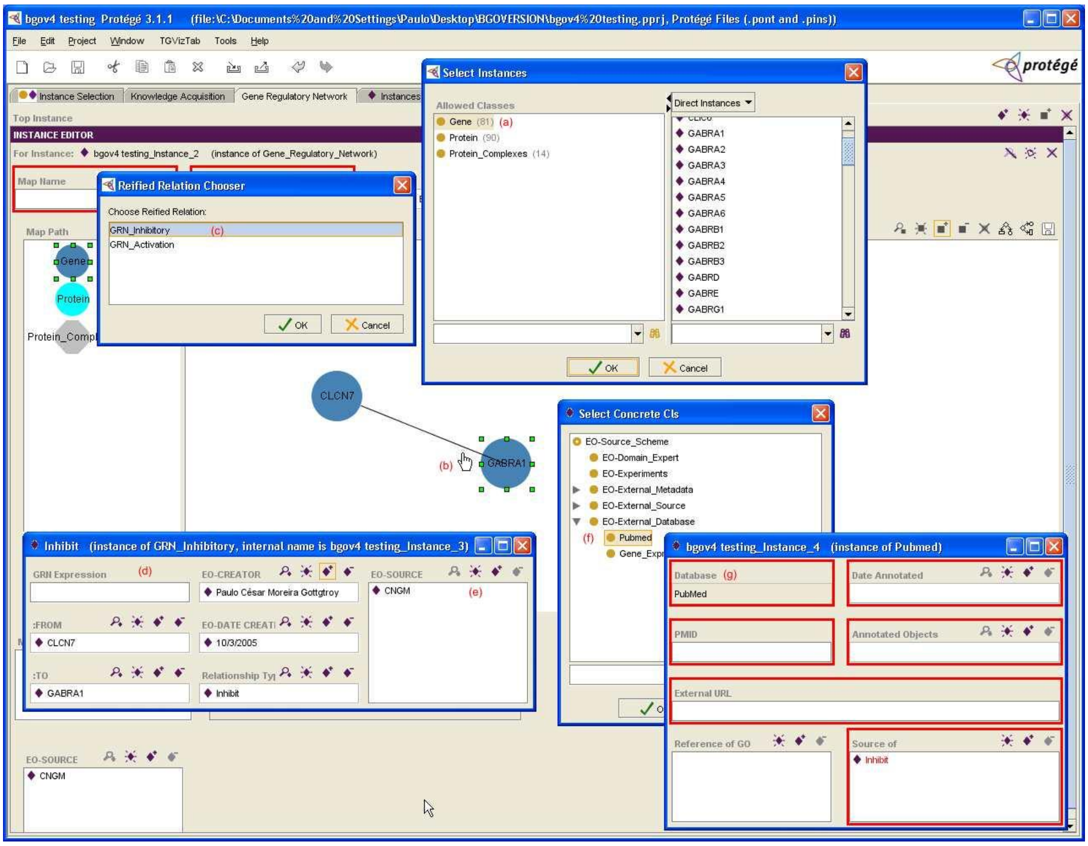
Simulation: simu_0000.wei

File Name: simu.lab

Files of Type: GRN Files

Figure B-10 – BGO System: The Gene Regulatory Network import process.

The GRN tab also enables the editing of a regulatory network by adding new genes, proteins, and protein complexes, or including information about the network such as annotations and uncertainty properties of a relationship. The Figure B-11 illustrates some of these capabilities.



For Instance: bgov4 testing_Instance_2 (instance of Gene_Regulatory_Network)

Choose Reified Relation:

- GRN_Inhibitory (c)
- GRN_Activation

Allowed Classes:

- Gene (81) (a)
- Protein (90)
- Protein_Complexes (14)

Direct Instances:

- GABRA1
- GABRA2
- GABRA3
- GABRA4
- GABRA5
- GABRA6
- GABRB1
- GABRB2
- GABRB3
- GABRD
- GABRE
- GABRG1

Select Concrete Cls:

- EO-Source_Scheme
  - EO-Domain_Expert
  - EO-Experiments
  - EO-External_Metadata
  - EO-External_Source
  - EO-External_Database
    - Pubmed
    - Gene_Expr

Inhibit (instance of GRN_Inhibitory, internal name is bgov4 testing_Instance_3)

GRN Expression (d):

EO-CREATOR: Paulo César Moreira Gottgtroy

EO-SOURCE: CNGM (e)

:FROM: CLCN7

EO-DATE CREATI: 10/3/2005

:TO: GABRA1

Relationship Typ: Inhibit

EO-SOURCE: CNGM

bgov4 testing_Instance_4 (instance of Pubmed)

Database (g): PubMed

Date Annotated:

PMID:

Annotated Objects:

External URL:

Reference of GO:

Source of: Inhibit

Figure B-11 – BGO System: A sample of a GRN editing process: (a)the CLCN and GABRA1 genes are created (b) a relationship is created between these genes (c)an inhibitory behaviour is selected (d) the new inhibitory relationship is opened (e) an annotation is added (f) the source of the annotation is selected (g) a PubMed annotation is created.

Appendix C

1. NeuCom - A Neuro-computing Decision Support Environment

NeuCom is self-programmable, learning and reasoning computer environment based on connectionist (Neurocomputing) modules. NeuCom learns from data, thus evolving new connectionist modules. The modules can adapt to new incoming data in an on-line incremental, life-long learning mode, and can extract meaningful rules that would help people discover new knowledge in their respective fields. NeuCom is based on the theory of Evolving Connectionist Systems (ECOS) (Kasabov, 2002).

NeuCom can be used to solve complex problems. Such problems are clustering, classification, prediction, adaptive control, data mining and pattern discovery from databases in a multidimensional, dynamic and possibly changing data environment. Applications span in all areas of Science, Engineering, Medicine, Bio-informatics, Business, Arts and Design, Education.

There are several algorithms available in NeuCom. Evolving Classification Algorithm is one of the algorithms based on the theory of Evolving Connectionist Systems. The following table exemplifies the generic ECF algorithm.

Evolving Classification Algorithm (ECF):

1. Enter the current input vector from the data set (stream) and calculate the distances between this vector and all rule nodes already created using Euclidean distance (by default). If there is no node created, create the first one that has the coordinates of the first input vector attached as input connection weights.

2. If all calculated distances between the new input vector and the existing rule nodes are greater than a max-radius parameter Rmax, a new rule node is created. The position of the new rule node is the same as the current vector in the input data space and the radius of its receptive field is set to the min-radius parameter Rmin; the algorithm goes to step 1; otherwise it goes to the next step.

3. If there is a rule node with a distance to the current input vector less then or equal to its radius and its class is the same as the class of the new vector, nothing will be changed; go to step 1; otherwise:

4. If there is a rule node with a distance to the input vector less then or equal to its radius and its class is different from those of the input vector, its influence field should be reduced. The radius of the new field is set to the larger value from the two numbers: distance minus the min-radius; min-radius. New node is created as in 2 to represent the new data vector.

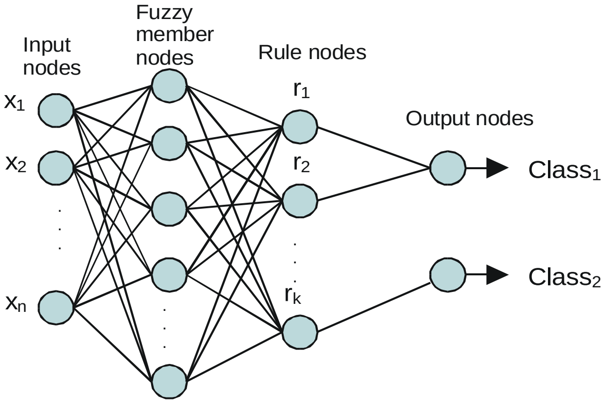
5. If there is a rule node with a distance to the input vector less than or equal to the max-radius, and its class is the same as of the input vector's, enlarge the influence field by taking the distance as a new radius if only such enlarged field does not cover any other rule nodes which belong to a different class; otherwise, create a new rule node in the same way as in step 2, and go to step 1.

Recall procedure (classification of a new input vector) in a trained ECF :

1. Enter the new vector in the ECF trained system; If the new input vector lies within the field of one or more rule nodes associated with one class, the vector is classified in this class;

2. If the input vector lies within the fields of two or more rule nodes associated with different classes, the vector will belong to the class corresponding to the closest rule node.

Example: 2-class ECF model



3. If the input vector does not lie within any field, then take m highest activated by the new vector rule nodes, and calculate the average distances from the vector to the nodes with the same class; the vector will belong to the class corresponding to the smallest average distance.

NeuCom can be used either as a decisions support system (DSS), where users specify their task and define data to be used, in order to obtain a solution, or - as a DSS development environment for building sophisticated problem oriented intelligent DSS. The end users in the former case are people who have never programmed computers, but have databases available and need a decision to be made based on existing data and/or human knowledge. In the latter case users are professional system developers who can develop DSS for various applications in collaboration with experts in the field.

Appendix D

The following publications are examples of publications and or researches which in some form are derived or are referencing this thesis.

Biomedical Research

- Calle, G., García, M., Maojo, V., Brookes, A., Voets, D. & Prosperi, M. (2007). INFOBIOMED: Structuring European Biomedical Informatics to Support Individualised Healthcare. Report: Data Interoperability and Management. Retrieved 01/09/07, from http://www.infobiomed.org/

- Rüping, S. (2007). Advancing Clinico-Genomic Clinical Trials on Cancer: Open Grid Services for improving Medical Knowledge Discovery. Report: Consolidated Requirement Analysis for Data Mining, Analysis and the Visualization Environment. Retrieved 01/03/07, from http://www.eu-acgt.org/documents/public-deliverables.html

Biomedical Thesis

- Gudivada, R. (2007). Discovery and Prioritization of Biological Entities Underlying Complex Disorders by Phenome - Genome Network .PhD Thesis. University of Cincinnati, Department of Biomedical Engineering. Retrieved 01/07/09, from http://etd.ohiolink.edu/send-pdf.cgi/GUDIVADA%20RANGA%20CHANDRA.pdf?ucin1195161740

- Coulet, A. (2008). Construction et utilisation d'une Base de Connaissances pharmacogenomique pour l'integration de donnees et la decouverte de connaissances.Doctorat de l'universite Henri Poincare - Nancy 1. Retrieved 01/07/09, from http://hal.archives-ouvertes.fr/docs/00/33/24/07/PDF/these_adrien.pdf

- Jain, V. (2008). Integrative approaches to modelling and knowledge discovery of molecular interactions in bioinformatics. PhD Thesis, Auckland University of Technology. Retrieved 01/10/08, from http://aut.researchgateway.ac.nz/bitstream/10292/439/5/JainV_a.pdf.

- Wang, Y. (2007). Ontology Engineering: The Brain Gene Ontology Case Study. Master Thesis. Auckland University of Technology, Auckland.

Publications

- Kasabov, N., Jain, V. & Benuskova, L. (2007). Integrating evolving brain-gene ontology and connectionist-based system for modeling and knowledge discovery, Neural Networks, Volume 21, Issues 2-3, Advances in Neural Networks Research: IJCNN '07, 2007 International Joint Conference on Neural Networks IJCNN '07, March-April 2008, Pages 266-275, ISSN 0893-6080.

- Coulet, A., Smaïl-Tabbone, M., Benlian, P., Napoli, A. & Devignes, D. (2008). Ontology-guided data preparation for discovering genotype-phenotype relationships. BMC Bioinformatics 2008, 9(Suppl 4).

Ontology Engineering Projects

Ontology Engineering Thesis

- Duane, D. (2006). Design, Implementation and Testing of A Common Data Model Supporting Autonomous Vehicle Compatibility And Interoperability. PhD Thesis, Naval Postgraduate School, USA. . Retrieved 01/07/06, from http://www.stormingmedia.us/62/6207/A620754.html

- Tseng, C. (2007). Evolutionary Mining of Association Rules with Ontological Information. PhD Thesis, I-Shou University, Twain. Retrieved 01/07/09, from http://ethesys.isu.edu.tw/ETD-db/ETD-search/view_etd?URN=etd-0615107-061820

- Del Rey, D. (2008). Un modelo de integración y preprocesamiento de información distribuida basado en ontologías. PhD Thesis, Universidad Politecnica de Madri. Retrieved 01/07/09, from http://oa.upm.es/1052/1/DAVID_PEREZ_DEL_REY.pdf

- Gudivada, R. (2007). Discovery and Prioritization of Biological Entities Underlying Complex Disorders by Phenome - Genome Network Integration. PhD Thesis, University of Cincinnati.

- Rui, H. (2005). KDD-Based Automatic Knowledge Acquisition and Its Applications. Master Thesis.

- Novacek, V. (2007). Inference Support for Ontology Acquisition. Master Thesis.

Ontology Engineering Publications

- Perez-Rey, D. & Anguita, A. (2009). OntoDataClean: Ontology-Based Integration and Preprocessing of Distributed Data. Publisher: Spring Verlag "Lecture Notes in Computer Sciences: Artificial Intelligence in Medicine"

- Nigro, H., Cisaro, S. & Xodo, H. (2006). Data Mining With Ontologies: Implementations, Findings and Frameworks: Information Science Reference.

- Pinto, F. & Santos, M. (2009). Considering Application Domain Ontologies for Data Mining. WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS.

- Kasabov, N. (2009). Integrative connectionist learning systems inspired by nature: current models, future trends and challenges. Natural Computing: an international journal 8, 2 (Jun. 2009), 199-218. DOI= http://dx.doi.org/10.1007/s11047-008-9066-z

- Nogueira, B.M., Santos, T.R.A. & Zarate, L.E (2007). Comparison of Classifiers Efficiency on Missing Values Recovering: Application in a Marketing Database with Massive Missing Data. Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on March 1 2007-April 5 2007 Page(s):66 - 72.

- Schlicht, A. (2007). Improving the Usability of Large Ontologies by Modularization 1 Problem Definition. Knowledge Web PhD Symposium 2007.

- Ghorbel, H.; Bahri, A.; Bouaziz, R. (2010). Fuzzy ontologies building method: Fuzzy OntoMethodology. Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American.

- Mohemad, R.; Noor, N.M.M.; Hamdan, A.R.; Othman, Z.A. (2010). Ontological-based for supporting multi criteria decision-making. Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on.

- Wen-Yang, Lin. (2006). Ontology-Based Data Mining -A Case in Multidimensional Association Mining. National University of Kaohsiung.

- Zarate, L.E.; Nogueira, B.M.; Santos, T.R.A.; Song, M.A.J. (2006). Techniques for Missing Value Recovering in Imbalanced Databases: Application in a Marketing Database with Massive Missing Data. Systems, Man and Cybernetics, 2006.

- Novacek, V. (2007). A Non-traditional Inference Paradigm for Learned Ontologies. In Proceedings of ESWC 2007 PhD Symposium. Innsbruck : CEUR Workshop Proceedings, 2007. pp. 57-62. Innsbruck, Austria.

- Pinto, F. M. and Santos, M. F. 2009. Ontological assistance for knowledge discovery in databases process. In Proceedings of the WSEAES 13th international Conference on Computers (Rodos, Greece, July 23 - 25, 2009). N. E. Mastorakis, V. Mladenov, Z. Bojkovic, S. Kartalopoulos, and A. Varonides, Eds. Recent Advances In Computer Engineering. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, 453-458.

- Khilwani, N., Harding, J. & Choudhar, A. (2009). Semantic web in manufacturing. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture.

- Bouamrane, M., Rector, A. & Hurrell, M. (2009). Semi-automatic generation of a patient preoperative knowledge-base from a legacy clinical database. On the Move to Meaningful Internet Systems: OTM 2009. Lecture Notes in Computer Science, 2009, Volume 5871/2009, 1224-1237.

- Wu C., Lin, W., Tseng M. & Wu, C. (2007). Ontology-Incorporated Mining of Association Rules In Data Warehouse. Journal of Internet Technology Volume: 8 Issue: 4 Pages: 477-483

These studies serve to reiterate that there are a number of research opportunities in this work that can be further explored. Those studies listed above all utilize parts of the work developed in this thesis and extend or build on it in different ways. Further and more detailed analysis can be undertaken to evaluate these research efforts; however, we believe that this thesis represents a complete iteration and achieved the originally proposed aims of the research.