Recommender System

Case iKnow: Composition of the required building blocks for the development of recomender systems

Ben Plessers

promotor :
Prof. Jeanne SCHREURS

Master thesis submitted to obtain the degree of Master of Management

universiteit hasselt

Preface

This master thesis is the culmination of my one year Master of Management, major Management Information Systems study at the University of Hasselt. This will we be the second Master I will obtain. After already obtaining a Master degree in Applied Economics, major Accountancy and Finance, the studies of Management Information Systems were very different concerning the content, yet very interesting.

When related systems with software agents, like recommender systems, where handled in various courses, they could always draw my attention. Also when encountering them during surfing the Web, I found them fascinating. This master thesis gave me the chance to take a closer look and to analyze these recommender systems.

The result of this master thesis is reached with the help from many people.
Firstly, I would like to thank professor Jeanne Schreurs for her help, suggestions, constructive feedback and introducing me to iKnow. This made me able to finish this master thesis with the following result.

Next, I would also like to express my gratitude to Dirk van Hyfte and iKnow. Not only did he make it possible to do a case study in cooperation with the company, he also provided a lot of help, fast and excellent feedback and extra information that could be used in this master thesis.

Finally, I am also grateful for the help and the unflagging moral support from friends and family, not only during the making of this master thesis, but during the whole five years of my studies.

Abstract

In a period of time where **knowledge management** is a very important issue and the problem of **information overload** is growing, there is a need for systems that can help to obtain information, knowledge and data.

Knowledge management is all about getting the right knowledge, in the right place, at the right time. It is based on the idea that an organisation’s most valuable resource is the knowledge of their people. Raw information may be widely available to a lot of organisations, but only some organisations will be able to convert the information into relevant knowledge and to use this knowledge to achieve their goals.

Information and communication technology has made information abundant. Because of the Internet you can basically get any information you might desire in seconds. However, the reliability, usefulness, and timeliness of the information is difficult to verify. There is an existence of excess low quality information that can lead to inefficient decision making.

To help organisations in the process of knowledge management and to overcome the problem of information overload, recommender systems can be used.

A **recommender system** is a software system whose main goal is to aid in the social collaborative process of indicating or receiving indication, when the number of options is enormous. They supply users with information in order to help them make decisions or to solve problems, without users have to search for it themselves.

Using recommender systems will lead to more **efficient knowledge acquisition**. They can provide a reduction in this very time-consuming process and this can lead to a decrease in costs for enterprises for which knowledge acquisition is important.

This master thesis will propose an architecture for a recommender system. The recommender system proposed, will be based on the technology of the **Semantic Web** and **Linked Data**. It will take advantage of the inference mechanisms involving semantic description developed in the Semantic Web. This will allow the recommender system to discover hidden semantic associations by exploring the knowledge and structure of the ontological model. Using Linked Data will make the recommender system able to create links between data, and by doing so suggesting more relevant information. In the case study performed in this master thesis, Linked Data from the Linking Open Data project was used.

Ontologies have become the cornerstone in the Semantic Web for two reasons. Firstly, because these conceptualizations represent formally a specific domain, ontologies enable inference processes to discover new knowledge from the formalized information. Secondly, ontologies make it easier to automate knowledge sharing, by allowing easy reuse between users and software agents. The ontologies that will supply the Semantic Web are and will be developed, managed and supported by practice communities.

Two other important parts of the recommender system proposed in this master thesis are the Resource Description Framework (RDF) and the SPARQL Protocol and RDF Query Language. **RDF** is a W3C standard format for storing arbitrary data on the Web and elsewhere. In other words, it is a framework for representing information in the Web. It was created as a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information. RDF provides a machine-readable way to say anything about anything.
SPARQL provides a query language that can be utilized to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware.

In the second part of this master thesis, a **case study** is executed in cooperation with the company **iKnow**. This case study was performed in order to get practical insight in recommender systems. The goal of this case study with iKnow is to get an understanding of the way these software systems are built and how they work. A step by step analysis of the process that a recommender system will adopt, reveals the architecture behind these recommender systems. The case study will also further clarify topics that were explained in the study of literature by giving practical examples.

The last part of the master thesis combines the results from the case study with the results from the study of literature in order to make a **proposition of the architecture for a recommender system**. The proposed recommender system will make use of the Semantic Web and Linked Data in order to suggest more relevant information. A Text Mining Module will analyze unstructured text and give input for a Recommender Module. This Recommender Module will provide the user with timely, relevant suggestions.

Table of contents

List of figures

Chapter 1: Introduction to the subject and defining a problem statement

The aim of this first chapter is to introduce and generally present the subject of this master thesis and its relevance. This chapter will also point out the purpose and limitations, as well as the central research question and the research sub-questions. Next to these topics, the research approach is explained and finally the structure of the master thesis is explicated.

1.1 Problem statement

In an age of globalisation and information explosion, creating and sharing market information and market knowledge is crucial for organisations. There is a significant information overload that can lead to inefficient decision-making. Organisations need to apply efficient methods when searching and sharing information and knowledge. The information and knowledge that is required for decision-making needs to be available and accessible in an easy way for the right person, at the time the information or knowledge is needed. The Internet is a source of a gigantic amount of data and information, but this data and information is often obscure and contains less relevant items. It is very important that correct and relevant supporting information reaches the right persons.

This is why there is the need to transform an enormous amount of unstructured information into clarified structured information. For this purpose, there already have been developed a number of software applications that can filter the most important and relevant information out of the mass of information. An example of such a software application that is capable of searching and filtering information and knowledge is a *recommender system*. The recommender system helps the user to make a decision when huge amounts of options are available or to make relevant suggestions. This is done by providing the user with information and data that is useful to make a decision.

There already exist many kinds of recommender systems. The most familiar recommender systems are the one use in E-commerce. These software systems make suggestions for products a customer might like to buy. However, these systems are not capable to recommend related information and data to professionals in real-time. Recommender systems that can make differences for organisations by assisting professional in this search for information and knowledge, are very different. These software systems need to suggest related articles, papers, publications, etc. to their users. There already exist various types recommender systems already existing that comply to this need, yet they are not yet widespread.

Recommender systems that meet the terms for modern and future requirements, need to include the latest technologies. This master thesis will give an overview of the existing recommender system and these latest technologies in order to propose a description for a modern recommender system.

1.2 Limitations

This master thesis has some limitations. The goal will not be to create a fully functional recommender system. There will not be any programming of software in this master thesis. The result of this master thesis will be to make a proposition of the architecture for a modern recommender system.

1.3 Purpose of research

The purpose of this research is, in the first instance, to explain what recommender systems are and why they are needed. The second purpose of this master thesis is to show how recommender systems are built and how their architectures look like. The final purpose of this master thesis is to combine the results from the study of literature and the case study in order to make a proposal for the architecture of a recommender system.

1.4 Research questions

1.4.1 Central research question

In order to meet the research objective, formulating research questions is necessary. The following question is central in this master thesis:

A recommender system that delivers just in time and task relevant information, will help professionals improving their knowledge-intensive tasks. What are the required technical elements of its architecture?

1.4.2 Research sub-questions

In order to help find the answer to the central research question, some sub-questions have been formulated:

1) What is a recommender system and what are its functions?
2) Why is a recommender system needed?
3) What are the technological building blocks of recommender systems?
4) How are recommender systems built in detail?
5) How does the architecture of a modern/next generation recommender system looks like?

1.5 Research approach

This part will justify the research approach that will be used in this master thesis. A research approach is a general plan for the method of operation that is employed, when trying to answer the research questions. The research approach will also take into account the purpose of the research.

In the first part of the master thesis, the existing literature about recommender systems and related topics will be analyzed. This will provide an answers on the "what", "why", "when" and "how" questions for recommender systems. The literature that will be used in this master thesis, was primarily obtained by searching the Web, the library of the University of Hasselt and via EBSCO Host. This literature contains papers published by reliable instances or articles published by scientific (information system) magazines. The goal is to provide a short, but an as complete as possible literature review of recommender systems and related topics by using the leading and most influential authors in these domains.

In the second part of the master thesis, practical applications will be studied in cooperation with a best practice company iKnow. This case study will make use of several topics described in the literature review. The case study will be a part by part analysis of the architecture of a recommender system. It will examine its building blocks by applying them in some real situations. Carrying out this case study will clarify how a recommender system works and will give a practical view of several topics discussed in the literature review.

In the third part, the literature review from the first part and the case study from the second part will be combined to make a proposal for a recommender system. This proposal for a recommender system will be based on the findings throughout the whole research carried out in this master thesis.

1.6 Structure of the master thesis

Chapter 2 of this master thesis will give a general view of knowledge management and the problem of information overload. This chapter gives reasons why a recommender system could or should be introduced. In the third chapter, the different technical building blocks of a recommender system that need to be clarified before a further analysis of these systems can be executed, are explained. Chapter 4 explains what recommender systems are and gives examples them. This chapter gives a case example by summarizing a paper about a complete recommender system that will give an introduction to the case study. The fifth chapter describes the case study with iKnow. Chapter 6 gives a proposition of the architecture for a recommender system by combining the results of the literature review with the case study. In the final chapter, chapter 7, the conclusions of this master thesis are drawn.

Chapter 2: Knowledge management and information overload

2.1 Knowledge management

The first part of the literature review will explain knowledge management. Knowledge management is at the foundation of the subject of this master thesis, because recommender systems are becoming an important alternative to support knowledge acquisition (X,2010).

We will start by defining **knowledge**. Knowledge will be an important term during the whole master thesis. Mekhilef, et al. (2004) use the following definition:
"Knowledge is the combination of data and information, to which is added expert opinion, skills and experience, to result in a valuable asset which can be used to aid decision making. Knowledge may be explicit and/or tacit, individual and/or collective" (Chaffey, 2009).
Knowledge is derived from information but, it is richer and more meaningful than information. "Knowledge includes familiarity, awareness and understanding gained through experience or study and results from making comparisons, identifying consequences, and making connections" (NHS, 2005). Some experts also include wisdom and insight in their definition of knowledge (NHS, 2005).

Next, we will give several definitions of **knowledge management***.* The literature concerning knowledge management provides a lot of definitions that define the term in their own way. We cannot give one single or one best definition because more definitions are suitable.

Here are a few **definitions:**

"Knowledge management is the management of the activities and processes for leveraging knowledge to enhance competitiveness through better use and creation of individual and collective knowledge resources." (Chaffey, 2009)

"Knowledge management is a process that emphasises generating, capturing and sharing information know how and integrating these into business practices and decision making for greater organisational benefit.” *By Maggie Haines, NHS Acting Director of KM* (NHS, 2005)

“The capabilities by which communities within an organisation capture the knowledge that is critical to them, constantly improve it, and make it available in the most effective manner to those people who need it, so that they can exploit it creatively to add value as a normal part of their work.” *By BSI’s A Guide to Good Practice in KM* (ibidem)

“Knowledge management in general tries to organize and make available important know-how, wherever and whenever it’s needed. This includes processes, procedures, patents, reference works, formulas, “best practices”, forecasts and fixes. Technologically, intranets, groupware, data warehouses, networks, bulletin boards videoconferencing are key tools for storing and distributing this intelligence.” *By Maglitta (1996)* (Malhotra, 2000)

“Knowledge management incorporates intelligent searching, categorizing and accessing of data from disparate databases, E-mail and files.” *By Willet & Copeland (1998)* (ibidem)

To summarize, good knowledge management is all about getting the right knowledge, in the right place, at the right time. Knowledge management is based on the idea that an organisation’s most valuable resource is the knowledge of their people. It is essentially about facilitating the processes that creates, shares and uses knowledge in organisations. It is not about setting up a new department or providing a new computer system. It is about making small changes to the way everyone in the organisation works (NHS, 2005). Raw information may be widely available to a lot of organisations, but only some organisations will be able to convert the information into relevant knowledge and to use this knowledge to achieve their goals. The processes by which these organisations do this are known as knowledge management strategies (Hovland, 2003)

Knowledge in organisations is often **classified into two types**: explicit and tacit knowledge.

Explicit knowledge is knowledge that can be confined and written down in documents or databases. Some examples of explicit knowledge are instruction manuals, written procedures, best practices, lessons learned and research findings. Explicit knowledge can be categorised structured or unstructured. Documents, databases, and spreadsheets are examples of structured knowledge, because the data or information in them is organised in a particular manner for future retrieval. On the other hand, e-mails, images, training courses, and audio and video selections are examples of unstructured knowledge because the information they enclose is not referenced for future retrieval (NHS, 2005).

Tacit knowledge is the knowledge that is less tangible than explicit knowledge. People carry in their heads. It is the experience of how to react to a situation when many different variables are involved (Chaffey, 2009). Tacit knowledge is context-specific and contains, among other things, insights, intuitions and experiences (Research Matters, 2008). It is like an "unspoken understanding" about something. Knowledge that is difficult to write down in a document or a database and can be difficult to access. In fact, most people are not aware of the knowledge they possess or of the value of this knowledge to others. Tacit knowledge is considered more valuable than explicit knowledge because it provides context for people, places, ideas and experiences (NHS, 2005).

One approach is to think of knowledge management in terms of **three components**, namely people, processes and technology.

People: The most important and yet often the most difficult part of knowledge management is getting the culture of an organisation "right" (NHS, 2005). The culture of organisations has to be appropriate or will need to be adapted in order to create a proper knowledge management environment.

Processes: To improve knowledge sharing, organisations often need to make changes to the manner in which their internal processes are structured, and sometimes even the organisational structure itself. An organisation needs to know if and how these processes can be adapted or what new processes can be introduced to support people in creating, sharing and using knowledge (ibidem).

Technology: A widespread misconception is that knowledge management is mainly about technology, for example: getting an intranet, linking people by e-mail, compiling information databases etc. Technology is in most cases a crucial element of knowledge management because it connects people with information, and people with each other, but is not the solution (ibidem).

The people component is the most important element of these three components, but all three components will need to be reviewed by organisations. That is why it is important for organisations to develop a knowledge-friendly culture and knowledge-friendly behaviours among its people. This should be supported by the appropriate processes and enabled through technology (ibidem).

2.2 Information overload

Studies have shown that as a decision maker is given more information, decision quality initially increases. Once the information level reaches a certain point, however, the quality of the decision decreases if he is given additional information. The idea is that at some point (many studies suggest between five and ten attributes per choice) people become overloaded with information and begin to make worse decisions (Paredes, 2003). This is simply because people have limits in the amount of information they can process. That is why parts of the information will be ignored, forgotten, distorted or otherwise lost (Heylighen, 2005a).

Information and communication technology has made information abundant. Because of the Internet you can basically get any information you might desire in seconds. By using information and communication technology, information can be retrieved, produced and distributed much more easier than in earlier periods (ibidem). However, the reliability, usefulness, and timeliness of the information is difficult to verify (Laud & Schepers, 2009). There is an existence of excess low quality information that is called *data smog* (Heylighen, 2005a).

The impact of information overload on decision making has been well researched over the past ten years as the introduction and the growth of the Internet caused a explosion of information (Davenport & Beck 2000). Executives needed to deal with new and huge volumes of data, determine relevancy, and identify reliability of information sources (Laud & Schepers, 2009).

For example, quickly changing business valuations, price-earnings multiples, mergers and acquisitions activity, restructuring, "unforeseen" restatements, and even bankruptcy have allowed some privileged stakeholders and corporate leaders to accrue windfall gains, while many other investors have suffered dramatic losses. The business reporting that surrounds these events is performed by skilled professionals including corporate and tax lawyers, accountants, auditors, board members, trust advisors, investment bankers, finance specialists, and analysts (ibidem).

Information overload can lead to anxiety and loss of control. This can eventually lead to increasing stress, that can go together with physical, psychological and social problems (Heylighen, 2005a). A world-wide survey, executed by Waddington (1996) found that two thirds of managers suffer from increased tension and one third from ill health as the result of information overload (ibidem).

The previous paragraphs may suggest that the increase of information can only be seen as a bad thing, but this is not correct. The availability of huge amounts of information, if managed properly, can be seen as an opportunity.

A solution to tackle information overload is proposed by the paper of Heylighen (2005b). The paper indicates that the solution is the integration of the three basic resources, namely *human intelligence, computer intelligence* (computers are much less limited in the amount of information that can be processed than humans), and *coordination mechanisms* that direct an issue to the cognitive resource (document, person, or computer program) most fit to address it. This requires a distributed, self-organizing system, formed by all individuals, computers and the communication links that connect them. The resulting information system would be available always and everywhere and would be able to react immediately to any request for guidance. New information would be constantly added, from the human users and computer agents (ibidem). The recommender system that is the subject of this master thesis is an example of a software system that will help tackling the problem of information overload.

2.3 Conclusion research sub-question 2: *Why is a recommender system needed?*

Knowledge management has become a very important concept. Knowledge management is all about getting the right knowledge, in the right place, at the right time. It is based on the idea that an organisation’s most valuable resource is the knowledge of their people. Raw information may be widely available to a lot of organisations, but only some organisations will be able to convert the information into relevant knowledge and to use this knowledge to achieve their goals.

Information and communication technology has made information abundant. Because of the Internet you can basically get any information you might desire in seconds. However, the reliability, usefulness, and timeliness of the information is difficult to verify. There is an existence of excess low quality information that can lead to inefficient decision making.

To help organisations in their knowledge management process and to overcome the problem of information overload, recommender systems can be used. Recommender systems will lead to more efficient knowledge acquisition. They can lead to a reduction of time in this very time-consuming process and this can result in a decrease of costs for enterprises for which knowledge acquisition is important.

Chapter 3: Technological blocks of recommender systems

A recommender system is a software system whose main goal is to aid in the social collaborative process of indicating or receiving indication, when the number of options is enormous (Resnick & Varian, 1997). Recommender systems are proactive devices and their goal is to supply people with information useful for decision making or to solve problems. This information may be about books, documents, music, restaurants and whatever (ibidem). Recommender systems will be explained in more detail in chapter 4 (p.30).
When creating a recommender system that complies to the needs of users nowadays, it is necessary to involve and understand the following topics.

3.1 Ontology

An ontology is a concept that has several meanings and can be explained in a number of ways. First of all, an ontology is a representation of a vocabulary, often specialized to some domain or subject matter. "More precisely, it is not the vocabulary as such that qualifies as an ontology, but the conceptualizations that the terms in the vocabulary are intended to capture" (Chandrasekaran, et al., 1999). According to Uschold & Gruninger (1996), "Ontology is a term used to refer to the shared understanding of some domain of interest which may be used as a unifying framework." They say that an ontology entails or embodies some sort of world view with respect to a given domain. This world view is often conceived as a set of concepts (e.g. entities, attributes, processes), their definitions and their inter-relationships, which is referred to as a conceptualization. It is important to point out that an ontology is language-dependent, while a conceptualisation is language independent (Guarino, 1998). So, two ontologies can use different vocabulary, for example ontologies in different languages, while sharing the same conceptualization (ibidem).

Blanco-Fernández, et al. (2008) say "an ontology characterizes that semantics in terms of concepts and their relationships, represented by classes and properties, respectively. Both entities are hierarchically organized in the conceptualization, which is populated by including specific instances of both classes and properties. For example, in the context of a recommender system, instances of classes represent the available items and their attributes, whereas instances of properties link the items and attributes to each other."

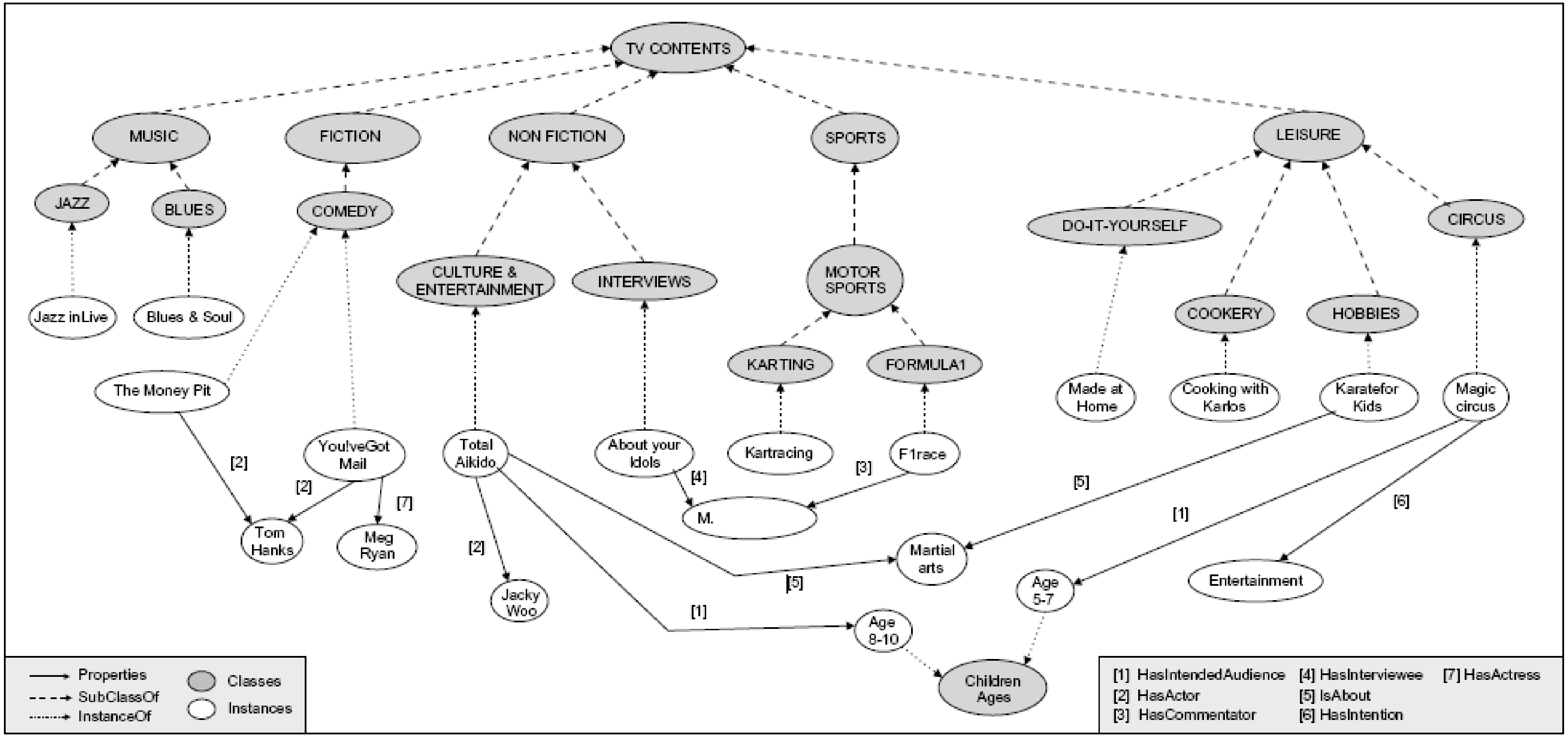
An example of a graphical illustration for an ontology of a TV domain is given in figure 3.1 (p.13).
"In this figure, we have several instances referred to specific TV programs that belong to a hierarchy of genres, whose root node is TV CONTENTS (e.g., FICTION, SPORTS, MUSIC, LEISURE). The attributes of these programs (e.g., cast, intended audience, topics) are also identified by hierarchically organized classes, and are related to each program by means of labelled properties (e.g., hasActor, hasIntendedAudience, isAbout)" (Blanco-Fernández, et al., 2008).

Ontological analysis clarifies the structure of knowledge and enables knowledge sharing. The knowledge representation language that an ontology uses can be shared with others who have similar needs for knowledge representation. Hereby, there is no need to replicate the knowledge that already has been created (Chandrasekaran, et al., 1999).

Staab & Studer (2004) say that it is not easy to agree on a common definition, because ontology is being used in such a diverse application context. They state that, in the informatics community, there has been some agreement on using the following definition: "An ontology is a formal explicit specification of a shared conceptualization for a domain of interest." The ontology has to be specified in a language that comes with a formal semantics. This formal approach ontologies will provide a machine interpretable meaning of concepts and relations that is expected when using an ontology-based approach (ibidem).

The type of ontology that will be used in the rest of this paper, is the use of an ontology as a database component. In this component, an ontology can be compared with the schema component of a database. At the development time, an ontology can play an important role in the requirement
analysis and conceptual modelling phase, "The resulting conceptual model can be represented as a computer processable ontology and from there mapped to concrete target platforms" (Guarino, 1998). Guarino also points out that an ontology can support queries regarding the content of a particular database.

Figure 3.1: A graphical illustration of an ontology



Source: Blanco-Fernández, et al. (2008)

The WWW Consortium (W3C) developed the Resource Description Framework (RDF) (see infra, section 3.2, p. 14), a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information (Noy & McGuinness, 2001). Other technologies, like the Web Ontology Language (OWL) (see infra, section 3.3., p.16), build on RDF and provide language for defining structured, Web-based ontologies which enable richer integration and interoperability of data among descriptive communities (http://www.w3.org/TR/owl-guide/).

Sharing common understanding of the structure of information among people or software agents is one of the common goals of developing ontologies. For example, when different websites of the same domain contain information about this domain, they can share and publish the same underlying ontology of the terms they use. If they al use the same ontology, computer agents can extract and aggregate information to answer the users' queries or as input data to other applications (Noy & McGuiness, 2001)

"Ontologies have become the cornerstone in the Semantic Web due to two reasons" (Blanco-Fernández, et al., 2008). On the one hand, as these conceptualizations represent formally a specific domain, ontologies enable inference processes to discover new knowledge from the formalized information. On the other hand, ontologies make it easier to automate knowledge sharing, by allowing easy reuse between users and software agents (ibidem).

However there are some **challenges** regarding ontology development and management. The ontologies that will supply the Semantic Web must be developed, managed and supported by practice communities (Shadbolt, et al. 2006). These communities need to provide the definitions used in the ontologies. Although some denotations are more persistent than others, others will not be fixed over all time. Some ontologies might endure long periods (for example terms describing the elements of the periodic table), while others are more volatile (for example terms describing parts of the Internet).
When different members of a community create their own ontology, containing autonomous entities, there will also be the problem of ontological inconsistency. Though full ontological consistency is not reachable in these large communities, there has to be some degree of consistency between ontologies (Zurawski, et al., 2008).
Communities and practice will change norms, conceptualisations and terminologies over time. The issue for the Semantic Web, that is built from these parts, is to know when

these parts needs revision. The problem here is the **cost of ontology development and maintenance**, an often quoted concern about the Semantic Web (ibidem).

3.2 Resource Description Framework (RDF)

3.2.1 What is RDF?

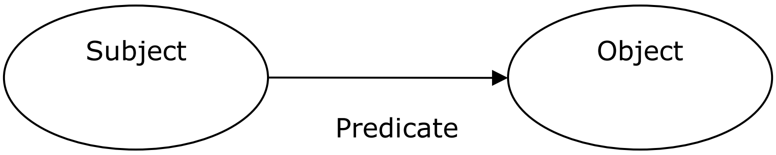
The Resource Description Framework (RDF) is a W3C standard format for storing arbitrary data on the Web and elsewhere (Ducharme, 2005). It is a standard model for data interchange on the Web (http://w3c.org/RDF/). In other words, it is a framework for representing information in the Web. RDF provides a machine-readable way to say anything about anything (Ducharme, 2005).

The WWW Consortium (W3C) developed the Resource Description Framework as a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information (Noy & McGuinness, 2001).

RDF allows the description of resources by expressing statements about them in the form of triples: subject, predicate and object. Resources are identified by a Unique Resource Identifier (URI). A Uniform Resource Identifier (URI) is a string of characters used to identify a name or a resource on the Internet. While Uniform Resource Locators (URL) have become familiar as addresses for documents and other entities that can be located on the Web, URIs provide a more generic way to identify any entity that exists in the world (Bizer, et al. 2009).

The underlying structure of any expression in RDF is a collection of triples. The structure is clarified in figure 3.2.

Figure 3.2: Representation of the parts of a triple



Source: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

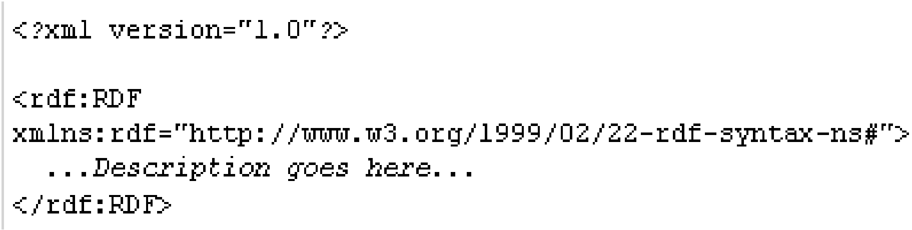
Each triple represents a statement of a relationship between the things denoted by the nodes that it links. Each triple has three parts:

1. a subject,
2. an object, and
3. a predicate (also called a property) that specifies a relationship.

3.2.2 Syntax of RDF

Now a part of the RDF syntax will be explained. The parts in this explanation, shown in figure 3.3, will be used in the case study later on in this master thesis.

Figure 3.3: Standard syntax of an RDF file



```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  ...Description goes here...
</rdf:RDF>
```

The first part of a typical RDF file contains the <?xml version="1.0"?> syntax. This syntax will specify that xml encoding will be used. The next part is the “rdf:RDF” element of the syntax. It defines the xml document to be an RDF document. The namespaces are given in the following line “xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">”

A namespace is an abstract container or environment created to hold a logical grouping of unique identifiers or symbols (i.e., names). An identifier defined in a namespace is associated with that namespace. XML namespaces provide a simple method for qualifying element and attribute names used in Extensible Markup Language documents by associating them with namespaces identified by URI references (http://www.w3.org/TR/REC-xml-names/). Thus, a namespace will give a URI for an element or attribute name in the beginning of the RDF file, so that it will not be necessary to give the whole URI again in the rest of the file, when the element or attribute is included.

For example, the URI "http://purl.org/**dc**/elements/1.1/**title**" can be shortened by including the namespace URI *"xmlns:**dc**="http:// purl.org/dc/elements/1.1/"* in the rdf:RDF start tag. Than it can be re-used as <**dc:title**> later on.

The namespace that is given in figure 3.3 is a namespace that needs to be included in every RDF file because it will provide a description of the syntax of the "RDF element". Next to this namespace, typical RDF files will contain more namespaces.

After these steps, the description of the RDF file will be given, providing the necessary information that was the reason for creating the file.

3.2.3 Problems with creating RDF

Halb, et al. (2008) point out a few bottlenecks when creating datasets of enormous size in RDF. First of all, it is a very time consuming process to generate a static file-structure with small RDF files. Another challenge can be the ambiguity of the raw data used for creating RDF. This only can be resolved by analysing the corresponding document. An example of this is given by Halb, et al. (2008) for raw data of Eurostat. They give the example of the time indicator "2007". This can stand for the value over a period of time (for example the entire year) or at the end of a reporting period (for example, 31 December) (ibidem).

3.3 OWL

Another term that needs to be explained about this subject is the Web Ontology Language (OWL). This an example of an ontology language based on RDF. "OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics" (http://www.w3.org/TR/owl-features/).

3.4 SPARQL

3.4.1 What is SPARQL?

SPARQL, an acronym standing for SPARQL Protocol and RDF Query Language, is a general term for both a protocol and a query language. Most uses of the SPARQL acronym refer to the RDF query language (Prud'hommeaux & Seaborne, 2008).

"SPARQL can be utilized to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware" (ibidem).

A typical SPARQL query consist of three parts. The first part will identify the **prefixes** used in the query, so that there is no need the type it in full every time it is referenced in the query. The next part is the **SELECT** clause that will identify the variables that will appear in the query results. The last part is the **WHERE** clause. This clause provides the basic graph pattern to match against the data graph (Prud'hommeaux & Seaborne, 2008). It will define what will be queried for.

SPARQL queries need to be executed at a SPARQL endpoint. A SPARQL endpoint makes it possible for users to query a knowledge base by using the SPARQL language. Results are typically returned in one or more machine-processable formats. Therefore, a SPARQL endpoint is mostly conceived as a machine-friendly interface towards a knowledge base. (http://semanticweb.org/wiki/SPARQL_endpoint).

3.4.2 Example of a SPARQL query

This example will clarify what is explained in the section above. It is an example that will query the DBpedia database and will look for all landlocked countries with a population greater than 15 million.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
  ?country a type:LandlockedCountries ;
           rdfs:label ?country_name ;
           prop:populationEstimate ?population .
  FILTER (?population > 15000000) .
}
```

First of al, the PREFIXES "rdfs:", "type:" and "prop:" are specified. The SELECT clause tells us that the variables "country_name" and "population" will be selected to present in the results. In the WHERE clause, the landlocked countries are separated from all the countries and a filter is provided to make sure that only countries with a population over 15 million are selected in the results.

3.5 The Semantic Web

The Semantic Web is not a separate Web, but an extension of the World Wide Web. The Semantic Web gives information a well-defined meaning that enables computers and people to work better in cooperation (Berners-Lee, et al. 2001). The Semantic Web allows people to share content beyond the boundaries of applications and websites (http://semanticweb.org/wiki/Main_Page). It will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can carry out sophisticated tasks for users (Berners-Lee, et al., 2001). Like the Internet, the Semantic Web is as decentralized as possible (ibidem).

"While a Web browser navigates along links between documents, a Semantic Web browser navigates along relationships (or predicates as explained in section 3.2, p.14) in a web of concepts" (Berners-Lee, et al., 2006). However, a Semantic Web browser must also include an awareness of the underlying web of documents and query services (ibidem).

In order for the Semantic Web to function, computers must have access to structured collections of information and sets of rules that they can use to conduct automatic reasoning. The two most important technologies for developing the Semantic Web are XML and RDF. eXtensible Markup Language (XML) allows everyone to add arbitrary structure to their documents but says nothing about what the structure means. The meaning is expressed in RDF (see infra, section 3.2, p.14) (Berners-Lee, et al., 2001).

A third basic component of the Semantic Web are ontologies (see infra, section 3.1, p.11). In context of the Semantic Web, they are used to provide a way to discover common meanings for whatever database it uses. Common meanings are encountered when a program wants to compare or combine information of two or more databases when two different terms are being used for the same thing (Berners-Lee, et al., 2001).

Oren, et al. (2008) say that the Semantic Web can be seen as a large knowledge-base formed by sources that serve information as RDF files or through SPARQL endpoints. A fundamental feature of the Semantic Web is that the graphs are decentralised, meaning that it has no single knowledge-base of statements but instead anyone can contribute statements by making them available. The sources of the knowledge bases (for example web pages) might have nothing in common, but by using URIs and shared terms, their information can be merged to offer useful services to both humans and software clients (ibidem).

Five years after the article *The Semantic Web* was published by Berners-Lee, et al. (2001), the progress of the Semantic Web is reviewed by Shadbolt, et al. (2006). Because there hasn’t yet been delivered a large-scale, agent-based mediation, some commentators argue that the Semantic Web has failed. The authors of this article argue with this and say that agents can only flourish when standards are well established and the Web standard necessary for the Semantic web have progressed steadily in the period after 2001 (ibidem).

After 2001, the need for shared semantics and a web of data and information derived from it, has increased. The importance of ontologies has also risen and they are utilized more and more.

The next step according to these authors will be making substantial reuse of existing ontologies and data. They see the Semantic Web as a linked information space in which data is being enriched and added. In 2006, they could already see an increasing need and a rising obligation for people and organizations to make their data available (ibidem).

The Semantic Web technologies allow to create a *web of data.* Such a web of data embeds interlinked ontologies, giving formal specifications of concepts and relationships relevant for describing a set of objects in a given domain. So, the data is linked to its meaning (Raimond, et al., 2007).

3.6 Linked Data

3.6.1 What is Linked Data?

A fully functional Semantic Web is based on the availability of large amounts of data as RDF. This RDF data has to be interlinked as a web of data and should not be in isolated datasets (Bizer, et al., 2007).
The term Linked Data refers to a set of best practices for publishing and connecting structured data on the Web. Key technologies that support Linked Data are URIs, HTTP and RDF (http://linkeddata.org/).

Linked Data is about using the Web to connect related data that was not yet linked before, or using the Web to lower the barriers to linking data that is currently linked by using other methods. Wikipedia defines Linked Data as "a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF" (ibidem).

According to Hausenblas (2009) and Bizer, et al. (2009), the fundamental idea of Linked Data has first been outlined by Sir Tim Berners-Lee in 2006. He described the four principles of Linked Data as:

1. All items should be identified using URIs;
2. Use HTTP URIs so that people can look up these names (dereferenceable);
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL);
4. Links to other URIs should be included in order to enable the discovery of
more data.

Linked Data is mainly about publishing structured data in RDF using URIs rather than focusing on the ontological level. This simplification lowers barriers to enter for data providers and by doing so, fosters a wide spread adoption (Hausenblas, 2009). The Linked Data initiative has given rise to an increasing number of RDF datasets of which many are accessible on line for free. The resources often arise as a result of database exports (Jaffri, et al., 2008)

Publishing Linked Data on the web needs to be done according to the Linked Data principles that involve the following three steps. The first step is to assign URIs to the entities described by the data set and provide for dereferencing these URIs over the HTTP protocol into RDF representation. The second step is to set RDF links to other data sources on the Web, so clients can navigate the Web of Data as a whole by following RDF links. The final step is to provide metadata about published data, so that clients can asses the quality of published data and choose between different means of access (Bizer, et al., 2009).

Opinions of the **relationship** between the Semantic Web and Linked Data differ somewhat. However a widely held view is that the Semantic Web is made up of Linked Data. This means that the Semantic Web is the whole, while Linked Data are the parts. Tim Berners-Lee has frequently described Linked Data as "the Semantic Web done right" (http://linkeddata.org/).

The goal of the *open data movement* is to make data freely available to everyone. The **Linking Open Data (LOD) project** aims at making these datasets available in RDF and create RDF links between them (Raimond, et al., 2007). The linking open data project is an open, collaborative effort carried out in the area of the W3C SWEO Community Projects initiative (Hausenblas, 2009). It aims at bootstrapping the Web of Data by publishing datasets in RDF on the Web and creating large numbers of links between these datasets (ibidem). The project began in 2007 with a relatively modest number of datasets and participants and has grown ever since in terms of depth, impact and contributors. "Currently, the project includes over 50 different datasets with over two billion RDF triples and three million (semantic) links at the time of writing, representing a steadily growing, open implementation of the linked data principles" (ibidem). The content of the cloud is very diverse, comprising data about geographic locations, people, companies, books, scientific publications, films, music, television and radio programmes, genes, proteins, drugs and clinical trials, online communities, statistical data, census results and reviews (Bizer, et al., 2009) Figure 3.4 (on page 24) shows the LOD cloud that illustrates these different datasets and shows their relations with each other. The arcs in figure 3.4 indicate the links that exist between items in the two connected datasets. Heavier arcs correspond to a greater number of links between the two datasets, while bidirectional arcs point out the outward links to the other exist in each dataset (Bizer, et al.,2009). However these interlinked datasets through reuse of common vocabulary and shared URIs are starting to expand, they are not yet widespread (Oren, et al., 2008). That is why Tim Berners-Lee, a very important man in this domain, summons everyone to publish *"Raw Data Now"* in his TED talk (2009). People, companies, governments, etc. should put their data on line as Linked Data to uncover a huge unlocked potential. An example of this can be given for the revenue of a company. At the moment, the revenue for a given company can, for example, be found in the DBpedia database. This is because a person who is a member of the Wikipedia community has uploaded this data in the database. The goal in the future will be that the company itself publishes its data (revenue) and link it to other data (for example DBpedia). This will make data more reliable and accurate.

Besides this problem, there is another difficulty concerning Linked Data, namely URI disambiguation. Linking data has been widely encouraged, but there has been little analysis of the accuracy of the links or the datasets themselves (Jaffri, et al., 2008). This is because datasets are often converted from existing sources that can themselves be incomplete or inaccurate. This can also be due to incorrect information that is publicised by members of the community. When linking these inconsistencies, a snowball effect of incomplete or inaccurate data is produced as more datasets are added. The main area in which problems arise is in *co-referencing* (ibidem). Co-referencing can occur in two ways. Firstly, when a single URI indentifies more than one resource. Secondly, when multiple URIs identify the same resource. Both situations occur frequently when studying Linked Data. An example of the first situation can be when a URI identifies a single author when, in fact, there are a number of people with the same name, that are wrongly identified as the same person. The second situation of URI multiplicity occurs much more frequently. An example of this situation can be given for the country Spain. This country would be identified by different URIs in different datasets. For example, the DBpedia and GeoNames will have different URIs to define Spain.

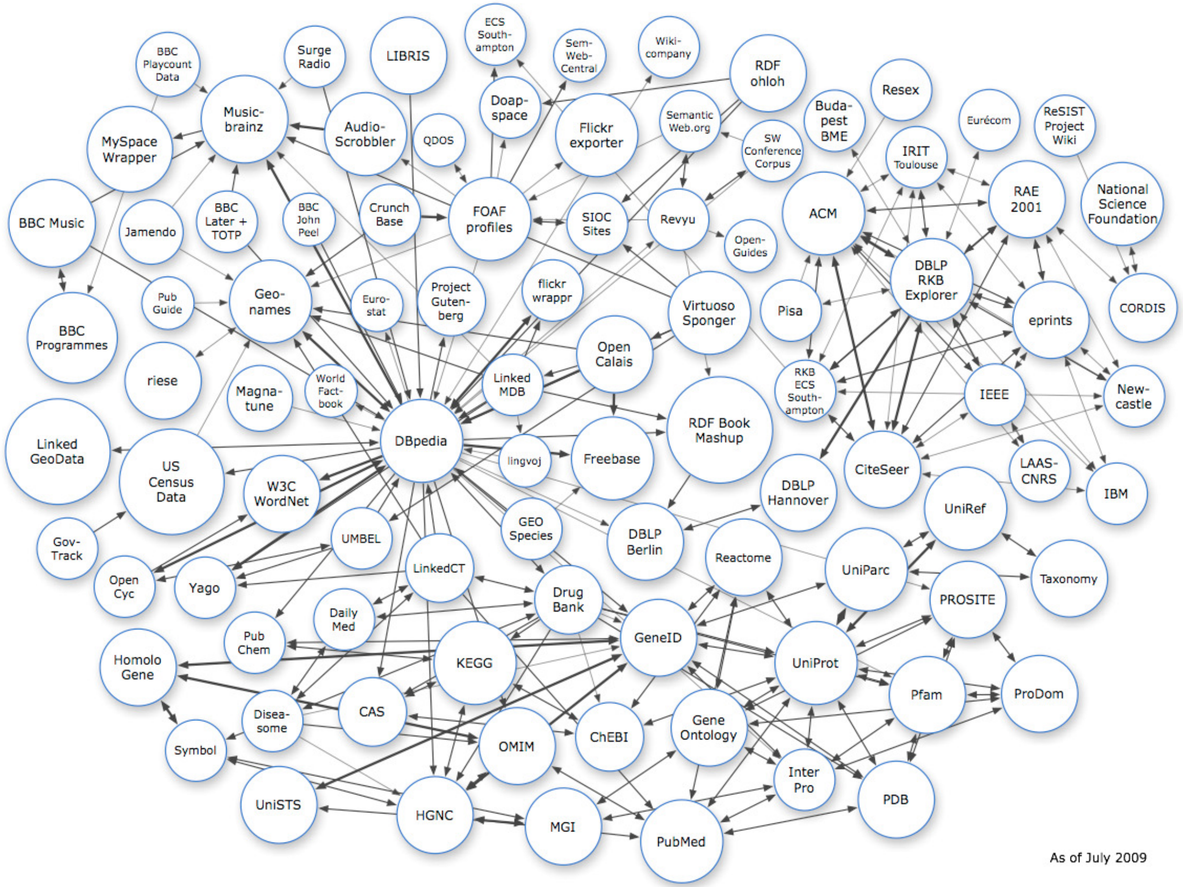
The problem occurs when these URIs are linked to other URIs via *owl:sameAs.* Because the URI identity can often depend on the context in which it is used, there can not be any guarantee that the two URIs are in fact the same. For a human, this can be simple to work out, but machines will struggle with this problem (ibidem).

The problem described in this paragraph is found in many digital repositories, due to the lack of resources that leads to insufficient time to rigorously check the input for correctness or completeness. This lack of resources can be explained by the free availability and community based approach of the Linked Data initiative. There need to be made investments in measuring and guaranteeing of the correctness of this information.

Shabir & Clarke (2009) say that the Linking Open Data movement has conducted excellent work, but they also have a fourfold critique on it.

Firstly, there is a concern about the *sustainability* of Linked Data. This is because many Linked Open Data endpoints are maintained by hobbyists or by projects or programmes with time-limited funding. Developers of applications that either link to this Linked Data or extract data from Linked Data sources, need to be confident that the data sources will continue to be available, even if the original host disappears. For the moment, there is not yet enough assurance that this will be the case. The second critique that the authors give is about the *provenance*. A user needs to be able to trace a piece of data that is represented on the Web of Linked Data. When data is aggregated together in RDF, it is not easy to identify how each data set contributed to the aggregated view. A third point of critique is *licensing*, that will to explain to users what they can and can not do with the original work. The problem for Linked Data is that large proportion of the data in the Linked Data cloud is not specifically licensed at all. The last concern about Linked Data is the *reliability* of the data. On the document Web, sites frequently disappear, leading users to experience, expect and work around 404 errors. Developers of Linked Data applications need to ensure that their systems will still be able to operate if Linked Data sources are temporarily or permanently unavailable (ibidem).

Figure 3.4: LOD cloud as of July 2009



Source: http://linkeddata.org/

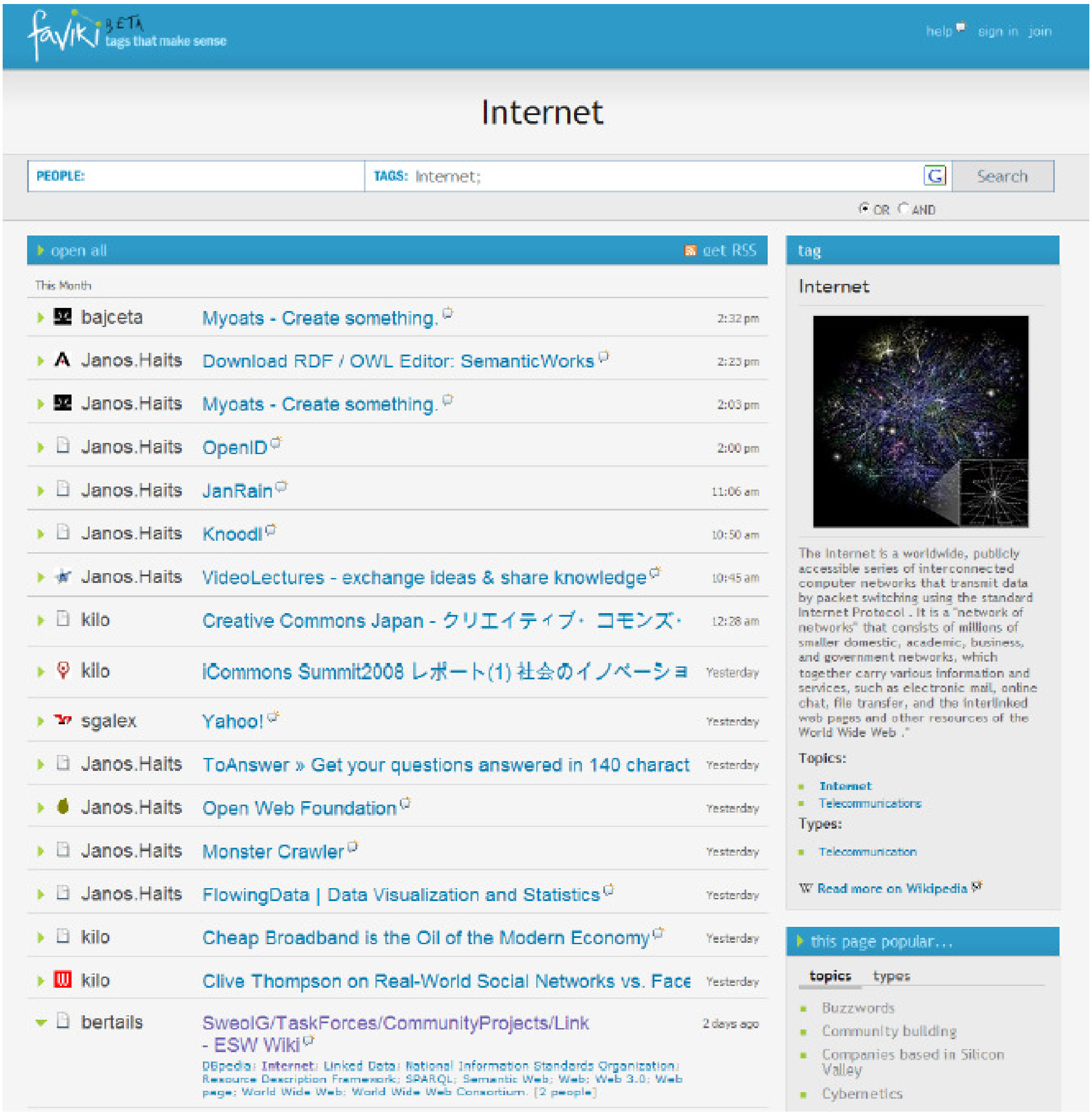
3.6.2 Examples of Linked Data Applications

An example of Linked Data is given by Raimond, et al. (2007). For example, *a festival happening in Montreal on 28 June 2007,* from a music database can be linked. The **Festival** can be linked to its geographic location from a geographical database (for example GeoNames). A user agent crawling the web of data can then jump from their knowledge base to the GeoNames one (ibidem). These links can also be made to other databases for bands performing at the festival, the companies that sponsor the festival, etc.

Other examples in which linked datasets are used, are provided in a paper from Hausenblas (2009). *Faviki* is a social bookmarking tool that allows to tag Web-pages with "semantic tags" originated from Wikipedia. The purpose of the Web of Data in this example is to provide unambiguous space for identifying concepts. The tool uses URIs from DBpedia for tagging. An example is given in figure 3.5. This example uses the DBpedia resource of the "Internet" as a tag. So anyone who is interested in this term can dereference this URI and is able to obtain further information about it.

Another example is *DBpedia mobile*, which is an application for mobile environments. It is a location-centric DBpedia application for mobile devices based on a GPS signal of a mobile, that will create a map indicating nearby locations from the DBpedia dataset, reviews from Revyu and photos from Flickr photo sharing API. DBpedia mobile also enables users to publish their current location, pictures and reviews to the Web as Linked Data, so they can be used in other applications (Bizer, et al., 2009) A screenshot of DBpedia mobile is shown in figure 3.6.

Figure 3.5: Screenshot of Faviki with the example of the “Internet” tag



Internet

PEOPLE:

TAGS: Internet;

● OR ○ AND

This Month

bajceta	Myoats - Create something.	2:32 pm
Janos.Haits	Download RDF / OWL Editor: SemanticWorks	2:23 pm
Janos.Haits	Myoats - Create something.	2:03 pm
Janos.Haits	OpenID	2:00 pm
Janos.Haits	JanRain	11:06 am
Janos.Haits	Knoodl	10:50 am
Janos.Haits	VideoLectures - exchange ideas & share knowledge	10:45 am
kilo	Creative Commons Japan - クリエイティブ・コモンズ・	12:28 am
kilo	iCommons Summit2008 レポート(1) 社会のイノベーショ	Yesterday
sgalex	Yahoo!	Yesterday
Janos.Haits	ToAnswer » Get your questions answered in 140 charact	Yesterday
Janos.Haits	Open Web Foundation	Yesterday
Janos.Haits	Monster Crawler	Yesterday
Janos.Haits	FlowingData | Data Visualization and Statistics	Yesterday
kilo	Cheap Broadband is the Oil of the Modern Economy	Yesterday
kilo	Clive Thompson on Real-World Social Networks vs. Face	Yesterday
bertails	SweoIG/TaskForces/CommunityProjects/Link - ESW Wiki DBpedia; Internet; Linked Data; National Information Standards Organization; Resource Description Framework; SPARQL; Semantic Web; Web; Web 3.0; Web page; World Wide Web; World Wide Web Consortium. [2 people]	2 days ago

tag

Internet

The Internet is a worldwide, publicly accessible series of interconnected computer networks that transmit data by packet switching using the standard Internet Protocol . It is a "network of networks" that consists of millions of smaller domestic, academic, business, and government networks, which together carry various information and services, such as electronic mail, online chat, file transfer, and the interlinked web pages and other resources of the World Wide Web ."

Topics:

- Internet
- Telecommunications

Types:

- Telecommunication

Read more on Wikipedia

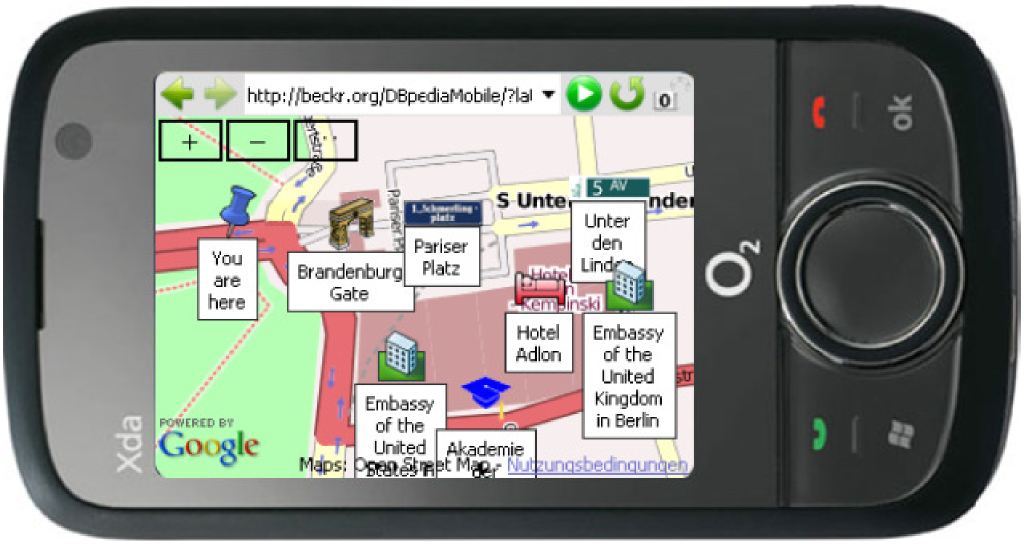
this page popular...

topics types

- Buzzwords
- Community building
- Companies based in Silicon Valley
- Cybernetics

Source: Hausenblas (2009)

Figure 3.6: Screenshot of DBpedia mobile



Source: Hausenblas (2009)

3.6.3 Challenges for Linked Data

There still remain some research challenges that must be overcome before the ultimate goal of using the Web like a single global database could be reached.

A first challenge is to create *user interfaces* that are able to integrate data from sources that are not explicitly selected by the user (Bizer, et al., 2009).
There are also *scalability problems* when widespread crawling and caching for huge amounts of Linked Data. Because the amount of data can be huge, it can be difficult to provide the results in a timely fashion.
Another problem can be found in the domain of *data fusion*. Data fusion is the process of integrating multiple data items representing the same real-world object into a single, consistent and clean representation. The most important challenge in data fusion is choosing a value in situations where multiple sources provide different values for the same property of an object (ibidem).

Because the content of Linked Data sources changes, when data about new entities is added or outdated data is changed or removed, the RDF links between data sources need to be updated. Today, *link maintenance* of these RDF links only occurs sporadically, which leads to dead links pointing at URIs that are no longer maintained and to potential links not being set as new data is published. Another challenges is addressing the *licensing issue*. It needs to be guaranteed that when using data from a publisher in a certain way, it does not infringe the rights of others.

One of the most significant challenges for Linked Data is how to ensure that the data that is most relevant or appropriate to the users' needs is identified and made available. This means *providing relevant trustworthy data of good quality*.

A last important challenge for Linked Data is *privacy*. The ultimate goal of Linked Data is to be able to use the Web like a single global database. When integrating data from distinct sources, it can't be allowed to violate the privacy of these sources (ibidem).

3.7 Conclusion research sub-question 3: *What are the technological building blocks of recommender systems?*

To formulate an answer to the second research sub-question, this chapter described the various building block that developers of a modern recommender systems need to consider. A first concept that was explained in the context of recommender systems are ontologies. An **ontology** is a representation of a vocabulary, often specialized to some domain or subject matter. An ontology characterizes the semantics in terms of concepts and their relationships, represented by classes and properties, respectively. In the context of a recommender system, instances of classes represent the available items and their attributes, whereas instances of properties link the items and attributes to each other.

Secondly, **RDF** is another concept explained in this chapter. The resource description framework (RDF) is a W3C standard format for storing arbitrary data on the Web and elsewhere. In other words, it is a framework for representing information in the Web. RDF provides a machine-readable way to say anything about anything.

SPARQL, an acronym standing for SPARQL Protocol and RDF Query Language, is a general term for both a protocol and a query language. SPARQL can be utilized to express queries across RDF data.

Another important building block which a recommender system can make use of is the **Semantic Web**. The Semantic Web is not a separate Web, but an extension of the World

Wide Web. The Semantic Web gives information a well-defined meaning that enables computers and people to work better in cooperation. It will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can carry out sophisticated tasks for users.

A fully functional Semantic Web is based on the availability of large amounts of data as RDF that is interlinked as a web of data. **Linked Data** refers to a set of best practices for publishing and connecting structured data on the Web in order to achieve this interlinking of data. There are projects, like the Linking Open Data project, that aim at making huge amounts of data freely available by publishing datasets in RDF and create RDF links between them.

Chapter 4: Recommender systems

This chapter will explain what recommender systems are, give examples of existing recommender systems, and give a case example of a recommender system that delivers relevant information during chat discussions. This chapter will give an answer to research sub-question 1: *What is a recommender system and what are its functions and?*

4.1 What is a recommender system?

A recommender system is a software system whose main goal is to aid in the social collaborative process of indicating or receiving indication, when the number of options is enormous (Resnick & Varian, 1997). "Recommender systems are proactive devices and their goal is to supply people with information useful for decision making. This information may be about books, documents, music, restaurants and whatever" (ibidem). The most important benefit of a recommender system is that it can supply information without people have to search for it. "Recommender systems are becoming an important alternative to support knowledge acquisition" (X,2010). Recommender systems will lead **more efficient knowledge acquisition**. The **reduction of time** in this time-consuming process can lead to a **decrease in costs** for enterprises for which knowledge acquisition is important, especially with the increased information overload nowadays.

As explained in section 2.2, information and communication technology has made information abundant. It has become a complex activity to gather information on the Web (Porcel, et al., 2008). Especially because information gathering on the Internet has become very important, a lot of recommender systems these days aim at helping to support end users with their search for information on the web.

According to Hanani, et al. (2001) and Resnick & Varian (1997), the recommender systems can be characterized because they:

- "are applicable for unstructured or semi-structured data (e.g. Web documents or e-mail messages),
- the users have long time information needs that are described by means of user profiles,
- handle large amounts of data,

- deal primarily with textual data and
- their objective is to remove irrelevant data from incoming streams of data items."

The most important authors agree that there are two main categories of recommender systems, namely *content-based recommender systems* and *collaborative recommender systems*.

Content-based recommender systems or *content-based filtering* suggests items to a user which are similar to those he/she liked in the past, by matching their respective content descriptions (i.e., the features defined in the user's profile and the attributes of the available items) (Blanco-Fernández, et al., 2008). The most important strength of this contend-based filtering is that recommendations can be done without knowing the preferences of others. However, this method has some drawbacks. For example, the content description for the available items is a time-consuming task and usually requires an expert to describe accurately the application domain (ibidem). Another downside is that this method will only recommend items that are excessively similar to those the user already knows (Adomavicius, G. & Tuzhilin, A., 2005).
The last problem of traditional content-based approaches is called "*new user ramp-up*". This problem occurs when a new user arrives in the system, the recommender usually knows little information about his/her preferences, and so the offered suggestions are poor and imprecise (Montaner, et al., 2003).

Collaborative recommender systems or *collaborative filtering* recommends to a user items which have been appealing to others with similar preferences, called neighbours. Firstly, the user's neighbourhood is formed. Next, his/her levels of interest in the items defined in the neighbours' profiles are predicted, and those with the highest ratings are finally recommended to the user (Blanco-Fernández, et al., 2008). "Since collaborative systems do not only consider the user's preferences – but also his/her neighbours' interests – they offer diverse recommendations, beyond the overspecialized suggestions of content-based approaches" (ibidem). This method also has some weaknesses according to Blanco-Fernández, et al. (2008).

- *Sparsity problem*: The effects of this limitation are apparent as the number of available items increases. In this case, it is unlikely that two users have rated the same items in their profiles, thus hampering the selection of the user's neighbours.

- *Scalability*: As the number of available items gets higher, the users' rating vectors also increase in size. Consequently, the creation of neighbourhoods (based on computing correlations between users' vectors) becomes very demanding in computational terms.
- *Latency problem*: Since collaborative approaches only suggest to the user items defined in their neighbours' profiles, new items available in the system cannot be used in a recommendation before a significant number of users have rated them in their profiles.

Because of this limitations, researchers have commonly opted for *hybrid systems,* where content-based and collaborative filtering are combined with the goal to make use of the advantages and lessen the weaknesses of both systems. "Hybrid approaches unify collaborative and content-based filtering under a single framework, by operating on both the user's ratings and the attributes of items" (Blanco-Fernández, et al., 2008).

There are also **Semantic-based approaches.** Instead of using techniques like in the previous systems, this system will reason about the semantics of the compared items. For that purpose, it takes advantage of the inference mechanisms involving semantic descriptions developed in the Semantic Web. This means it will discover hidden semantic associations by exploring the knowledge and structure of the ontological model. "Such an enhanced reasoning process permits the recommender system to learn additional knowledge about the users' preferences, thus improving the accuracy of the final suggestions" (ibidem). With a semantically structured representation of Web data, recommender systems can use semantic-based similarity measures in order to improve their effectiveness (Drumond & Girardi, 2008).

The next generation recommender systems will rely more on implicit information, such as the items that a user clicks on while navigating a site (Monroe, 2009). Recommendations will be made, based on, for example, your navigation patterns or your correlating products (ibidem).

An important remark that needs to be made with these recommender systems is that the **privacy** of the user has to be protected. "Privacy is an important issue in recommender

applications. In order to provide personal recommendations, recommender systems must know something about the customers" (Schafer, et al. 2001). Users are reasonably concerned about what information is collected, whether it is stored, and how it is used. Users that share data about themselves need to be assured that this data will be carefully protected. For instance, a user may give false information when he is not protected properly (ibidem).
When agents of recommender systems are crawling, they collect and index data from websites. This involves ethical issues with regard to the behaviour of the crawler. Crawling should be done in accordance with accepted ethics of good behaviour. The privacy of data publishers should be protected (Oren, et al., 2008)

4.2 Examples of recommender systems

In the following section, a series of existing recommender systems are described. There are numerous *forms* of recommender systems existing in many *domains*. It is not the intention to give a complete overview of the existing recommender systems, but to give some examples to clarify these systems.

Well known recommender systems are found in the domain of **E-commerce**. They are created to help customers find products to purchase and to boost revenue from on line sales. In a way, recommender systems enable the creation of a new stores that are personally designed for each consumer (Schafer, et al., 2001). Recommender systems can boost E-commerce sales in three ways. The first one is by converting visitors of a website into buyers by helping them find the products they want to buy. A manner in which sales can be enhanced by recommender systems is increased cross-selling. Increased cross-selling can be accomplished by suggesting additional products to the customer to purchase. A last way in which a recommender system can increase sales is by building loyalty. Recommender systems increase the customers loyalty by creating a value-added relationship between the site and the customer (ibidem). "Sites invest in learning about their customers, use recommender systems to operationalize that learning, and present custom interfaces that match consumer needs. Consumers repay these sites by returning to the ones that best match their needs" (ibidem).

A well-known E-commerce recommender system is the one used by **Amazon.com** in their book section. This recommender system contains several items that help a customer find products to purchase. "Customers Who bought" is a first part of the system. This recommends books frequently purchased by customers who purchased a selected book. "Your Recommendation" uses customer feedback to select other books they might like. Other features of the recommender system of Amazon.com can provide the customer with new information, about for example an author who has published a new book. "Customer Comments" allows customers to receive text recommendations based on the opinions of other customers. A last part of the recommender system used by Amazon.com are the "Purchase circles". This feature provides the customer with a "top 10" list for a given geographic region, company, educational institution, government or other organisation (Schafer, et al., 2001).

The iTunes Music Store uses very similar features to recommend music songs to customers. However, they use another feature called Genius. Genius scans the library of songs owned by the customer (this library might also contain songs not purchased at the iTunes Music Store). After scanning the tags of the songs in the library, it makes a list of suggestions of songs that the customer might like.

The example of Amazon.com makes recommendations that comply the customers' preferences. However, this advice might not be useful when a user returns to the site and is searching for a different type of product (Zhang, et al., 2007). Zhang, et al. (2007) present a system of website personalisation by using **data mining** in order to make the navigation of a site easier, enabling products that could meet the customer's requirements to be easily located. "This personal support is based on their current (rather than previous) navigation behaviour, which is discovered in real-time during their current visit" (ibidem). However, Zhang, et al. (2001) also point out that real-time recommendations by using data mining is a very computer intensive tasks and is not yet possible with today's processing capability. Though, their results show strong potential for data mining to be used by recommender services. The engine they propose only needs to collect the active user's click trail data and match this to the discovered off line patterns in order to generate a set of recommendations (ibidem).

Another kind of recommender system is the **Web page recommender system**. These recommender systems predict the information needs of users and provide them with recommendations to facilitate their navigation. By analyzing the actions of the current user, the goal is to predict which Web pages will be accessed next. Many Web sites on the Internet use Web page recommender systems to increase their usability and user satisfaction (Göksedef & Gündüz-Ög˘üdücü, 2010).

An example of a recommender system using the **Semantic Web** is given by Blanco-Fernández, et al. (2008). Their recommender system, named AVATAR, will make recommendations to select interesting TV programs for viewers of Digital TV. Instead of using the traditional approaches, they employ a reasoning-based strategy that will discover semantic relationships between users' preferences and the items available in the domain ontology. These relationships provide the system with more information about the users' interest, so it can make more accurate recommendations (Blanco-Fernández, et al., 2008).

Recommender systems that will provide users with suggestions for **documents like papers or scientific articles,** also exist. An example of such a recommender system is *Infonorma,* described by Drumond & Girardi (2008). Infonorma is a system that makes recommendations of legal normative instruments they might be interested in. The system classifies these legal normative instruments represented as Semantic Web documents into legal branches and performs a content-based similarity analysis (Drumond & Girardi, 2008). Infonorma uses an ontology, which is written according to Semantic Web standards, as information source. A recommender system in the legal domain can be very useful, because legal information sources are updated continuously since new laws are written every day (ibidem).

A model of a recommender system that will provide a user with suggestions regarding **news** that might interest them is given by Medo, et al. (2009). The model that they suggest, is very different from other existing recommender systems that use collaborative and content-based filtering. Medo, et al. (2009) aim at creating a model that will personalize news recommendation by observing the past reading patterns of readers, identify their "taste mates" and constructing a directed local neighbourhood network. The model is based on an either "approval" or "disapproval" of the readers. When news is approved, it will spread in the neighbourhood network to the next prospective readers. This process is very comparable to an epidemic spreading in a social network or to a rumour spreading in a society. "Simultaneously with the spreading of news, the network of contacts gradually evolves to best capture the users' similarities" (Medo, et al., 2009).

Zhen, et al. (2010) propose a model of a **inner-enterprise knowledge recommender system**. Because the core of the enterprise is moving towards being knowledge intensive, knowledge management is becoming a critical issue for enterprises' management. As already mentioned in chapter 2, knowledge searching is a very time consuming task. The goal of this system is to provide recommendations for office workers of an enterprise. It will suggest information to the user on topics like technical standards, patents, design formulas, design rules, references to successful or failure cases of product design in the past, contact information of experts, etc. The recommendations will be based on a user profile containing information about the users' role and tasks in the enterprise as well as his background and personal interests.

4.3 Case example: A recommender system to deliver relevant information during chat discussions

The following section will give a summary of the article *Recommendation of Complementary Material during Chat Discussions* (X,2010). Summarizing this article introduces a recommender system and will give a view on the general picture of these systems.

4.3.1 Introduction

The number of technology-based environments that support knowledge sharing is growing in a very fast way. In the context of the World Wide Web, such environments enable the rise of Virtual Learning Communities, that gather people that are geographically scattered but have similar interests. People in these communities exchange knowledge, documents, bibliographic references and other information sources about similar topics.

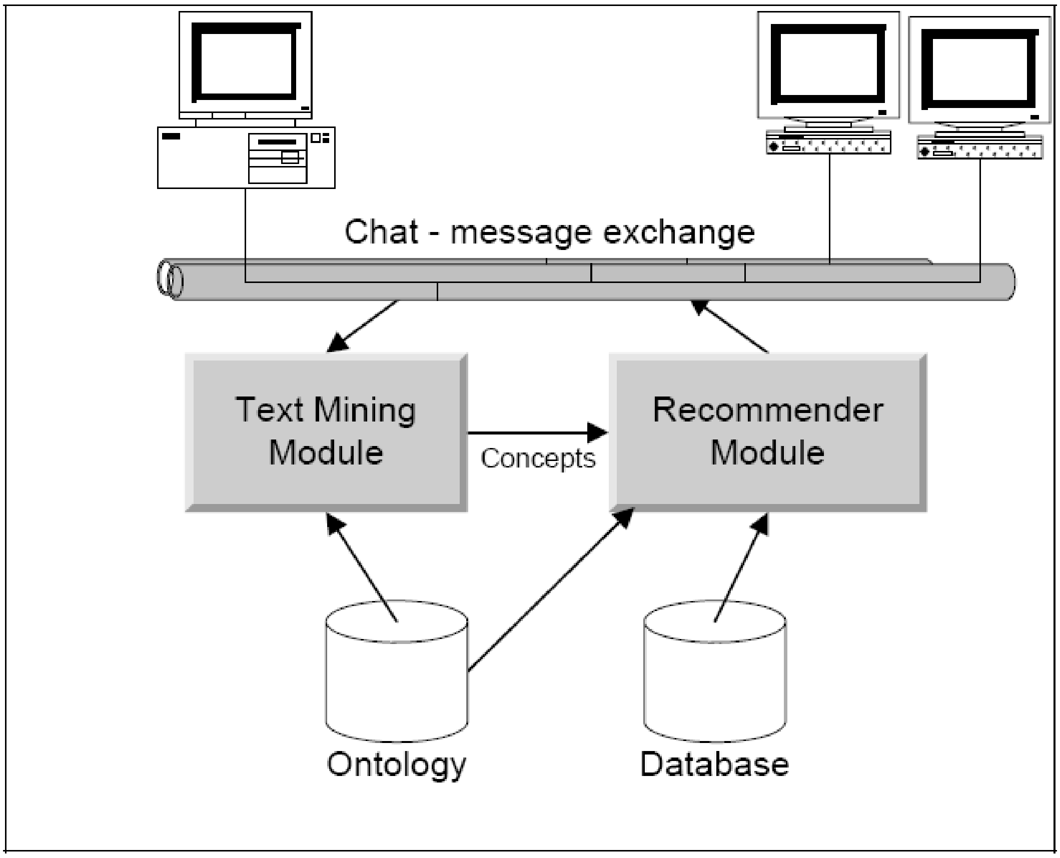
The paper presents a recommender system for online discussions. The system consists in a Web chat, where users exchange messages. The textual messages posted in the chat are analyzed so that relevant complementary information can be recommended during the chat session according to the topics being discussed. Recommendations are personalized to the profile of the users.

4.3.2 Description of the proposed recommender system

The system consists of a Text Mining Module that analyzes each message posted in the chat. The words in the message are compared against terms present in a domain ontology. After that, it passes the identified concept to the recommender system module, that searches in the database for items to suggest.
The **database** is composed by:
- a *Digital Library*, including electronic documents, Web links and bibliographic references;
- a base of *Past Discussions*, containing historical discussions; and
- a *Profile base*, including the profiles of the registered users.

Figure 4.1 shows the architecture of the recommender system

Figure 4.1: Architecture of the recommender system



Source: X, 2010

The **Web chat** works like a traditional chat over the Web, but the chat in this system is specifically designed for the proposed system and it is not open to non-registered users.

The main component of the system is a **Text Mining Module**. It works by examining each message sent in the chat. This module is responsible for identifying themes or subjects in the messages. Themes are identified by comparing words that are sent in the message against terms defined in the ontology. Each message is compared online against all the concepts in the ontology. The concepts identified in the messages represent the topics being discussed in the chat and are forwarded to the Recommender Module.

The **ontology** is used to identify themes in textual messages sent in the chat, to automatically classify items of the digital library and to relate people to subjects for identifying interest areas (stored in the profile of the users). The ontology is also utilized to retrieve items from the Digital Library or to search the base of past discussions. A group of experts should be responsible for creating and updating the ontology. These experts should define the concepts of the domain ontology and the relationships among concepts (the hierarchy).

The **Digital Library** is a repository of information sources containing electronic documents, links to Web pages and bibliographic references.

The **profile base** contains identification of the authorized users. Next to administrative data like name, institution, department, e-mail, etc., the profile base also stores the interest areas of each person, as well an associated degree, informing the degree of interest of user in the area. These areas are related to concepts in the ontology.

This **base of past discussions** records everything that occurs in the chat, during a discussion session. Discussions are stored by sessions, identified by data. Associated to the session, the base must also store who participated in the session, all the messages sent (with a label indicating who sent it), the concept identified in each message, the recommendations that where during the session for each user and the documents that where downloaded or read during the session.

The goal of the **Recommender Module** is to provide information stored in the different bases to the participants of the chat discussion. The action of this module starts when it receives a concept from the Text Mining Module. After this, it searches the different bases for items classified in the same concept. Each time the Text Mining Module identifies a concept in a message, it sends this concept to the Recommender Module that searches the database for more items to recommend. Since the discussion in the chat is synchronous, recommendations should not interrupt the participants of the chart discussion. That is why indications are given in a separate frame and not inside the chat window. To avoid information overload, experts advise that the list of suggestions is minimized. The quality of the recommendations depends directly on the quality of the ontology and on the text mining method that is used on the chat messages and on the documents in the digital library.

An **experiment** of the recommender system was executed. The findings of the experiment pointed out that the majority of the test subject reported benefits when using the system because they did not have to search the digital library for documents and the system returned new and interesting documents. However, none of the students was comfortable to read an entire document during the session. Some test subjects reported that, when they were viewing the content of recommended documents, they lost part of the discussion. On the other side, they reported that this is not necessarily a disadvantage of the system because in some way the process is like searching the Web with search engines. We can conclude that the system is better suited for retrieving documents to the user, hoping that the user will see the documents after the chat session.

Chapter 5: Case study iKnow: Experimenting with the building blocks of a recommender system by applying them in some real situations

5.1 Introduction

In order to get practical insight on the subject of this master thesis, a case study is performed in cooperation with the company iKnow. The company has a great deal of knowledge and experience with recommender systems. The goal of the cooperation with iKnow was to get an understanding of the way these software systems are built and how they work. A step by step analysis of the process that a recommender system will use, reveals the architecture behind these recommender systems.

“**iKnow** is a young, small software company located in Diepenbeek (Belgium). The company is a specialist in “**Knowledge streaming**”, the automatic extraction of important information from data coming from different sources. iKnow also focuses on “**Information forensics**”, a process that reveals links and interprets complex and ambiguous information, so users can understand situations very fast and real-time. The “important information” will be identified by a semantic analysis motor. According to Michaël Brands, co-founder of iKnow, is the relation between things, the network of relationships of great essence” (Doucet, 2009).

iKnow has the following vision: “The digital information universe is continuously expanding. The problem is no longer gathering but targeting the right information to the right people, processes and applications at the right place and at the right time. Simply said the question is no longer “Where can I find it ?” but it became “What’s in it and how can I use it ? More and more this capability to efficiently transform relevant information into applicable knowledge will be a critical success factor for individuals, companies and organisations.”

iKnow specialises in transforming data into information, and thus into knowledge. This process needs to be fast and has to take place autonomously because enterprises in the medical, publishing, financial and public industry no longer have the time to consult knowledge at experts, search for additional information and make a synthesis. iKnow focuses on these domain because they are all confronted with unstructured information flows.

Knowledge Streaming (KS) bundles multiple information sources into a single stream. In this way, enterprises have all the necessary information available in real-time using only one interface.
Transforming data into information, and thus into knowledge ought to be fast and take place autonomously. Enterprises no longer have the time to consult experts for additional information and to make a synthesis. KS automates this process. As soon as new data occur, it automatically streams the information in order to provide enterprises with critical knowledge.

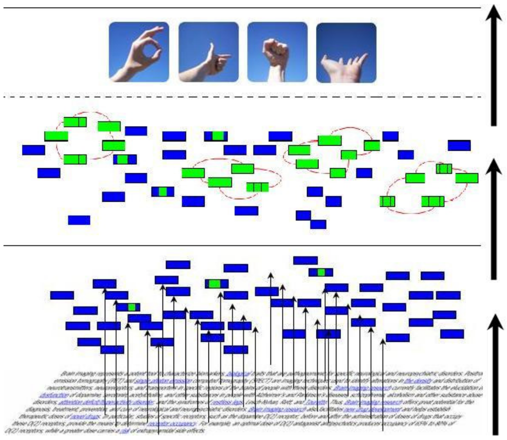
- Real-time knowledge
- From enterprises
- On the spot

At the base of this Knowledge Streaming process lies **Information Forensics (IF).** IF is a digital information expert system, which digs into piles of data with humanlike precision and automatically brings knowledge to the surface. By providing businesses with critical knowledge, IF reveals links between facts so that enterprises can grasp business situations in on the spot. Information Forensics (IF) delivers to enterprises business critical knowledge. In this way, enterprises can grasp medical situations in just a few seconds time and make well considered decisions.

Information Forensics builds up indexes of meaningful elements (Smart Indexing) and links terms via meaningful relations between the terms (int.for®) (ibidem).

Figure 5.1 gives a visualisation of the filtering of the software from iKnow described in the paragraphs above.

Figure 5.1: Information Forensics



int.for®: topicalizes, condenses, streams and gives direct access to knowledge.

int.for®: contextually clusters all meaningful units.

Smart Indexing: gives a complete overview of all meaningful terms.

Source: www.iknow.be

5.2 Case Study

As explained in the introduction, this case study contains a deeper step by step analyses of the building blocks of recommender systems. The case study aims at explaining how a recommender system is built and how it works. The project is split up into different steps that can be linked together in the end. The step by step analyses is explained in the following sections.

As already mentioned, the case study is executed in order to get a better understanding of the building blocks of a recommender system, how it is built, and how it works. Before starting the case study, iKnow provided an explanation about the tasks that needed to be done to reach these goals. After this, it was my task to create examples of these building blocks to clarify these goals. So, it was my task to create the examples in the following sections about various subjects (here: geographic locations and companies) in compliance with the directions given by iKnow, with the intention of revealing more information about the building blocks of the architecture for recommender systems.

5.2.1 Unstructured text and selecting labels

The first step in the case study is to search random **news articles (unstructured text)** from random websites containing words (**labels**) referring to specific subjects. For example: cities, countries, companies, music bands, diseases, etc. One of the articles used in the project is illustrated beneath. In this simplified example, only a geographic locations and companies are used for deeper analysis. In the news article beneath, the company labels are marked in grey and the geographic locations labels in yellow. Recommender systems make use of **Text Mining Modules** to identify these labels by comparing them to the ontology that is defined.

Luxury carmaker Spyker moves assembly line to Coventry

LONDON - Dutch-based luxury carmaker Spyker is to relocate its assembly line to Coventry, creating up to 45 new jobs.

Spyker Cars said its new 20,000sq ft plant at CPP Manufacturing in Whitley, would help it to increase production levels and cut costs.

Moving from Zeewolde in the Netherlands to Coventry will bring Spyker's assembly operation to the same area as some of its major suppliers.

The plant will be able to turn out five Spyker Aileron supercars each week.

Spyker's founder Victor Muller said: "This move makes sense on many different levels.

"More than half our components are sourced from the UK, so moving here will bring us considerable efficiency savings, which is vital for a car company of our size."

In January the firm announced it is to buy Sweden's loss-making Saab from General Motors.

The next step is to link the specific labels in the article with data that is already available on these subjects. The Linked Data community comprises a great amount of datasets that are linked together as explained in the literature review in chapter 3. These datasets are databases of ontologies that describe these subjects.

For example **GeoNames** is a geographical database that contains more than eight million geographical names and 7 million unique features comprising 2.6 million populated places and 2.8 million alternate names (http://www.geonames.org/about.html).
Another database that will be used in this case study is **DBpedia**. “DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link other data sets on the Web to Wikipedia data. The developers of DBpedia hope this will make it easier for the amazing amount of information in Wikipedia to be used in new and interesting ways, and that it might inspire new mechanisms for navigating, linking and improving the encyclopaedia itself”

(http://wiki.dbpedia.org/About). "The DBpedia knowledge base currently describes more than 3.4 million things, out of which 1.5 million are classified in a consistent ontology, including 312,000 persons, 413,000 places, 94,000 music albums, 49,000 films, 15,000 video games, 140,000 organizations, 146,000 species and 4,600 diseases. The DBpedia data set features labels and abstracts for these 3.2 million things in up to 92 different languages; 1,460,000 links to images and 5,543,000 links to external web pages; 4,887,000 external links into other RDF datasets, 565,000 Wikipedia categories, and 75,000 YAGO categories. The DBpedia knowledge base altogether consists of over 1 billion pieces of information (RDF triples) out of which 257 million were extracted from the English edition of Wikipedia and 766 million were extracted from other language editions". (ibidem).

5.2.2 Creating the RDF file

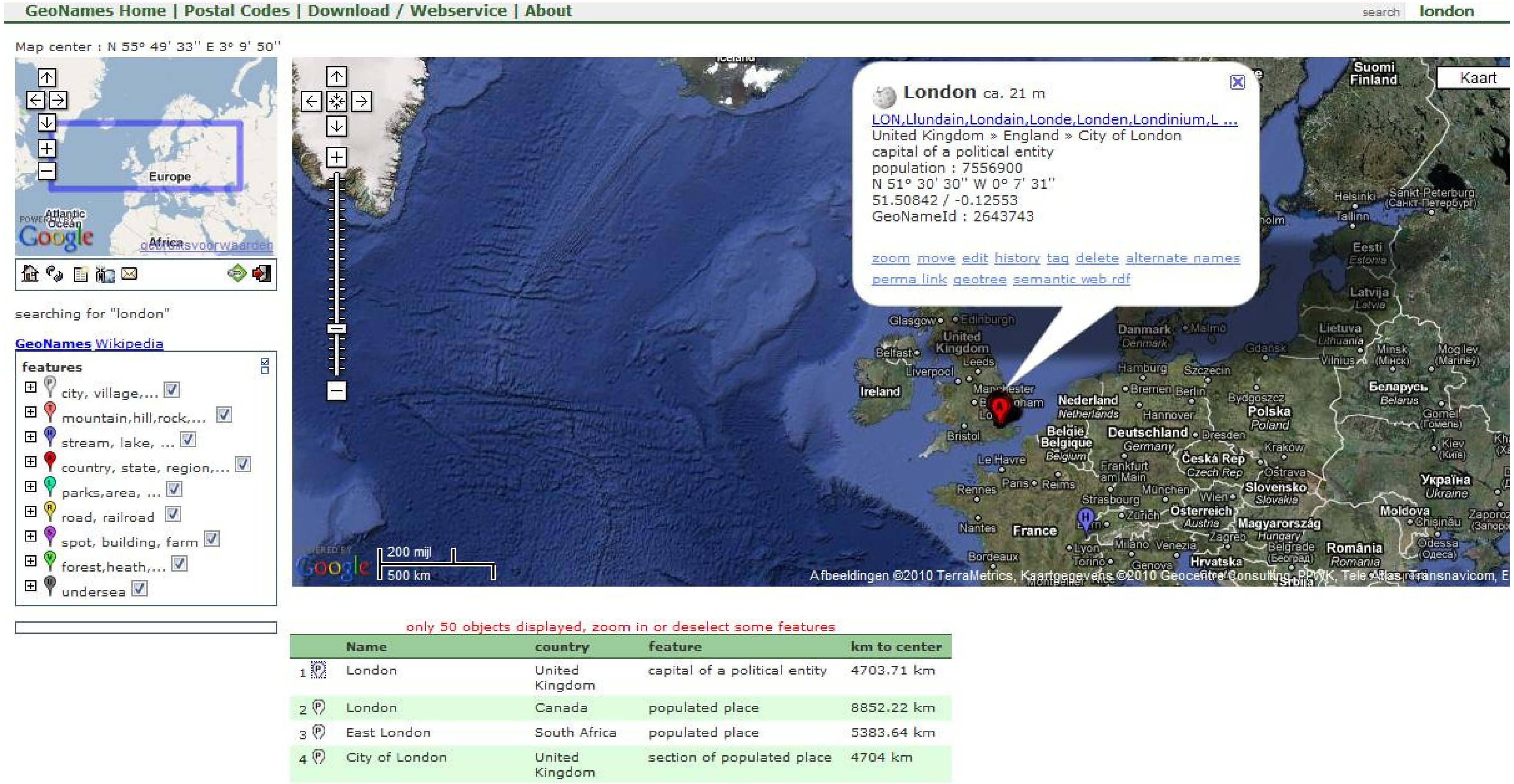
In order to link these labels, extracted from the news article above, with the already existing data in the Linked Data databases (in this example GeoNames and DBpedia), an RDF file has to be created. This file will link the labels selected from the news article with the existing databases that contain information about the label.
In the following examples, the creation of the RDF syntax for two of the labels from the article will be explained to illustrate the different parts of the RDF file. The first example is the geographic location "London" from the GeoNames database and the second example is the company "General motors" from the DBpedia database.

Example 1: GeoNames with the label "London"

- **Query and RDF location:** In order to find the RDF file of the concerning geographic location, the GeoNames database needs to be queried. This action will be demonstrated in the next steps.

At the GeoNames website (http://www.geonames.org/), users can query for a geographic location, in this example: "London". After entering "London" in the search bar, GeoNames will provide a list containing several geographic locations with the string "London" together with the class they belong to. From this list, the geographic location conforming with the search query needs to be selected. After this step, a map shows the results of the query. This map can be seen in figure 5.2.

Figure 5.2: Screenshot of the search query “London” by GeoNames


GeoNames Home | Postal Codes | Download / Webservice | About
search
london
Europe
Atlantic Ocean
Africa
Kaart
200 mijl
500 km
Afbeeldingen ©2010 TerraMetrics, Kaartgegevens ©2010 Geocentre Consulting, PPWK, Tele Atlas, Transnavicom, E

Map center : N 55° 49' 33'' E 3° 9' 50''

searching for "london"

GeoNames Wikipedia

features

- ⊞ city, village,... ☑
- ⊞ mountain,hill,rock,... ☑
- ⊞ stream, lake, ... ☑
- ⊞ country, state, region,... ☑
- ⊞ parks,area, ... ☑
- ⊞ road, railroad ☑
- ⊞ spot, building, farm ☑
- ⊞ forest,heath,... ☑
- ⊞ undersea ☑

London ca. 21 m

LON,Llundain,Londain,Londe,Londen,Londinium,L ...

United Kingdom » England » City of London

capital of a political entity

population : 7556900

N 51° 30' 30'' W 0° 7' 31''

51.50842 / -0.12553

GeoNameId : 2643743

zoom move edit history tag delete alternate names

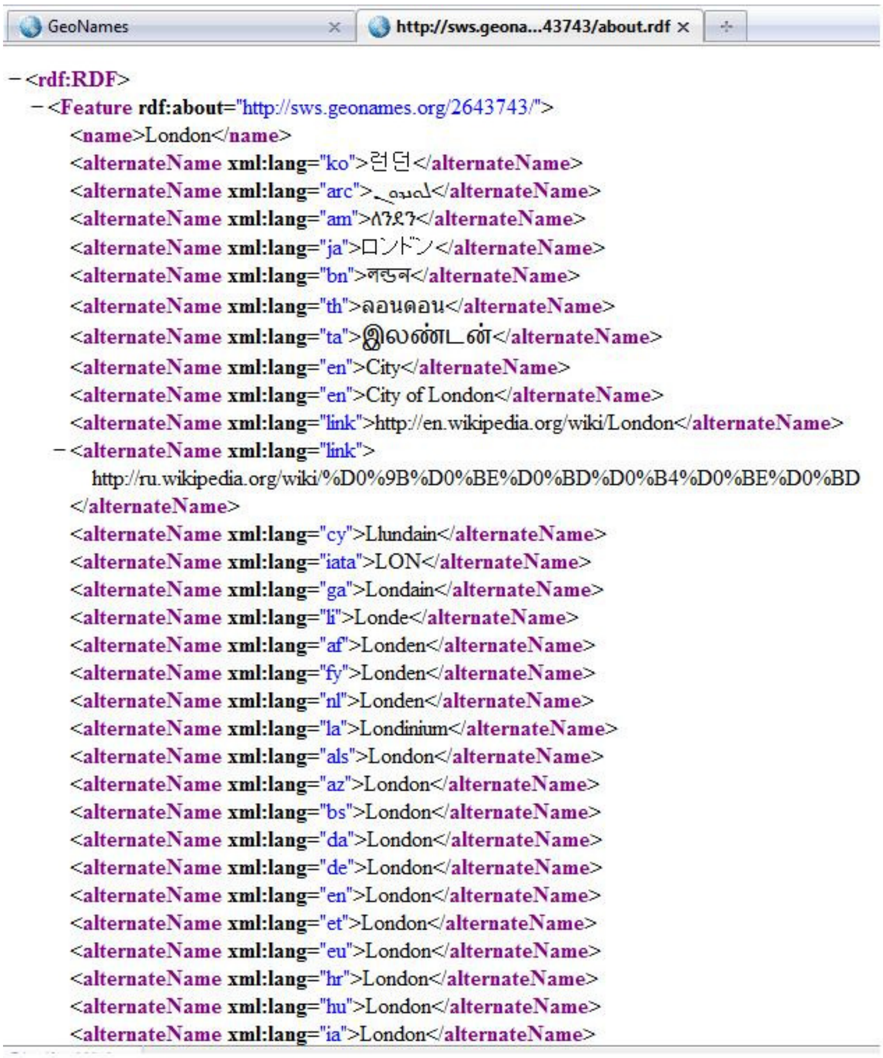
perma link geotree semantic web rdf

only 50 objects displayed, zoom in or deselect some features

	Name	country	feature	km to center
1	London	United Kingdom	capital of a political entity	4703.71 km
2	London	Canada	populated place	8852.22 km
3	East London	South Africa	populated place	5383.64 km
4	City of London	United Kingdom	section of populated place	4704 km

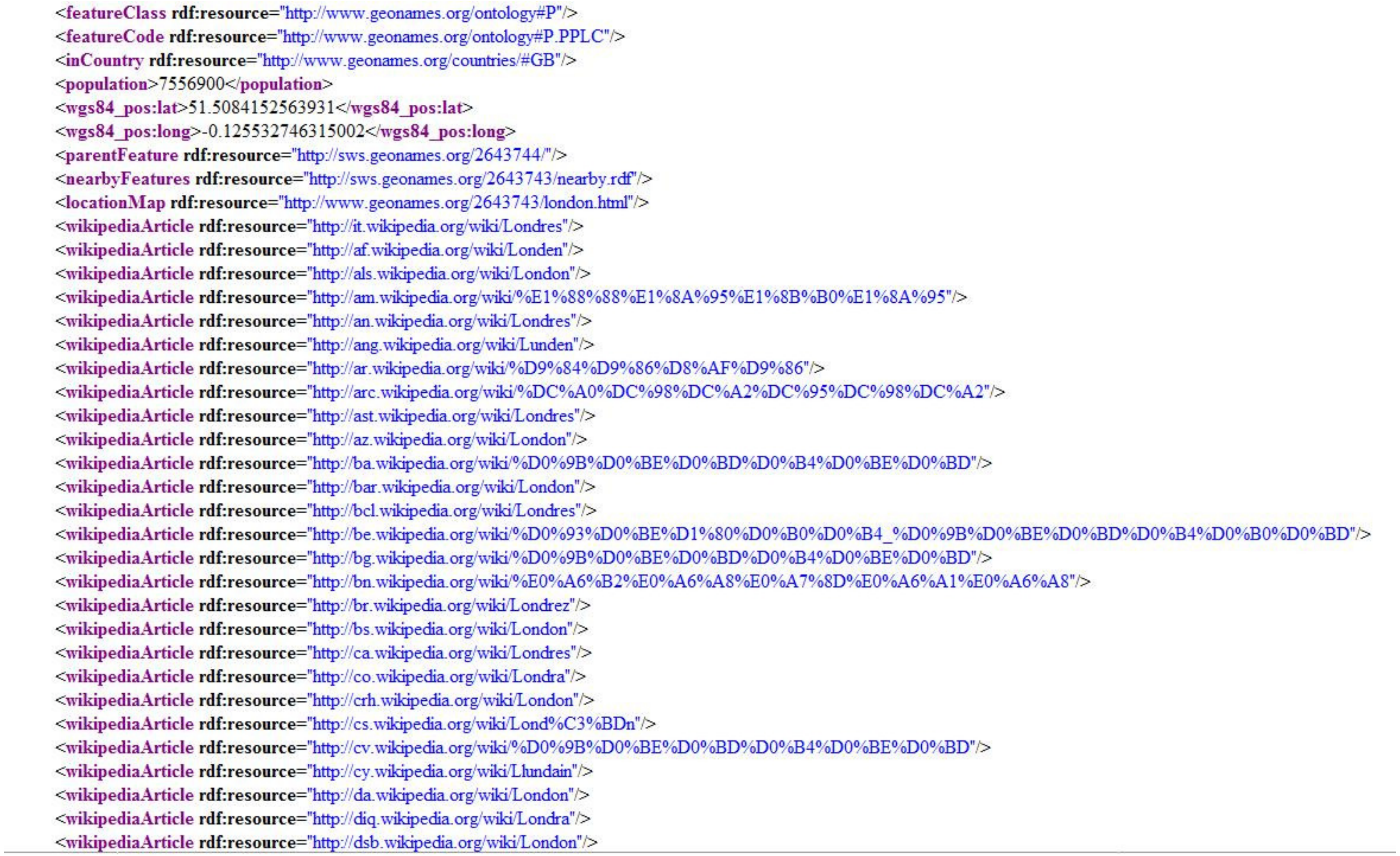
The text balloon, created by GeoNames, not only contains a lot of information such as a description of the geographic location (class), the population, the geographic coordinates, etc., but also the link to RDF file ("semantic web rdf"), partly illustrated in figure 5.3 and figure 5.4. Figure 5.3 shows the first part of the RDF file. It gives the anchor of the city London: "Feature rdf:about=http://sws.geonames.org/2643743/". Next to this anchor, it displays a list of alternate names that refer to London. Figure 5.4 shows the part of the RDF file below the list of the alternate names. This part of the RDF file contains the links to other databases that provide more information about London.

Figure 5.3: Part of the GeoNames RDF file of London (1)



```
-<rdf:RDF>
  -<Feature rdf:about="http://sws.geonames.org/2643743/">
      <name>London</name>
      <alternateName xml:lang="ko">런던</alternateName>
      <alternateName xml:lang="arc">ܠܘܢܕܘܢ</alternateName>
      <alternateName xml:lang="am">ለንደን</alternateName>
      <alternateName xml:lang="ja">ロンドン</alternateName>
      <alternateName xml:lang="bn">লন্ডন</alternateName>
      <alternateName xml:lang="th">ลอนดอน</alternateName>
      <alternateName xml:lang="ta">இலண்டன்</alternateName>
      <alternateName xml:lang="en">City</alternateName>
      <alternateName xml:lang="en">City of London</alternateName>
      <alternateName xml:lang="link">http://en.wikipedia.org/wiki/London</alternateName>
    -<alternateName xml:lang="link">
        http://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BD%D0%B4%D0%BE%D0%BD
      </alternateName>
      <alternateName xml:lang="cy">Llundain</alternateName>
      <alternateName xml:lang="iata">LON</alternateName>
      <alternateName xml:lang="ga">Londain</alternateName>
      <alternateName xml:lang="li">Londe</alternateName>
      <alternateName xml:lang="af">Londen</alternateName>
      <alternateName xml:lang="fy">Londen</alternateName>
      <alternateName xml:lang="nl">Londen</alternateName>
      <alternateName xml:lang="la">Londinium</alternateName>
      <alternateName xml:lang="als">London</alternateName>
      <alternateName xml:lang="az">London</alternateName>
      <alternateName xml:lang="bs">London</alternateName>
      <alternateName xml:lang="da">London</alternateName>
      <alternateName xml:lang="de">London</alternateName>
      <alternateName xml:lang="en">London</alternateName>
      <alternateName xml:lang="et">London</alternateName>
      <alternateName xml:lang="eu">London</alternateName>
      <alternateName xml:lang="hr">London</alternateName>
      <alternateName xml:lang="hu">London</alternateName>
      <alternateName xml:lang="ia">London</alternateName>
```

Figure 5.4: Part of the GeoNames RDF file of London (2)



```
<featureClass rdf:resource="http://www.geonames.org/ontology#P"/>
<featureCode rdf:resource="http://www.geonames.org/ontology#P.PPLC"/>
<inCountry rdf:resource="http://www.geonames.org/countries/#GB"/>
<population>7556900</population>
<wgs84_pos:lat>51.5084152563931</wgs84_pos:lat>
<wgs84_pos:long>-0.125532746315002</wgs84_pos:long>
<parentFeature rdf:resource="http://sws.geonames.org/2643744/"/>
<nearbyFeatures rdf:resource="http://sws.geonames.org/2643743/nearby.rdf"/>
<locationMap rdf:resource="http://www.geonames.org/2643743/london.html"/>
<wikipediaArticle rdf:resource="http://it.wikipedia.org/wiki/Londres"/>
<wikipediaArticle rdf:resource="http://af.wikipedia.org/wiki/Londen"/>
<wikipediaArticle rdf:resource="http://als.wikipedia.org/wiki/London"/>
<wikipediaArticle rdf:resource="http://am.wikipedia.org/wiki/%E1%88%88%E1%8A%95%E1%8B%B0%E1%8A%95"/>
<wikipediaArticle rdf:resource="http://an.wikipedia.org/wiki/Londres"/>
<wikipediaArticle rdf:resource="http://ang.wikipedia.org/wiki/Lunden"/>
<wikipediaArticle rdf:resource="http://ar.wikipedia.org/wiki/%D9%84%D9%86%D8%AF%D9%86"/>
<wikipediaArticle rdf:resource="http://arc.wikipedia.org/wiki/%DC%A0%DC%98%DC%A2%DC%95%DC%98%DC%A2"/>
<wikipediaArticle rdf:resource="http://ast.wikipedia.org/wiki/Londres"/>
<wikipediaArticle rdf:resource="http://az.wikipedia.org/wiki/London"/>
<wikipediaArticle rdf:resource="http://ba.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BD%D0%B4%D0%BE%D0%BD"/>
<wikipediaArticle rdf:resource="http://bar.wikipedia.org/wiki/London"/>
<wikipediaArticle rdf:resource="http://bcl.wikipedia.org/wiki/Londres"/>
<wikipediaArticle rdf:resource="http://be.wikipedia.org/wiki/%D0%93%D0%BE%D1%80%D0%B0%D0%B4_%D0%9B%D0%BE%D0%BD%D0%B4%D0%B0%D0%BD"/>
<wikipediaArticle rdf:resource="http://bg.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BD%D0%B4%D0%BE%D0%BD"/>
<wikipediaArticle rdf:resource="http://bn.wikipedia.org/wiki/%E0%A6%B2%E0%A6%A8%E0%A7%8D%E0%A6%A1%E0%A6%A8"/>
<wikipediaArticle rdf:resource="http://br.wikipedia.org/wiki/Londrez"/>
<wikipediaArticle rdf:resource="http://bs.wikipedia.org/wiki/London"/>
<wikipediaArticle rdf:resource="http://ca.wikipedia.org/wiki/Londres"/>
<wikipediaArticle rdf:resource="http://co.wikipedia.org/wiki/Londra"/>
<wikipediaArticle rdf:resource="http://crh.wikipedia.org/wiki/London"/>
<wikipediaArticle rdf:resource="http://cs.wikipedia.org/wiki/Lond%C3%BDn"/>
<wikipediaArticle rdf:resource="http://cv.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BD%D0%B4%D0%BE%D0%BD"/>
<wikipediaArticle rdf:resource="http://cy.wikipedia.org/wiki/Llundain"/>
<wikipediaArticle rdf:resource="http://da.wikipedia.org/wiki/London"/>
<wikipediaArticle rdf:resource="http://diq.wikipedia.org/wiki/Londra"/>
<wikipediaArticle rdf:resource="http://dsb.wikipedia.org/wiki/London"/>
```

- **Parts of the RDF syntax:**

The complete RDF syntax of this label that will be admitted in the RDF file is:

london;capital of a political entity;xmlns:geonames =
"http://www.geonames.org/ontology#" <rdf:type
rdf:resource="http://www.geonames.org/2643743"/>

If we subdivide this syntax we come to these different segments:

a. london: label, word from the news article
b. capital of a political entity: ontological concept according to the GeoNames ontology
c. xmlns:geonames = http://www.geonames.org/ontology#: defining the namespace (see infra, section 3.2.2, p.15).
d. <rdf:type rdf:resource="http://www.geonames.org/2643743"/>: anchor, this is a unique code that the database uses to indentify a single unit of the database.

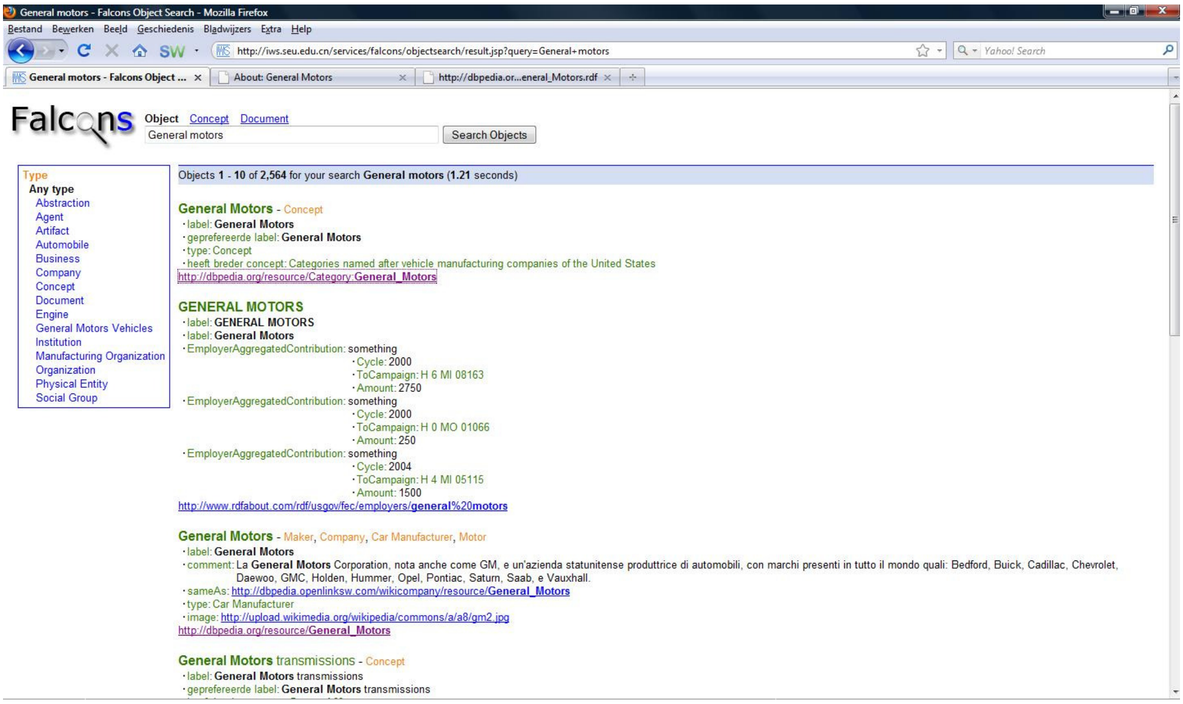
Example 2: DBpedia with the label "General Motors"

- **Query and RDF location:**

The second example is the company "General Motors", which is part of the DBpedia database.
An easy way to search the Semantic Web is provided by Falcons (http://iws.seu.edu.cn/services/falcons/objectsearch/index.jsp)
Falcons is a keyword-based search engine that queries the Semantic Web. Falcons provides a keyword-based search engine for URIs that identify objects, concepts (classes and properties), and documents on the Semantic Web. Falcons allows users to search the Semantic Web in the same way as the Google search engine. Figure 5.5 gives the result given by Falcons when searching for "General Motors". Because in this example DBpedia is used, the third result needs to be used, namely http://dbpedia.org/resource/General_Motors. This link gives a very clear arrangement of the available data of General Motors is given. Figure 5.6 illustrates this with a screenshot of this link. By clicking on the RDF Data symbol (upper right corner of figure 5.6), the

RDF data file of General Motors is provided. A screenshot of this RDF file is shown in figure 5.7.

Figure 5.5: Screenshot of the Falcons search engine


General motors - Falcons Object Search - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help
SW
http://iws.seu.edu.cn/services/falcons/objectsearch/result.jsp?query=General+motors
Yahoo! Search
General motors - Falcons Object ...
About: General Motors
http://dbpedia.or...eneral_Motors.rdf
Falcons
Object
Concept
Document
Search Objects

General motors

Type
Any type
Abstraction
Agent
Artifact
Automobile
Business
Company
Concept
Document
Engine
General Motors Vehicles
Institution
Manufacturing Organization
Organization
Physical Entity
Social Group

Objects **1** - **10** of **2,564** for your search **General motors** (**1.21** seconds)

General Motors - Concept
- label: **General Motors**
- geprefereerde label: **General Motors**
- type: Concept
- heeft breder concept: Categories named after vehicle manufacturing companies of the United States

http://dbpedia.org/resource/Category:**General_Motors**

GENERAL MOTORS
- label: GENERAL MOTORS
- label: **General Motors**
- EmployerAggregatedContribution: something
    - Cycle: 2000
    - ToCampaign: H 6 MI 08163
    - Amount: 2750
- EmployerAggregatedContribution: something
    - Cycle: 2000
    - ToCampaign: H 0 MO 01066
    - Amount: 250
- EmployerAggregatedContribution: something
    - Cycle: 2004
    - ToCampaign: H 4 MI 05115
    - Amount: 1500

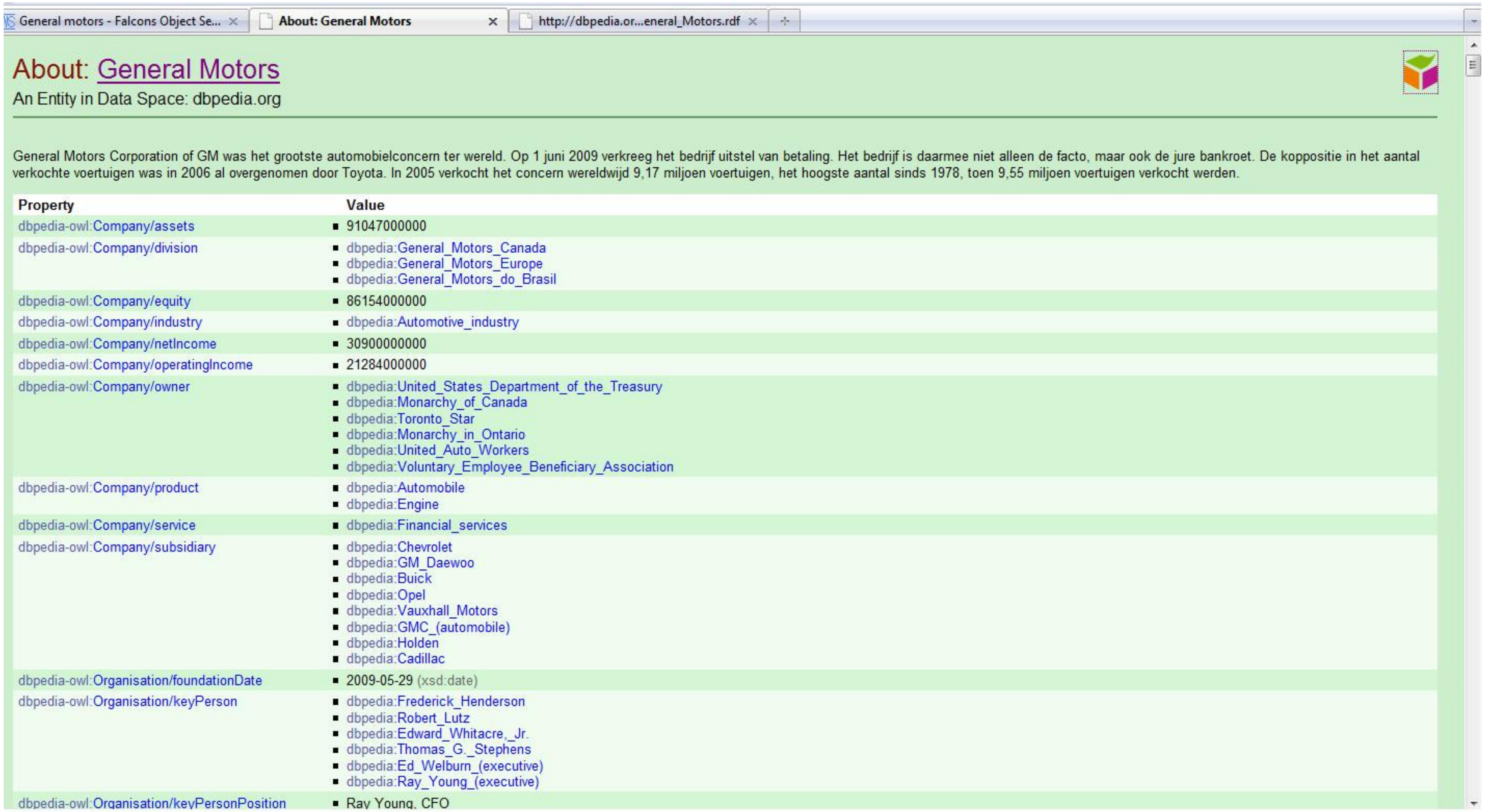
http://www.rdfabout.com/rdf/usgov/fec/employers/**general%20motors**

General Motors - Maker, Company, Car Manufacturer, Motor
- label: **General Motors**
- comment: La **General Motors** Corporation, nota anche come GM, e un'azienda statunitense produttrice di automobili, con marchi presenti in tutto il mondo quali: Bedford, Buick, Cadillac, Chevrolet, Daewoo, GMC, Holden, Hummer, Opel, Pontiac, Saturn, Saab, e Vauxhall.
- sameAs: http://dbpedia.openlinksw.com/wikicompany/resource/**General_Motors**
- type: Car Manufacturer
- image: http://upload.wikimedia.org/wikipedia/commons/a/a8/gm2.jpg

http://dbpedia.org/resource/**General_Motors**

General Motors transmissions - Concept
- label: **General Motors** transmissions
- geprefereerde label: **General Motors** transmissions

Figure 5.6: DBpedia (resource) page of General Motors



General motors - Falcons Object Se...
About: General Motors
http://dbpedia.or...eneral_Motors.rdf

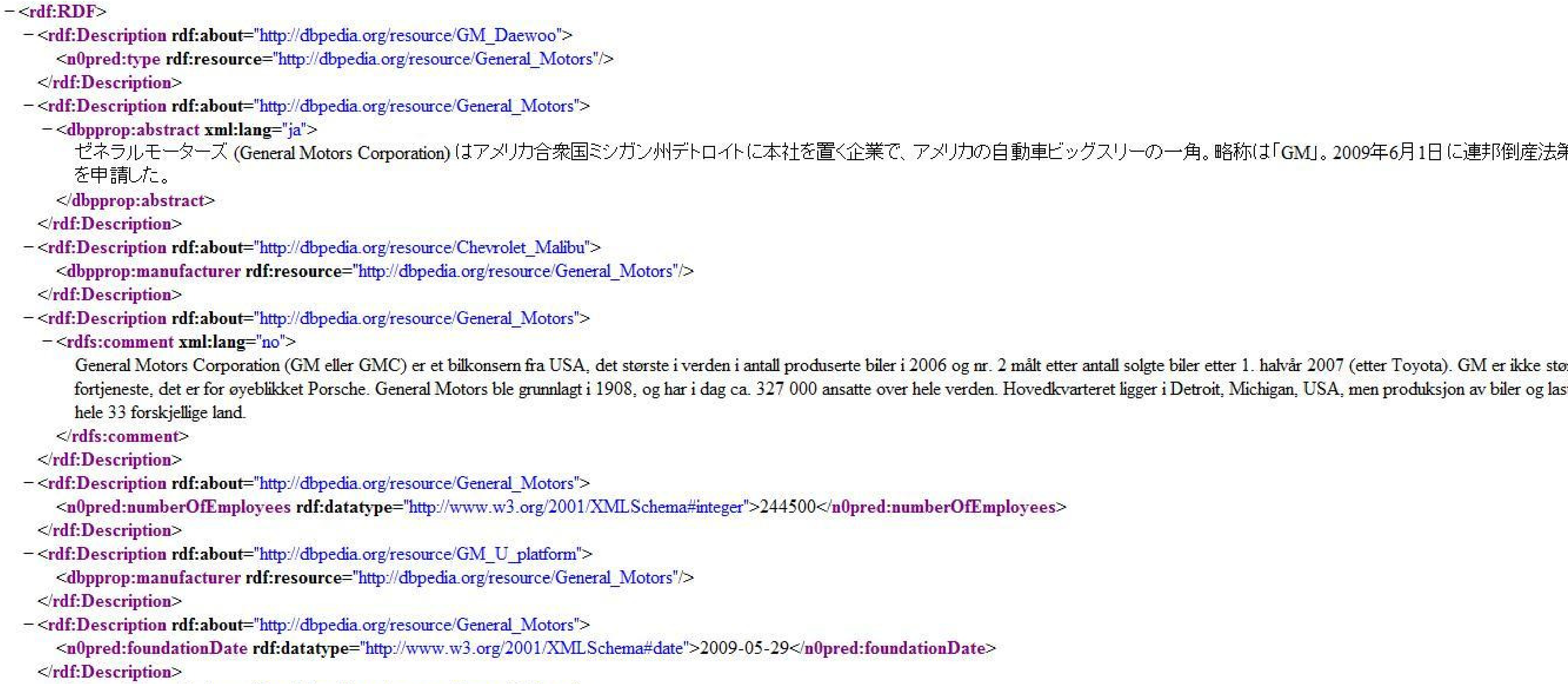
About: General Motors

An Entity in Data Space: dbpedia.org

General Motors Corporation of GM was het grootste automobielconcern ter wereld. Op 1 juni 2009 verkreeg het bedrijf uitstel van betaling. Het bedrijf is daarmee niet alleen de facto, maar ook de jure bankroet. De koppositie in het aantal verkochte voertuigen was in 2006 al overgenomen door Toyota. In 2005 verkocht het concern wereldwijd 9,17 miljoen voertuigen, het hoogste aantal sinds 1978, toen 9,55 miljoen voertuigen verkocht werden.

Property	Value
dbpedia-owl:Company/assets	▪ 91047000000
dbpedia-owl:Company/division	▪ dbpedia:General_Motors_Canada ▪ dbpedia:General_Motors_Europe ▪ dbpedia:General_Motors_do_Brasil
dbpedia-owl:Company/equity	▪ 86154000000
dbpedia-owl:Company/industry	▪ dbpedia:Automotive_industry
dbpedia-owl:Company/netIncome	▪ 30900000000
dbpedia-owl:Company/operatingIncome	▪ 21284000000
dbpedia-owl:Company/owner	▪ dbpedia:United_States_Department_of_the_Treasury ▪ dbpedia:Monarchy_of_Canada ▪ dbpedia:Toronto_Star ▪ dbpedia:Monarchy_in_Ontario ▪ dbpedia:United_Auto_Workers ▪ dbpedia:Voluntary_Employee_Beneficiary_Association
dbpedia-owl:Company/product	▪ dbpedia:Automobile ▪ dbpedia:Engine
dbpedia-owl:Company/service	▪ dbpedia:Financial_services
dbpedia-owl:Company/subsidiary	▪ dbpedia:Chevrolet ▪ dbpedia:GM_Daewoo ▪ dbpedia:Buick ▪ dbpedia:Opel ▪ dbpedia:Vauxhall_Motors ▪ dbpedia:GMC_(automobile) ▪ dbpedia:Holden ▪ dbpedia:Cadillac
dbpedia-owl:Organisation/foundationDate	▪ 2009-05-29 (xsd:date)
dbpedia-owl:Organisation/keyPerson	▪ dbpedia:Frederick_Henderson ▪ dbpedia:Robert_Lutz ▪ dbpedia:Edward_Whitacre,_Jr. ▪ dbpedia:Thomas_G._Stephens ▪ dbpedia:Ed_Welburn_(executive) ▪ dbpedia:Ray_Young_(executive)
dbpedia-owl:Organisation/keyPersonPosition	▪ Ray Young, CFO

Figure 5.7: Part of the DBpedia RDF file of General Motors



```
-<rdf:RDF>
  -<rdf:Description rdf:about="http://dbpedia.org/resource/GM_Daewoo">
     <n0pred:type rdf:resource="http://dbpedia.org/resource/General_Motors"/>
   </rdf:Description>
  -<rdf:Description rdf:about="http://dbpedia.org/resource/General_Motors">
    -<dbpprop:abstract xml:lang="ja">
       ゼネラルモーターズ (General Motors Corporation) はアメリカ合衆国ミシガン州デトロイトに本社を置く企業で、アメリカの自動車ビッグスリーの一角。略称は「GM」。2009年6月1日に連邦倒産法
       を申請した。
     </dbpprop:abstract>
   </rdf:Description>
  -<rdf:Description rdf:about="http://dbpedia.org/resource/Chevrolet_Malibu">
     <dbpprop:manufacturer rdf:resource="http://dbpedia.org/resource/General_Motors"/>
   </rdf:Description>
  -<rdf:Description rdf:about="http://dbpedia.org/resource/General_Motors">
    -<rdfs:comment xml:lang="no">
       General Motors Corporation (GM eller GMC) er et bilkonsern fra USA, det største i verden i antall produserte biler i 2006 og nr. 2 målt etter antall solgte biler etter 1. halvår 2007 (etter Toyota). GM er ikke stø
       fortjeneste, det er for øyeblikket Porsche. General Motors ble grunnlagt i 1908, og har i dag ca. 327 000 ansatte over hele verden. Hovedkvarteret ligger i Detroit, Michigan, USA, men produksjon av biler og las
       hele 33 forskjellige land.
     </rdfs:comment>
   </rdf:Description>
  -<rdf:Description rdf:about="http://dbpedia.org/resource/General_Motors">
     <n0pred:numberOfEmployees rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">244500</n0pred:numberOfEmployees>
   </rdf:Description>
  -<rdf:Description rdf:about="http://dbpedia.org/resource/GM_U_platform">
     <dbpprop:manufacturer rdf:resource="http://dbpedia.org/resource/General_Motors"/>
   </rdf:Description>
  -<rdf:Description rdf:about="http://dbpedia.org/resource/General_Motors">
     <n0pred:foundationDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2009-05-29</n0pred:foundationDate>
   </rdf:Description>
```

- **Parts of the RDF syntax:**

The complete RDF syntax of the company label that will be admitted in the RDF file is:

general motors;company;xmlns:dbpedia="http://dbpedia.org/ontology/" <rdf:type rdf:resource="http://dbpedia.org/page/General_Motors"/>

If we subdivide this syntax we come to these different segment:

a. General motors: label, word from the news article
b. company: ontological concept according to the DBpedia ontology
c. "http://dbpedia.org/ontology/": defining the namespace (see infra, section 3.2.2, p.15).
d. <rdf:type rdf:resource="http://dbpedia.org/page/General_Motors"/>: anchor, this is a unique code that the database uses to indentify a single unit of the database.

A common remark about this RDF syntax is that the unstructured text may contain several labels that have the same meaning. For example the label "General Motors" has the same meaning as the label "General Motor Company", but the link between these to labels will not be made automatically. That is why the RDF syntax has created "alternate names". In order to make these two different labels consistent with each other, the following syntax needs to be included in the RDF file.

general motors company; **alternatename**;
xmlns:dbpedia="http://dbpedia.org/ontology/" <rdf:type
rdf:resource="http://dbpedia.org/page/General_Motors"/>

This can be very important for RDF files about geographic locations, because the notation of geographic names may differ significantly across languages. So by including the "alternatename" syntax in the RDF file for these labels, the corresponding information is available for queries in different languages.

On the **next page**, the **complete RDF syntax** for the geographical and company labels from the **news article** is given. This example contains only **one** news article. This needs repeated for a series of news articles and also for other subjects, like music bands, diseases, sport topics, etc. In the end, the goal is to link a lot of articles, so that the labels from several articles are linked.

RDF of geographical locations from news article (GeoNames):

london;capital of a political entity;xmlns:geonames = "http://www.geonames.org/ontology#" <rdf:type rdf:resource="http://www.geonames.org/2643743"/>

coventry;populated place;xmlns:geonames = "http://www.geonames.org/ontology#" <rdf:type rdf:resource="http://www.geonames.org/2652221"/>

zeewolde;populated place;xmlns:geonames = "http://www.geonames.org/ontology#" <rdf:type rdf:resource="http://www.geonames.org/2743997"/>

netherlands;independent political entity;xmlns:geonames = "http://www.geonames.org/ontology#" <rdf:type rdf:resource="http://www.geonames.org/2750405"/>

the netherlands;alternatename;xmlns:geonames = "http://www.geonames.org/ontology#" <rdf:type rdf:resource="http://www.geonames.org/2750405"/>

united kingdom;independent political entity;xmlns:geonames = "http://www.geonames.org/ontology#" <rdf:type rdf:resource="http://www.geonames.org/2635167"/>

uk;alternatename;xmlns:geonames = "http://www.geonames.org/ontology#" <rdf:type rdf:resource="http://www.geonames.org/2635167"/>

sweden;independent political entity;xmlns:geonames = "http://www.geonames.org/ontology#" <rdf:type rdf:resource="http://www.geonames.org/2661886"/>

RDF of companies from news article (DBpedia):

saab;company;xmlns:dbpedia="http://dbpedia.org/ontology/" <rdf:type rdf:resource="http://dbpedia.org/page/Saab"/>

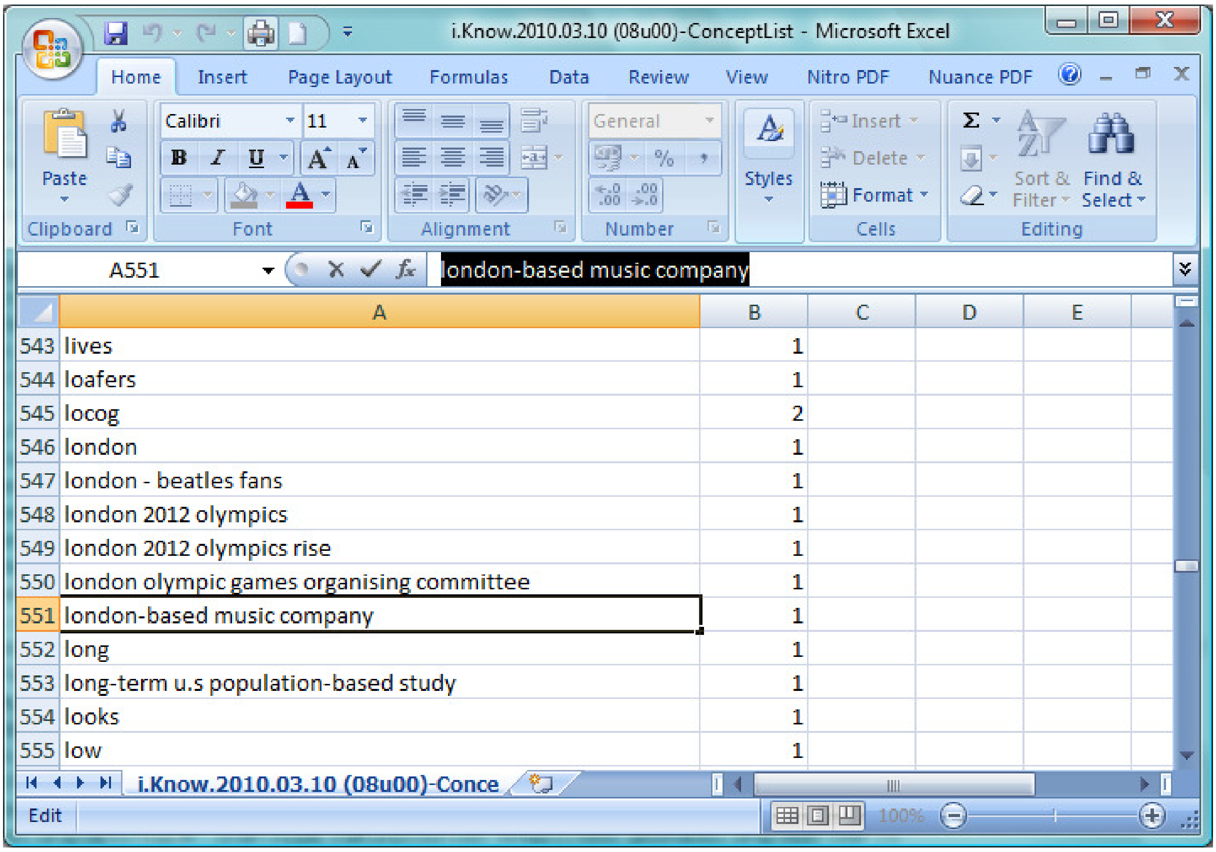
spyker;company;xmlns:dbpedia="http://dbpedia.org/ontology/" <rdf:type rdf:resource="http://dbpedia.org/page/Spyker_Cars"/>

general motors;company;xmlns:dbpedia="http://dbpedia.org/ontology/" <rdf:type rdf:resource="http://dbpedia.org/page/General_Motors"/>

5.2.3 Linking with similar labels

The next step in the process is to link other similar labels in the articles with the existing keys. Before this can be done, a concept cluster has to be created. The concept cluster in this case study was made by iKnow with specialised software. This software analyzes unstructured text (in this case: the news articles). After the analysis, the software creates an output file that contains a list of concepts. The software is designed to recognize related words: clusters. A part of this output file is illustrated in figure 5.8.

Figure 5.8: Example of a part of a concept list



A551 | london-based music company

	A	B	C	D	E
543	lives	1			
544	loafers	1			
545	locog	2			
546	london	1			
547	london - beatles fans	1			
548	london 2012 olympics	1			
549	london 2012 olympics rise	1			
550	london olympic games organising committee	1			
551	london-based music company	1			
552	long	1			
553	long-term u.s population-based study	1			
554	looks	1			
555	low	1			

This figure shows an example of the concept "london-based music company"
This concept was a part of a sentence from one of the newspaper articles. A closer look at the output of this will make this process more comprehensible.

Terra Firma may inject as much as 120 million pounds into EMI to prevent the London-based music company from breaching debt levels.

The software analyzes the phrase and gives the following result (table 5.1).

Table 5.1: Concept list of the above sentence

Concept	**Frequency**
terra firma	1
120 million pounds	1
emi	3
london-based music company	1
breaching debt levels	1

As mentioned earlier, these concepts need to be linked with the existing keys. For example, the concept “london-based music company” can be linked to the existing key for the city London (step 5.2.2).

Example 1:

The RDF syntax of “london” is (see 5.2.2):

london;capital of a political entity;xmlns:geonames =
"http://www.geonames.org/ontology#" <rdf:type
rdf:resource="http://www.geonames.org/2643743"/>

The string “london” can be replaced by the related concept “london-based music company":

london-based music company;capital of a political entity;xmlns:geonames =
"http://www.geonames.org/ontology#" <rdf:type
rdf:resource="http://www.geonames.org/2643743"/>

Example 2:

In the following example, the subject of the RDF syntax is a company. The article in 5.2.1 includes the following sentence:

In January the firm announced it is to buy Sweden's loss-making Saab from General Motors.

One of the concepts that the software identifies is "sweden's loss-making saab"

If we analyze this concept, we can see that the concept "sweden's loss-making saab" includes two existing keys, namely "sweden" and "saab" (see 5.2.2). The RDF syntax of these two keys is:

sweden;independent political entity;xmlns:geonames =
"http://www.geonames.org/ontology#" <rdf:type
rdf:resource="http://www.geonames.org/2661886"/>

saab;company;xmlns:dbpedia="http://dbpedia.org/ontology/" <rdf:type
rdf:resource="http://dbpedia.org/page/Saab"/>

The concept "sweden's loss-making saab" can be linked to "sweden" or to "saab". In this example it is linked to "saab".

sweden's loss-making saab;company;xmlns:dbpedia="http://dbpedia.org/ontology/"
<rdf:type rdf:resource="http://dbpedia.org/page/Saab"/>

5.2.4 Query with SPARQL

SPARQL queries can be executed at a **SPARQL endpoint.**
The SPARQL endpoint that will be used in the next examples, is the DBpedia SPARQL endpoint: http://dbpedia.org/snorql/. At this SPARQL endpoint the whole DBpedia database can be queried.
The most important prefixes used in SPARQL queries are already included in this endpoint, so there is no need to add them again in the query itself.
To be consistent with the previous examples, geographic locations and companies are queried.

a) Writing of a SPARQL query

The first step in executing SPARQL queries is writing the query itself. For more information about SPARQL queries, see section 3.4 (on page 17).

Example of a geographic location

The following query is an example of a query written for the country "**Belgium**". As can be seen in SPARQL query 1, the name (commonName), capital, population (populationCensus), GDP (gdpPpp), GDP per capita (gdpPppPerCapita), currency and official languages will be selected.

SPARQL Query 1

```
SELECT ?commonName ?capital ?populationCensus ?gdpPpp ?gdpPppPerCapita ?currency
?officialLanguages
WHERE {
{ <http://dbpedia.org/resource/Belgium> <http://dbpedia.org/property/commonName>
?commonName}
{ <http://dbpedia.org/resource/Belgium> <http://dbpedia.org/ontology/capital>
?capital }
{ <http://dbpedia.org/resource/Belgium>
<http://dbpedia.org/property/populationCensus>
?populationCensus}
{ <http://dbpedia.org/resource/Belgium> <http://dbpedia.org/property/gdpPpp>
?gdpPpp }
{ <http://dbpedia.org/resource/Belgium>
<http://dbpedia.org/property/gdpPppPerCapita>
?gdpPppPerCapita}
{ <http://dbpedia.org/resource/Belgium> <http://dbpedia.org/property/currency>
?currency}
{ <http://dbpedia.org/resource/Belgium>
<http://dbpedia.org/property/officialLanguages>
?officialLanguages}
}
```

Example of a company

The following query is written for the company "**General Motors**". As can be seen in SPARQL query 2, the name, industry, location, revenue, number of employees and homepage will be selected.

SPARQL Query 2

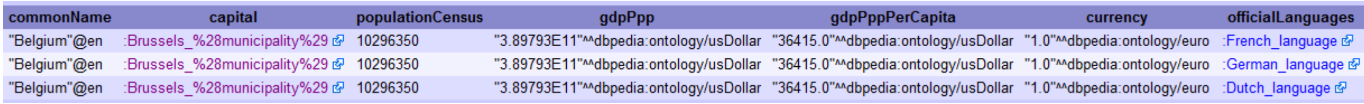
```
SELECT ?name ?industry ?type ?location ?revenue ?numEmployees ?homepage
WHERE {
{ <http://dbpedia.org/resource/General_Motors> <http://dbpedia.org/property/name>
?name }
{ <http://dbpedia.org/resource/General_Motors>
<http://dbpedia.org/ontology/industry>
?industry }
{ <http://dbpedia.org/resource/General_Motors> <http://dbpedia.org/property/type>
?type }
{ <http://dbpedia.org/resource/General_Motors>
<http://dbpedia.org/ontology/location>
?location }
{ <http://dbpedia.org/resource/General_Motors>
<http://dbpedia.org/ontology/revenue>
?revenue }
{ <http://dbpedia.org/resource/General_Motors>
<http://dbpedia.org/property/numEmployees>
?numEmployees }
{ <http://dbpedia.org/resource/General_Motors>
<http://dbpedia.org/property/homepage>
?homepage }
}
```

b) Enter query in SPARQL endpoint

Example of a geographic location

Entering SPARQL query 1 in the DBpedia endpoint gives the results in figure 5.9.

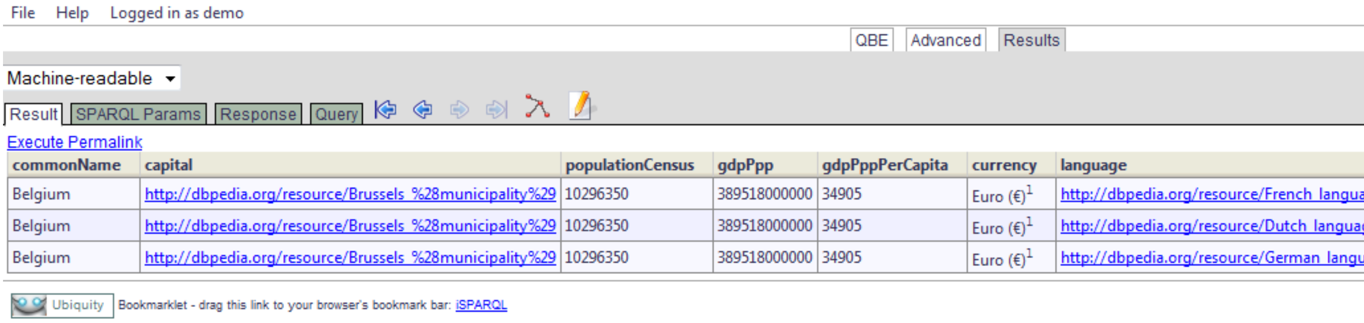
Figure 5.9: Results from SPARQL query 1 (http://dbpedia.org/snorql/)



commonName	capital	populationCensus	gdpPpp	gdpPppPerCapita	currency	officialLanguages
"Belgium"@en	:Brussels_%28municipality%29	10296350	"3.89793E11"^^dbpedia:ontology/usDollar	"36415.0"^^dbpedia:ontology/usDollar	"1.0"^^dbpedia:ontology/euro	:French_language
"Belgium"@en	:Brussels_%28municipality%29	10296350	"3.89793E11"^^dbpedia:ontology/usDollar	"36415.0"^^dbpedia:ontology/usDollar	"1.0"^^dbpedia:ontology/euro	:German_language
"Belgium"@en	:Brussels_%28municipality%29	10296350	"3.89793E11"^^dbpedia:ontology/usDollar	"36415.0"^^dbpedia:ontology/usDollar	"1.0"^^dbpedia:ontology/euro	:Dutch_language

OpenLink iSPARQL (http://demo.openlinksw.com/isparql/) is better suited to present SPARQL queries. This is illustrated in figure 5.10.

Figure 5.10: Results from SPARQL query 1 (http://demo.openlinksw.com/isparql/)



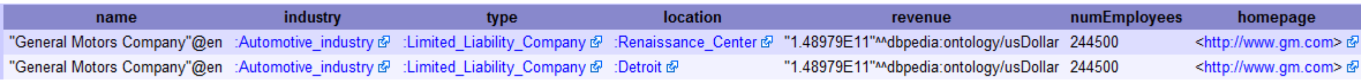
commonName	capital	populationCensus	gdpPpp	gdpPppPerCapita	currency	language
Belgium	http://dbpedia.org/resource/Brussels_%28municipality%29	10296350	389518000000	34905	Euro (€)[1]	http://dbpedia.org/resource/French_langua
Belgium	http://dbpedia.org/resource/Brussels_%28municipality%29	10296350	389518000000	34905	Euro (€)[1]	http://dbpedia.org/resource/Dutch_languag
Belgium	http://dbpedia.org/resource/Brussels_%28municipality%29	10296350	389518000000	34905	Euro (€)[1]	http://dbpedia.org/resource/German_langu

Ubiquity
Bookmarklet - drag this link to your browser's bookmark bar: iSPARQL

Example of a company

Entering SPARQL query 2 in the DBpedia endpoint gives the results in figure 5.11.

Figure 5.11: Results from SPARQL query 2 (http://dbpedia.org/snorql/)



name	industry	type	location	revenue	numEmployees	homepage
"General Motors Company"@en	:Automotive_industry	:Limited_Liability_Company	:Renaissance_Center	"1.48979E11"^^dbpedia:ontology/usDollar	244500	<http://www.gm.com>
"General Motors Company"@en	:Automotive_industry	:Limited_Liability_Company	:Detroit	"1.48979E11"^^dbpedia:ontology/usDollar	244500	<http://www.gm.com>

5.2.5 Visualising the results

The last step in this case study is to visualise the results of these SPARQL queries. The information that was available on the researched labels, it can be visualised in tables, maps, lists of recommended articles or websites, etc. In this case study, simple visualisations from Google API will be used (http://code.google.com/intl/nl/apis/charttools/). These Google Chart Tools can be used to create image charts or interactive charts (in JavaScript). To illustrate the possibilities of these visualisation, some examples will be given.

Example 1: Visualisation of information about countries

This example will visualise, among others, SPARQL Query 1 from the previous part (5.2.4). For this example, we have added other countries to the example using the same query.
After slightly adjusting the output from the SPARQL queries (changing the column titles, deleting hyperlinks, sorting the numbers, etc.) the results are displayed in table 5.2.

Table 5.2: Input for visualisation for example 1

Name	Capital	Population Estimate	GDP	GDP per capita	Currency	Language
Belgium	Brussels	10.665.867	$ 389.518.000.000,00	$ 34.905,00	Euro (€)	Dutch language, French language, German Language
the Netherlands	Amsterdam	16.500.156	$ 675.375.000.000,00	$ 40.431,00	Euro (€)	Dutch language
Germany	Berlin	82.060.000	$ 2.910.000.000.000,00	$ 35.442,00	Euro (€)	German language
Russia	Moscow	142.008.838	$ 2.261.000.000.000,00	$ 15.922,00	Russian ruble	Russian language
Iran	Tehran	74.196.000	$ 819.799.000.000,00	$ 11.250,00	Rial (ریال)	Persian language
Finland	Helsinki	5.342.344	$ 190.862.000.000,00	$ 36.217,00	Euro (€)	Finnish language, Swedish language
Cuba	Havana	11.451.652	$ 108.200.000.000,00	$ 9.500,00	Cuban peso	Spanish language
Argentina	Buenos Aires	40.482.000	$ 572.860.000.000,00	$ 14.413,00	Argentine peso	Spanish language
Cameroon	Yaoundé	19.522.000	$ 41.723.000.000,00	$ 2.152,00	Central African CFA franc	French language, English language
Japan	Tokyo	127.590.000	$ 4.354.000.000.000,00	$ 34.100,00	¥	Japanese language

Google API can use this table as the basis for a lot of different visualisations. Now, some examples of maps and tables are illustrated.

A. Map:

One of the applications of Google API is to quickly create a map. For instance, a map of the first column of table 5.2 gives the result shown in figure 5.12. Figure 5.12 gives a static illustration as an image. However, Google makes it possible to export this map as JavaScript, so an interactive version of this map can be saved as html. This interactive version allows users to zoom in on, click and drag the map.

B. Table:

Another example of a possible visualisation is a table as illustrated in figure 5.13. Again, the table gives a static illustration as an image. Google makes it possible to export this table as JavaScript, so an interactive version of this table can be saved as html. This interactive version allows users to change the graph. For instance, it is possible to group the rows, and to make calculations.

Figure 5.12: Screenshot of the Google map output of example 1

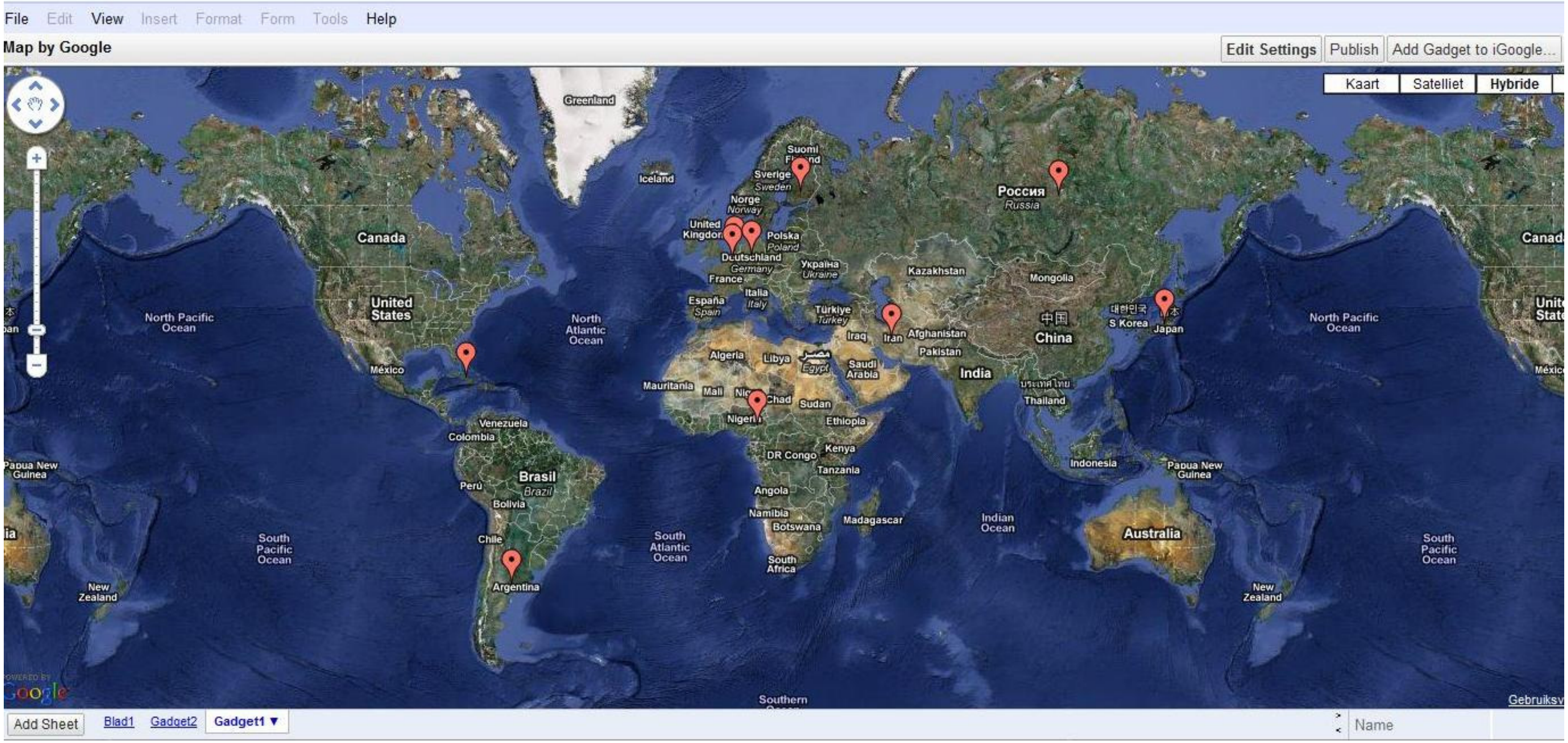
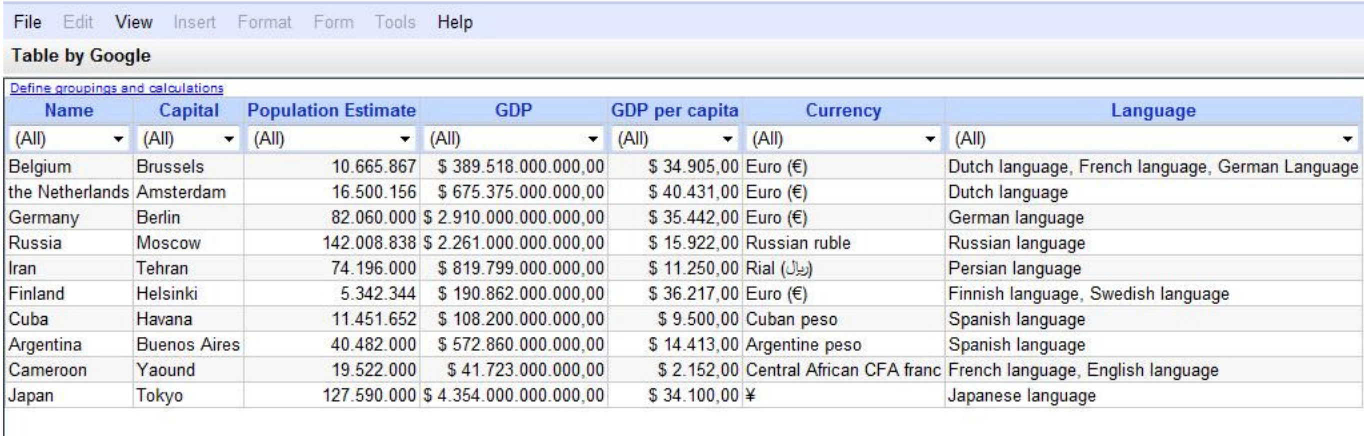


Figure 5.13: Screenshot of the Google table output of example 1



Name	Capital	Population Estimate	GDP	GDP per capita	Currency	Language
(All)	(All)	(All)	(All)	(All)	(All)	(All)
Belgium	Brussels	10.665.867	$ 389.518.000.000,00	$ 34.905,00	Euro (€)	Dutch language, French language, German Language
the Netherlands	Amsterdam	16.500.156	$ 675.375.000.000,00	$ 40.431,00	Euro (€)	Dutch language
Germany	Berlin	82.060.000	$ 2.910.000.000.000,00	$ 35.442,00	Euro (€)	German language
Russia	Moscow	142.008.838	$ 2.261.000.000.000,00	$ 15.922,00	Russian ruble	Russian language
Iran	Tehran	74.196.000	$ 819.799.000.000,00	$ 11.250,00	Rial (ریال)	Persian language
Finland	Helsinki	5.342.344	$ 190.862.000.000,00	$ 36.217,00	Euro (€)	Finnish language, Swedish language
Cuba	Havana	11.451.652	$ 108.200.000.000,00	$ 9.500,00	Cuban peso	Spanish language
Argentina	Buenos Aires	40.482.000	$ 572.860.000.000,00	$ 14.413,00	Argentine peso	Spanish language
Cameroon	Yaound	19.522.000	$ 41.723.000.000,00	$ 2.152,00	Central African CFA franc	French language, English language
Japan	Tokyo	127.590.000	$ 4.354.000.000.000,00	$ 34.100,00	¥	Japanese language

Example 2: Visualisation of information about companies

This example will visualise, among others, SPARQL Query 2 from the previous part (5.2.4). For this example, we have added other companies to the example using the same query.

After slightly adjusting the output from the SPARQL queries (changing the column titles, deleting hyperlinks, ordering the numbers, etc.), are displayed in table 5.3. The parts of the table in red were not available using the same SPARQL query, because of ontological inconsistency, a topic which is discussed in section 3.1 (on page 11). Another problem can be found in the column "Revenue". This column shows the revenue of the different companies the currency of the country of origin. Because of this inconsistency, it is not easy to compare the results with each other.

Table 5.3: Input for visualisation for example 1

Company Name	Industry	Type	Location Headquarters	Revenue	Number of Employees	Homepage
General Motors Company	Automotive	Limited liability company / Government-owned company	Renaissance Center - Michigan - Detroit - United States	148.979.000.000 US Dollar	244,5	<http://www.gm.com>
Saab AB	Aerospace / Defence - military	Public Company	Järfälla - Sweden	23.021.000.000 Swedish Krona	13.757	<http://www.saabgroup.com/en/index.htm>
Spyker Cars N.V.	Automotive	Public Company	Zeewolde	36.293.000 euro	-	<http://www.spykerworld.com/>
Umbro	Textile	-	Cheadle, Greater Manchester - England	-	-	<http://www.umbro.com
Twitter, Inc	Social network service / Micro blogging	Privately held company	California - San Fransisco - United States	400.000 US Dollar	74	<http://twitter.com/>

Again, Google API can use this table as the basis for a lot of different visualisations.

A. Map:

As in the previous example, a map can be created with Google API. In this example we can create a map with the locations of the headquarters of the companies. The result is shown in figure 5.14. Again, figure 5.14 gives a static illustration as an image. Again, Google makes it possible to export this map as JavaScript, so an interactive version of this map can be saved as html. This interactive version allows users to zoom in on, click and drag the map.

B. Table:

Another example of a possible visualisation is a table as in figure 5.15. Again, the table gives a static illustration as an image. Google makes it possible to export this table as JavaScript, so an interactive version of this table can be saved as html. This interactive version allows users to change the graph. For instance, it is possible to group the rows, and to make calculations.

Figure 5.14: Screenshot of the Google map output of example 2

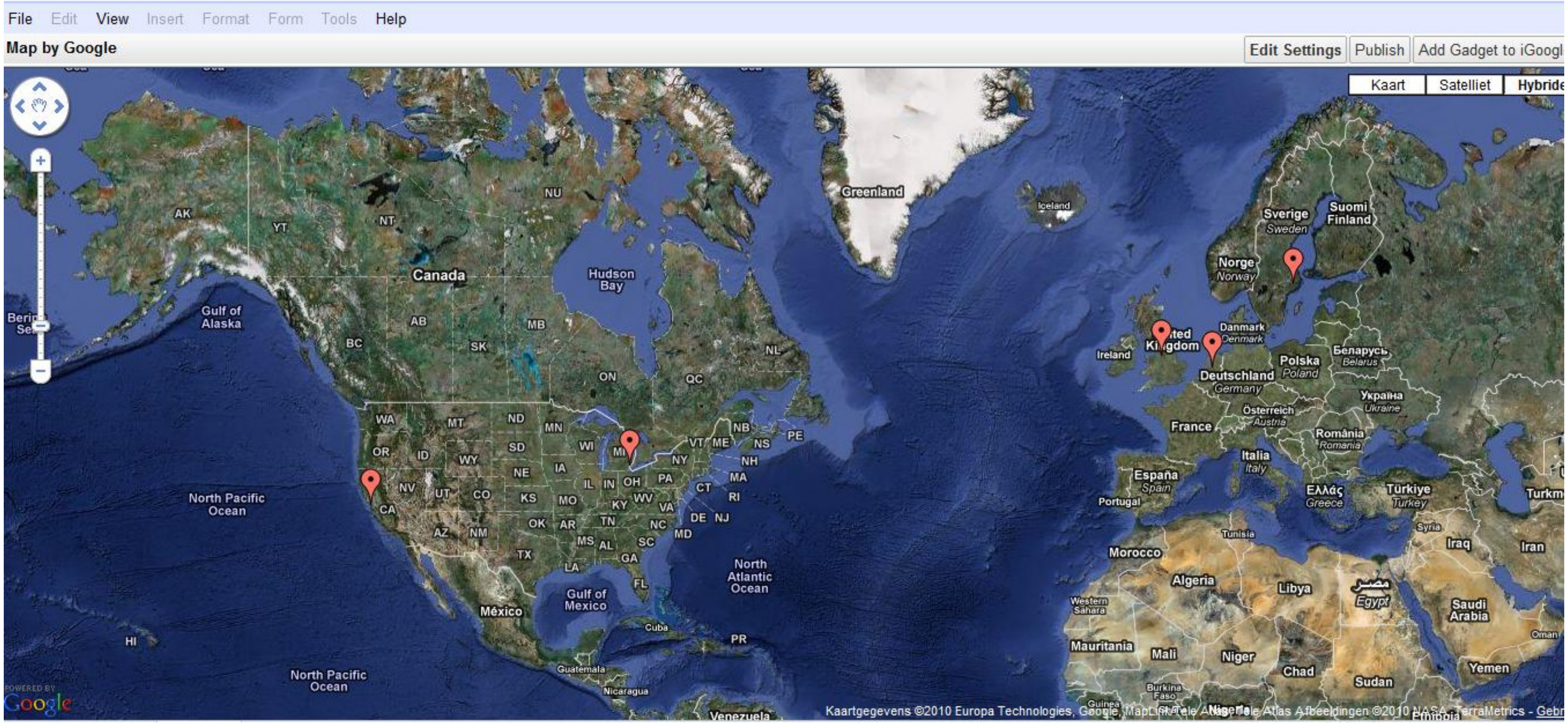
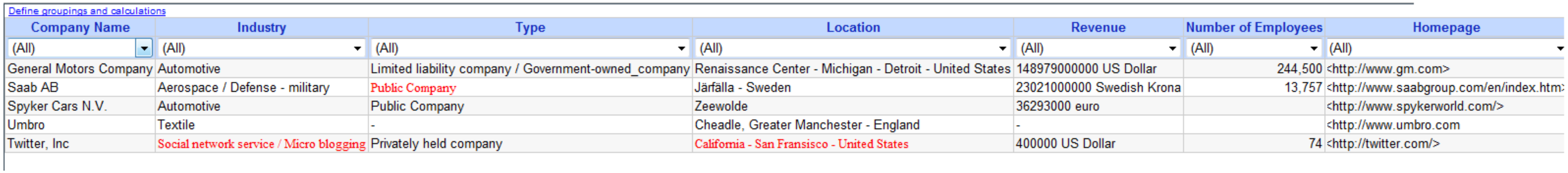


Figure 5.15: Screenshot of the Google table output of example 2



Define groupings and calculations

Company Name	Industry	Type	Location	Revenue	Number of Employees	Homepage
(All)	(All)	(All)	(All)	(All)	(All)	(All)
General Motors Company	Automotive	Limited liability company / Government-owned_company	Renaissance Center - Michigan - Detroit - United States	148979000000 US Dollar	244,500	<http://www.gm.com>
Saab AB	Aerospace / Defense - military	Public Company	Järfälla - Sweden	23021000000 Swedish Krona	13,757	<http://www.saabgroup.com/en/index.htm
Spyker Cars N.V.	Automotive	Public Company	Zeewolde	36293000 euro		<http://www.spykerworld.com/>
Umbro	Textile	-	Cheadle, Greater Manchester - England	-		<http://www.umbro.com
Twitter, Inc	Social network service / Micro blogging	Privately held company	California - San Fransisco - United States	400000 US Dollar	74	<http://twitter.com/>

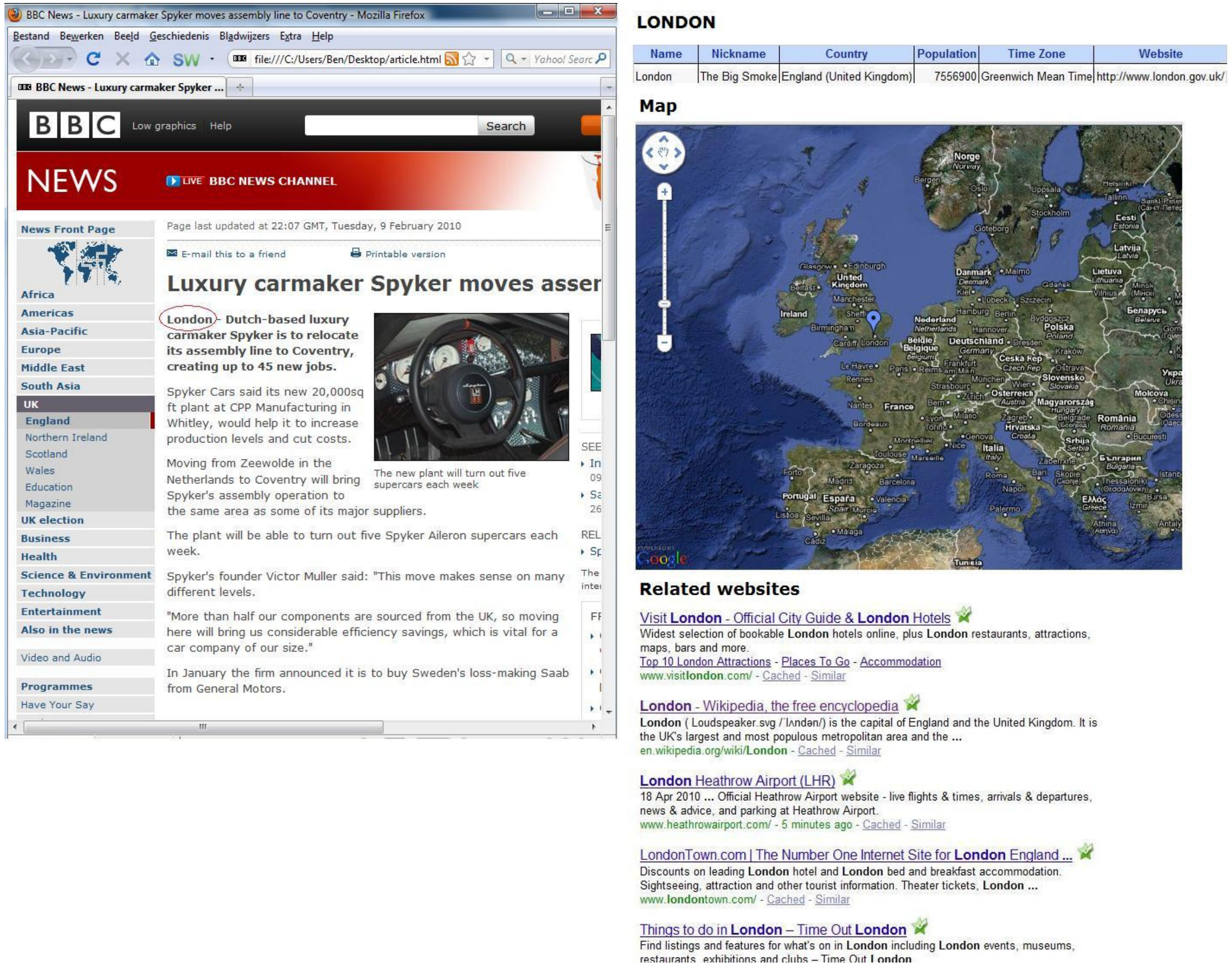
5.3 Example of possible output

In the previous section (5.3), the different parts of a recommender system are explained. In this part of the case study, the result of bringing together these previous parts will be illustrated. The goal of this case study was not to create a fully operational recommender system, but rather to understand the architecture. Because a fully operational recommender system is to complicated to develop, we will now try to demonstrate how this system could look like by providing an example.

Figure 5.16 and figure 5.17 are suggestions of how a recommender system could look like. It will be presumed in this example that these figures are the **graphical user interface** of a simple recommender system. Figure 5.16 is the possible result of a simple recommender system for the city "London" and figure 5.17 is the result for the company "General Motors".

Both figures have the same disposition. The unstructured text, in this case a news article, can be found on the left-hand side of the figures. The results (the recommendations) regarding the labels "London" and "General Motors" are given at the right-hand side of the figures. The recommendations contain three different parts, namely a table, a map and related websites. The table provides more information about the subject. The map in figure X gives the location of the city "London" and the map in figure X shows the location of the General motors' global headquarters. The "related websites" section gives a list of websites that might be of interest. These three parts are merely suggestions. Additional options could be a list of relevant scientific papers, a series of photos, complementary or similar products, etc.

Figure 5.16: Visualisation of a suggestion for a recommender system for “London”



BBC News - Luxury carmaker Spyker moves assembly line to Coventry - Mozilla Firefox

Page last updated at 22:07 GMT, Tuesday, 9 February 2010

E-mail this to a friend

Printable version

Luxury carmaker Spyker moves asser

London - Dutch-based luxury carmaker Spyker is to relocate its assembly line to Coventry, creating up to 45 new jobs.

Spyker Cars said its new 20,000sq ft plant at CPP Manufacturing in Whitley, would help it to increase production levels and cut costs.

The new plant will turn out five supercars each week

Moving from Zeewolde in the Netherlands to Coventry will bring Spyker's assembly operation to the same area as some of its major suppliers.

The plant will be able to turn out five Spyker Aileron supercars each week.

Spyker's founder Victor Muller said: "This move makes sense on many different levels.

"More than half our components are sourced from the UK, so moving here will bring us considerable efficiency savings, which is vital for a car company of our size."

In January the firm announced it is to buy Sweden's loss-making Saab from General Motors.

LONDON

Name	Nickname	Country	Population	Time Zone	Website
London	The Big Smoke	England (United Kingdom)	7556900	Greenwich Mean Time	http://www.london.gov.uk/

Map

Related websites

Visit **London** - Official City Guide & **London** Hotels
Widest selection of bookable **London** hotels online, plus **London** restaurants, attractions, maps, bars and more.
Top 10 London Attractions - Places To Go - Accommodation
www.visit**london**.com/ - Cached - Similar

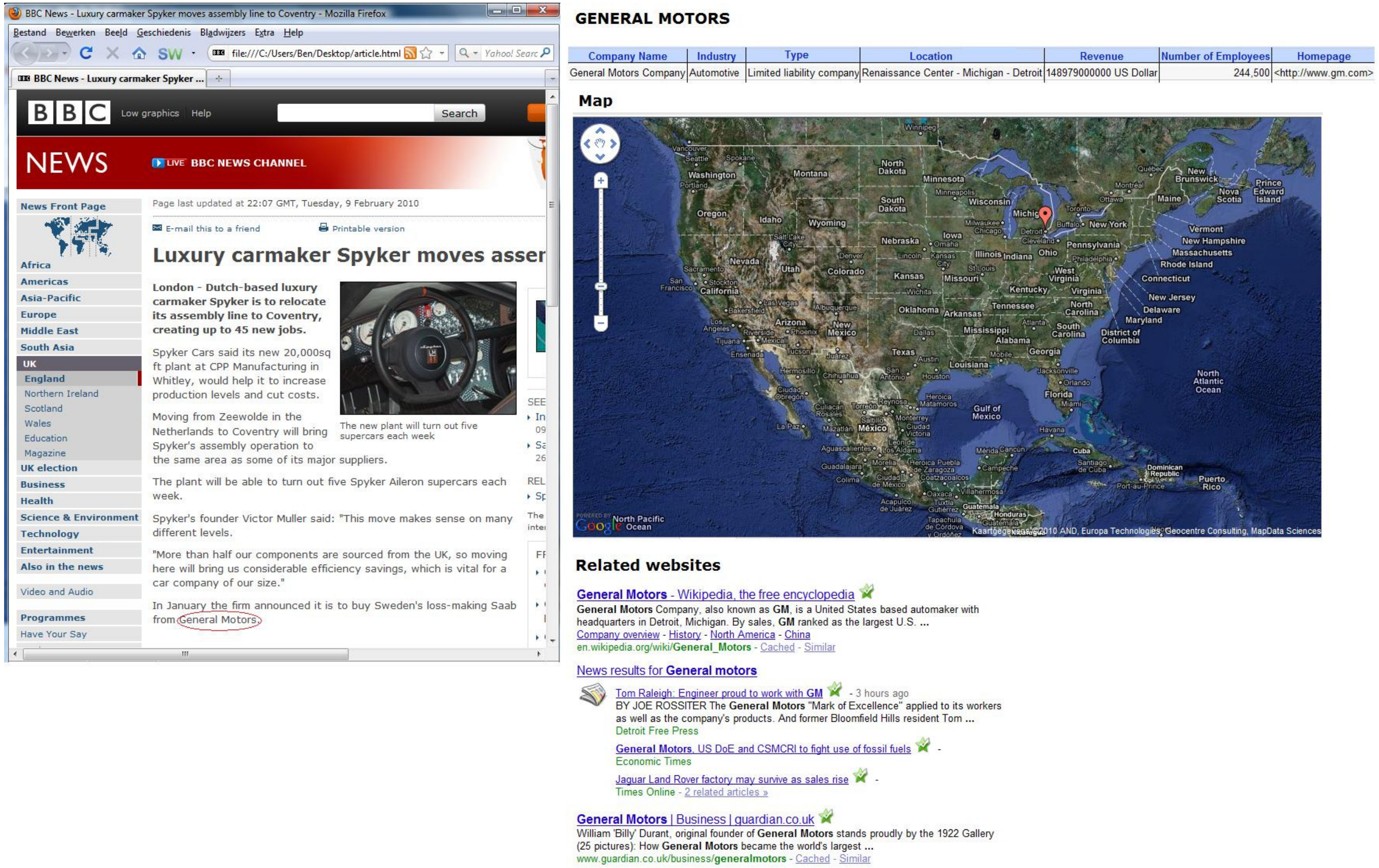
London - Wikipedia, the free encyclopedia
London (Loudspeaker.svg / lʌndən/) is the capital of England and the United Kingdom. It is the UK's largest and most populous metropolitan area and the **...**
en.wikipedia.org/wiki/**London** - Cached - Similar

London Heathrow Airport (LHR)
18 Apr 2010 **...** Official Heathrow Airport website - live flights & times, arrivals & departures, news & advice, and parking at Heathrow Airport.
www.heathrowairport.com/ - 5 minutes ago - Cached - Similar

LondonTown.com | The Number One Internet Site for **London** England **...**
Discounts on leading **London** hotel and **London** bed and breakfast accommodation. Sightseeing, attraction and other tourist information. Theater tickets, **London ...**
www.**london**town.com/ - Cached - Similar

Things to do in **London** – Time Out **London**
Find listings and features for what's on in **London** including **London** events, museums, restaurants, exhibitions and clubs – Time Out **London**

Figure 5.17: Visualisation of a suggestion for a recommender system for “General Motors”



Page last updated at 22:07 GMT, Tuesday, 9 February 2010

Luxury carmaker Spyker moves asser

London - Dutch-based luxury carmaker Spyker is to relocate its assembly line to Coventry, creating up to 45 new jobs.

Spyker Cars said its new 20,000sq ft plant at CPP Manufacturing in Whitley, would help it to increase production levels and cut costs.

The new plant will turn out five supercars each week

Moving from Zeewolde in the Netherlands to Coventry will bring Spyker's assembly operation to the same area as some of its major suppliers.

The plant will be able to turn out five Spyker Aileron supercars each week.

Spyker's founder Victor Muller said: "This move makes sense on many different levels.

"More than half our components are sourced from the UK, so moving here will bring us considerable efficiency savings, which is vital for a car company of our size."

In January the firm announced it is to buy Sweden's loss-making Saab from General Motors

GENERAL MOTORS

Company Name	Industry	Type	Location	Revenue	Number of Employees	Homepage
General Motors Company	Automotive	Limited liability company	Renaissance Center - Michigan - Detroit	148979000000 US Dollar	244,500	<http://www.gm.com>

Map

North Pacific Ocean
Gulf of Mexico
North Atlantic Ocean
Kaartgegevens ©2010 AND, Europa Technologies, Geocentre Consulting, MapData Sciences

Related websites

General Motors - Wikipedia, the free encyclopedia
General Motors Company, also known as GM, is a United States based automaker with headquarters in Detroit, Michigan. By sales, GM ranked as the largest U.S. ...
Company overview - History - North America - China
en.wikipedia.org/wiki/General_Motors - Cached - Similar

News results for General motors

Tom Raleigh: Engineer proud to work with GM - 3 hours ago
BY JOE ROSSITER The General Motors "Mark of Excellence" applied to its workers as well as the company's products. And former Bloomfield Hills resident Tom ...
Detroit Free Press

General Motors, US DoE and CSMCRI to fight use of fossil fuels -
Economic Times

Jaguar Land Rover factory may survive as sales rise -
Times Online - 2 related articles »

General Motors | Business | guardian.co.uk
William 'Billy' Durant, original founder of General Motors stands proudly by the 1922 Gallery (25 pictures): How General Motors became the world's largest ...
www.guardian.co.uk/business/generalmotors - Cached - Similar

5.4 Conclusion research sub-question 4: *How are recommender systems built in detail?*

Chapter 5 conducted a more detailed analyses of some building blocks from chapter 3 in order to get a better insight in them.

A recommender system contains a **text mining tool** that has to goal to analyze unstructured text and uncover labels in them. These labels are words or concepts that are part of a certain domain that will be the basis for providing relevant additional information. The identified labels will be the input for the **Recommender Module**.

An RDF file will contain information about these labels. URIs in this RDF file will provide more information about this label and are the source for other RDF files that contain data that is linked to this URI.

Chapter 5 presents some examples of how the URI links of labels can be found. Examples of RDF syntaxes is presented in the case study to demonstrate how this is used in practice.

When the RDF data is available, this data can be queried by SPARQL via SPARQL endpoints in order for the **Recommender Module** to give suggestions for relevant additional information. The case study provides some examples of simple SPARQL queries.

Chapter 5 also presents some examples of **visualising the results** of SPARQL queries in tables and maps. These tables and maps are examples of how the results from the Recommender Module could be presented to users of the recommender system. It also illustrates an example of a suggestion for a **graphical user interface** that shows how the recommended information can be shown to the user.

Chapter 6: Proposed architecture for the recommender system composed of building blocks

6.1 Introduction

The case study executed in cooperation with iKnow, in combination with the literature review, will now result in a proposition of the architecture for a recommender system. This chapter will also give an answer to research sub-question 5: *How does the architecture of a modern/next generation recommender system looks like?*

6.2 Important technical building blocks of the proposed recommender system

The recommender system proposed here will make use of the Semantic Web. The Semantic Web is not a separate Web, but an extension of the World Wide Web. The Semantic Web gives information a well-defined meaning that enables computers and people to work better in cooperation. It will take advantage of the inference mechanisms involving semantic description developed in the Semantic Web. This will allow the recommender system to discover hidden semantic associations by exploring the knowledge and structure of the ontological model (Blanco-Fernández, et al., 2008). It can use semantic-based similarity measures in order to improve their effectiveness (Drumond & Girardi, 2008). The recommender system will also make use of Linked Data, that is based on the availability of large amount of data in RDF. This will make the recommender system able to create links between data, and by doing so suggesting more relevant information. In the case study performed in this master thesis, Linked Data from the Linking Open Data project (LOD) was used. However, these interlinked datasets trough reuse of common vocabulary are not yet widespread, they are starting to expand.

As already mentioned, the Linking Open Data project utilizes an ontology that is based on reuse and common vocabulary. This makes it easier for recommender systems to interpret data and make good suggestions to its users. The Linking Open Data project aims at making datasets freely available in RDF and create RDF links between them (Raimond, et al., 2007). It aims at bootstrapping the Web of Data by publishing datasets in RDF on the web and creating large numbers of links between these datasets (Hausenblas, 2009).

6.3 Analyzing unstructured text by the Text Mining Module

The creation of input for the proposed recommender system was described in the case study. The first step in creating input for the recommender system will be the analyses of the unstructured text that will be the basis of recommendations by a Text Mining Module.

The Text Mining Module will analyze and filter the unstructured text in order to identify labels (words from specific domains) that can be the basis for further suggestions. This Text Mining Module will identify themes and subjects by analyzing the labels in the unstructured text and pass them trough to the Recommender Module. The examples of labels used in the case study come from two domains, namely geographic locations or companies. For the proposed recommender system, there need to be added numerous other domains. The number of domains can be determined by the developers of the recommender system and depends on the kind of recommendations that need to be made. When a domain is used, it should be based on a proper and complete ontology.

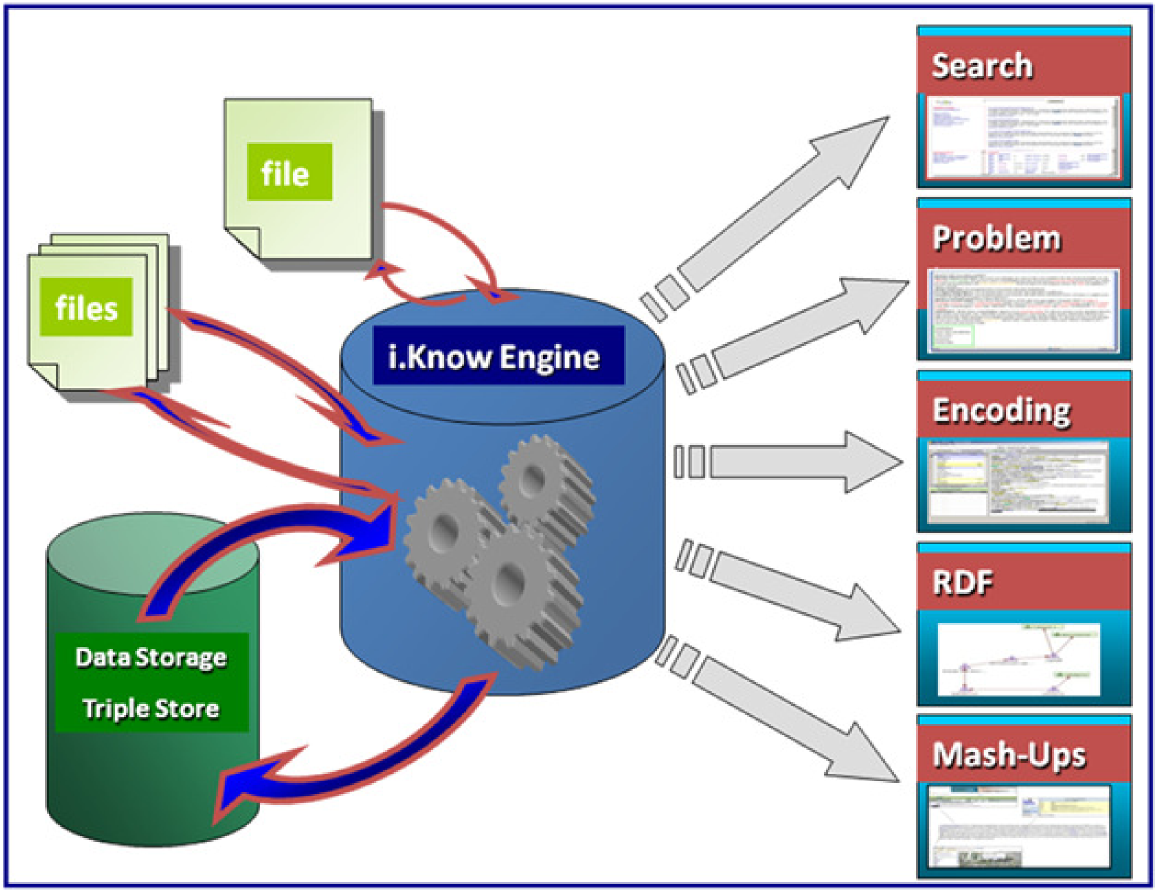
Analogous with this filtering, a concept cluster needs to be created in order to link similar labels to each other. The concept cluster in this case study (a csv file) was made by iKnow with their specialised software. This software analyzes unstructured text (in the case study: news articles). After the analysis, the software creates an output file that contains a list of concepts. The software is designed to recognize related words: clusters.

When the labels and the similar labels are identified, they need to be linked with URIs to the RDF files that contain information about them. These RDF files also contain URI links to other data that is linked with these concerning labels. A Uniform Resource Identifier (URI) is a string of characters used to identify a name or a resource on the Internet.

After the unstructured text is analyzed, the engine of the recommender system will create an RDF file that will be the input for the Recommender Module. This RDF file contains information that will be the basis for suggestions that the proposed recommender system will provide.

To visualise the previous paragraph, figure 6.1 is given. This figure shows the input for the engine as a separate data and (RDF) triple storage, together with various files. The (i.Know) engine forms the centre of this figure. When the processes in the engine are complete, the output is delivered. In this figure, a lot of possible outputs are illustrated. For the proposed recommender system in this master thesis, the output of the engine will be an RDF file.

Figure 6.1: Visualisation of the input of the engine, the engine and the output of the engine for the proposed recommender system



Source: iKnow

6.4 The Recommender Module

The goal of the Recommender Module is to provide information stored in the different data sets to the users. The action of this module starts when it receives a concept (label) from the Text Mining Module. After this, it searches the different data sets for items classified in the same concept. Each time the Text Mining Module identifies a concept in a message, it sends this concept to the Recommender Module that searches the data sets for more items to recommend.

6.4.1 Querying datasets

When a recommender system makes suggestions, the Recommender Module has to query datasets in order to obtain results. In the recommender system that is proposed, querying the RDF files will be done with the SPARQL query language at SPARQL endpoints.

However, there might be some problems when querying via SPARQL endpoints. Much Semantic Web data lives inside triple stores and can only be accessed by querying a SPARQL endpoint with SPARQL queries. It can be difficult to connect information in these stores with other external data sources. The problem is that the URIs in a lot of these datasets are not dereferenceable as mentioned in section 3.6 (see infra, p.20). Dereferenceable means that HTTP URIs are used, so that people can look them up. A possible solution is provided by *Pubby* (http://www4.wiwiss.fu-berlin.de/pubby/). Pubby is a publishing tool that can be used to add Linked Data interfaces to SPARQL. When setting up a Pubby server for a SPARQL endpoint, these not deferenceable URIs will be translated into dereferenceable URIs. Pubby will handle requests by connecting to the SPARQL endpoint, asking it for information about the original URI, and pass the results back to the client. That is why the SPARQL endpoints used in this proposed recommender system should be set up with a Pubby or a similar server.

6.4.2 Presenting the results from the Recommender Module

The results of the queries that are created by the Recommender Module can be very diverse and in different formats. Everything depends on the type of recommendation that need to be made by the recommender system. For example, a recommender system for scientific researchers can suggest related PDF papers or presentations to the topic he is reading about. When a journalist is writing an article, a recommender system can suggest information about the topic he is writing about. This can be information like the information that was provided in the case study. For example, if the journalist is writing an article that includes a company name, the recommender system can filter this label from the text and suggest related information in real-time about this company like for example existing articles about the company, the number of employees, revenue, location of the headquarters or website.

Keep in mind that the recommendations are made by comparing the output from the Text Mining Module against an ontology. The quality of the recommendations are depending directly on the quality of the ontologies and the text mining tools used. This is as well a reason why experts should be responsible for updating the ontology.

The result of the recommendation system can also be personalized by equipping it with a profile of the user. This profile can contain administrative information about the user like name, institution, job description, as well as his interests areas. This can make the recommender system better capable to provide relevant suggestions.

An evaluation system by users can also be built into the recommender system to make itself able to improve after time. If users are able to criticize the suggestions made by the Recommender Module with "good" or "bad", the suggestions could be adapted in the future.

A possible subject of discussion could be in what manner the recommendations should be made. Should they be given in a separate frame to avoid interruption, or should they be given in the same frame to draw the attention of the user.

It is also important to limit the number of suggestion that will be made by the recommender system. When a too large list of recommendations is provided, the problem of information overload will still be present.

Chapter 7: Conclusions

This chapter will draw the conclusions of this master thesis, will give answers to the central research question and the research sub-questions and will give recommendations for further research.

A **recommender system** is a software system whose main goal is to aid in the social collaborative process of indicating or receiving indication, when the number of options is enormous. They supply users with information in order to help them make decisions or to solve problems, without users have to search for it themselves.

Knowledge management has become a very important subject. Knowledge management is all about getting the right knowledge, in the right place, at the right time. It is based on the idea that an organisation's most valuable resource is the knowledge of their people. Raw information may be widely available to a lot of organisations, but only some organisations will be able to convert the information into relevant knowledge and to use this knowledge to achieve their goals.
Information and communication technology has made information abundant. Because of the Internet you can basically get any information you might desire in seconds. However, the reliability, usefulness, and timeliness of the information is difficult to verify. There is an **information overload** due to an existence of excess low quality information that can lead to inefficient decision making.
To help organisations in their knowledge management process and to overcome the problem of information overload, recommender systems can be used. Recommender systems will lead to **more efficient knowledge acquisition**. They can lead to a reduction of time in this very time-consuming process and this can result in a **decrease in costs** for enterprises for which knowledge acquisition is important.

Modern recommender systems exist of several technical building blocks. The proposal for the recommender system in this master thesis will contain the building blocks described next. A first building block, that is also used in many other recommender systems, is an **ontology**. An ontology is a representation of a vocabulary, often specialized to some domain or subject matter. An ontology characterizes the semantics in terms of concepts and their relationships, represented by classes and properties, respectively. In the context of a recommender system, instances of classes represent the available items and their attributes, whereas instances of properties link the items and attributes to each other.

Modern recommender system are based on **RDF** for storing data**.** The resource description framework (RDF) is a W3C standard format for storing arbitrary data on the Web and elsewhere. In other words, it is a framework for representing information in the Web. RDF provides a machine-readable way to say anything about anything.

SPARQL is another building block of modern recommender systems, used for querying the RDF data sets. SPARQL, an acronym standing for SPARQL Protocol and RDF Query Language, is a general term for both a protocol and a query language. SPARQL can be utilized to express queries across RDF data.

Another important building block from which a modern recommender system can make use of is the **Semantic Web**. The Semantic Web is not a separate Web, but an extension of the World Wide Web. The Semantic Web gives information a well-defined meaning that enables computers and people to work better in cooperation. It will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can carry out sophisticated tasks for users.

A fully functional Semantic Web is based on the availability of large amounts of data as RDF that is interlinked as a web of data. **Linked Data** refers to a set of best practices for publishing and connecting structured data on the Web in order to achieve this interlinking of data. There are projects, like the Linking Open Data (LOD) project, that aim at make huge amounts of data freely available by publishing datasets in RDF and create RDF links between them. Linked Data from the LOD project is at the **foundation** of the proposed recommender system in this master thesis. **The use of Linked Data from the LOD project distinguishes the proposed recommender system from already existing recommender systems**.

The proposed recommender system consists of a **Text Mining Module** and a **Recommender Module.** The Text Mining Module will analyze and filter the unstructured text in order to identify **labels or concepts** that can be the basis for further suggestions. These labels are words or concepts that are part of a certain domain that will be the basis for providing relevant additional information. The identified labels will be the input for the Recommender Module. The examples of labels used in the case study came from two domains, namely geographic locations or companies. For the proposed recommender system, there need to be added numerous other domains. The number of domains can be determined by the developers of the recommender system and depend on the kind of recommendations that need to be made. When a domain is used, it should be based on a suitable and complete ontology.

After the unstructured text is analyzed, the engine of the recommender system will create an **RDF file** that will be the input for the Recommender Module. This RDF file contains information that will be the basis for suggestions that the proposed recommender system will provide.

The RDF file will contain **information about the labels in URIs** that will provide more information about these labels and are the source for other RDF files **that contain data that is linked to this URI**.

When a recommender system makes suggestions, it has to **query datasets** in order to obtain results. In the recommender system that is proposed, querying of the RDF files will be done with the **SPARQL query language** at SPARQL endpoints.

When the RDF data from labels is available, this data can be queried by SPARQL via SPARQL endpoints in order for the Recommender Module to give suggestions for relevant additional information.

When the corresponding recommendations have been found by the Recommender Module, they need to be **visualised** for the user. The results of the queries that are created by the Recommender Module can be very diverse and in different formats. Everything depends on the type of recommendation that needs to be made by the recommender system. The case study presented an example of visualising the results of the SPARQL queries in tables and maps. It also illustrates an example of a suggestion for a **graphical user interface** that shows how the recommended information can be shown to the user.

We can conclude that a recommender system that is based on the Semantic Web and Linked Data has **a lot of potential**. The results of recommended information can be considerably improved by using these two building blocks. However, there are **some problems** with the use of Linked Data. The interlinked datasets through reuse of common vocabulary and shared URIs, that are the basis for Linked Data, are starting to expand, but they are not yet widespread. The data from the Open Linked Data movement needs to be sustainable, licensed, reliable and able to trace the provenance. Besides these problems, there is also the difficulty of URI disambiguation. Linking data has been widely encouraged, but there has been little analysis of the accuracy of the links or the datasets themselves. This is because datasets are often converted from existing sources that can themselves be incomplete or inaccurate. This can also be due to incorrect information that is publicised by members of the community. When linking these inconsistencies, a snowball effect of incomplete or inaccurate data is produced as more datasets are added. This problem, due to the lack of resources that leads to insufficient time to rigorously check the input for correctness or completeness, can be found in many digital repositories. This lack of resources can be explained by the free availability and community based approach of the Linked Data initiative. There need to be made investments in measuring and guaranteeing of the correctness of this information.

There are some **recommendations for further research** that I would like to give. First of all, there needs to be done more research about the accuracy, reliability, sustainability and licensing of Linked Data. This research is necessary to make sure that the proposed recommender system in this master thesis has a solid basis. Another problem that has come up in this master thesis is the technical capability of providing real-time suggestions. This problem also needs additional examination. Furthermore, the recommender system proposed in this master thesis, that is based on the Semantic Web and Linked Data, could be developed in order to experiment with the possibilities and limitations of this software system.

References

Adomavicius, G. & Tuzhilin, A., 2005, Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering, no. 17 (6),* pp. 739–749

Berners-Lee, T., Chen, Y., Chilton, L., Connoly, D., Dhanaraj, R., Hollenbach, J., Lerer, A. & Sheets, D., 2006, Tabulator: Exploring and Analyzing linked data on the Semantic Web

Berners-Lee, T., Hendler, J. & Lasilla, O., 2001, The Semantic Web, *Scientific American, May 2001*, pp. 29-37

Bizer, C., Heath, T. & Berners-Lee, T., 2009, Linked Data – The story so far, *International Journal on Semantic Web and Information Systems (IJSWIS), special issue*

Blanco-Fernández, Y., Díaz-Redondo, R., Fernández-Vilas, A., García-Duque, J., Gil-Solla, A., López-Nores, M., Pazos-Arias, J. & Ramos-Cabrer, M., 2008, Exploiting synergies between semantic reasoning and personalization strategies in intelligent recommender systems: A case study, *The Journal of Systems and Software, no. 81,* pp. 2371–2385

Bizer, C., Heath, T., Ayers, D. & Raimond, Y., 2007, Interlinking Open Data on the Web, *Demonstrations Track, 4th European Semantic Web Conference (ESWC2007), Innsbruck, Austria*

Chaffey, D., 2009, *E-Business and E-Commerce management*, Harlow: Pearson Education Limidted

Chandrasekaran, B., Josephson, J. & Benjamins, R., 1999, What Are Ontologies, and Why Do We Need Them?*, IEEE Intelligent Systems, no. 14,* pp. 20-26

Davenport, T. & Beck, J. 2000, The Attention Economy, *Boston: Harvard Business School Press*, pp. 1–16

Doucet, B., 2009, i.Know interpreteert informatie semantisch, *Data news, no.29*

Drumond, L. & Girardi, R., 2008, A multi-agent legal recommender system, *Artif Intell Law, no. 16,*, pp. 175-207

Ducharme, B., 2005, RDF: The Resource Description Framework

Hanani, U., Shapira, B., & Shoval, P., 2001, Information filtering: Overview of issues, research and systems, *User Modeling and User-Adapted Interaction, no. 11,* pp. 203–259

Halb, W., Raimond, Y. & Hausenblas, M., 2008, Building Linked Data For Both Humans and Machines, *WWW 2008 Workshop: Linked Data on the Web (LDOW2008), Beijing, China*

Göksedef, M. & Gündüz-Ög˘üdücü, S., 2010, Combination of Web page recommender systems, *Expert Systems with Applications, no. 37,* pp. 2911–2922

Guarino, N., 1998, Formal Ontology in Information Systems*, Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press,* pp. 3-15.

Hasanali, F., 2002, Critical Success Factors of Knowledge Management

Hausenblas, M., 2009, Exploiting Linked Data For Building Web Applications

Heylighen, F., 2005a, Complexity and Information Overload in Society: why increasing efficiency leads to decreasing control, *Technological Forecasting and Social Change*

Heylighen, F., 2005b, Tackling Complexity and Information Overload : intelligence amplification, attention economy and the global brain, *Technological Forecasting and Social Change*

Hovland, I., 2003, Knowledge Management and Organisational Learning: An International Development Perspective: An Annotated Bibliography

Jaffri, A., Glaser, H. & Millard, I., 2008, URI Disambiguation in the Context of Linked Data, *Linked Data on the Web (LDOW2008), Beijing, China*

Laud, R. & Schepers, D., 2009, Beyond Transparency: Information Overload and a Model for Intelligibility, *Business and Society Review, Vol. 114 Issue: no. 3,* pp. 365-391

Malhotra, Y., 2000, From Information Management to Knowledge Management: Beyond the 'Hi-Tech. Hidebound' Systems, *Knowledge Management for the Information Professiona, Medford, N.J.: Information Today Inc.,* pp. 37-61

Medo, M., Zhang, Y. & Zhou, T., Adaptive model for recommendation of news, *Europhysics Letters, no. 88,* pp. 38005

Monroe, D., 2009, Just For You, *Communications of the ACM, volume 52, no. 8*, pp. 15-17

Montaner, M., López, B. & de La Rosa, J., 2003, A taxonomy of recommender agents on the Internet, *Artificial Intelligence Review, no. 19,* pp. 285–330

NHS (National Library for Health), 2005, ABC of Knowledge Management

Noy, N. & McGuinness, D., 2001, Ontology Development 101: A Guide to Creating Your First Ontology

Oren, E., Belbru, R., Carasta, M., Cygniak, R., Stenzhorn, H. & Tummarello, G., 2008, Sindice.com: a document-oriented lookup index for open linked data, *International Journal of Metadata, Semantics and Ontologies 2008 - Vol. 3, No. 1,* pp. 37 – 52

Paredes, T., 2003, Blinded by the light: information overload and its consequences for securities regulation

Porcel, C., López-Herrera, A. & Herrera-Viedma, E., 2008, A recommender system for research resources based on fuzzy linguistic modelling, *Expert Systems with Applications, no. 36,* pp. 5173–5183

Prud'hommeaux, E. & Seaborne, A., 2008, SPARQL Query Language for RDF, requested via http://www.w3.org/TR/rdf-sparql-query/

Raimond, Y., Abdallah, S., Sandler, M. & Giasson, F., 2007, The music ontology, *Proceedings of the International Conference on Music Information Retrieval*, pp. 417–422

Research Matters (RM), 2008, Knowledge translation toolkit: A resource for researchers

Resnick, P. & Varian, H.,1997, Recommender systems, *Communications of the ACM, volume. 40, no.3,* pp. 56-58

Schafer, J., Konstan, J. & Riedl, J.,2001, E-Commerce Recommendation Applications, *Data Mining and Knowledge Discovery, volume 5, Issue 1-2,* pp. 115-153

Shabir, N. & Clarke, C., 2009, Using Linked Data as a basis for a Learning Resource Recommendation System. *1st International Workshop on Semantic Web Applications for Learning and Teaching Support in Higher Education (SemHE'09)*, *ECTEL'09, Nice, France*

Shadbolt, N., Hall, W. & Berners-Lee, T., 2006, The Semantic Web Revisited, *IEEE Intelligent Systems Journal, volume 21, no.3,* pp. 96-101

Staab, S. & Studer, R., 2004, *Handbook on ontologies,* Berlin Heidelberg*,* Springer – Verlag

Uschold, M. & Gruninger, M., 1996, Ontologies: Principles, Methods and Applications, *The Knowledge Engineering Review, no. 11,* pp. 93-136

X, 2010, Recommendation of Complementary Material during Chat discussions, *Knowledge Management & E-Learning: An International Journal (KM&EL)*

Zhang, X., Edwards, J. & Harding, J., 2007, Personalised online sales using web usage data mining, *Computers in Industry, no. 58,* pp. 772–782

Zhen, L., Huang, G. & Jiang, Z., 2010, An inner-enterprise knowledge recommender system, *Expert Systems with Applications, no. 37,* pp. 1703–1712

Zurawski, M., Smaill, A. & Robertson D., 2008, Bounded Ontological Consistency for Scalable Dynamic Knowledge Infrastructures; paper from *The Semantic Web*, pp. 212-226*,* Berlin / Heidelberg: Springer

Websites

http://linkeddata.org/, Linked Data - Connect Distributed Data across the Web [online], 2010

http://semanticweb.org/wiki/Main_Page, Semantic Web [online], 2009

http://w3c.org/RDF/, Resource Description Framework (RDF) [online], 2010

http://www4.wiwiss.fu-berlin.de/pubby/, Pubby; A Linked Data Frontend for SPARQL Endpoints [online], 2009

http://www.iknow.be, iKnow [online], 2010

http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, Resource Description Framework (RDF): Concepts and Abstract Syntax [online], 2004

http://www.w3.org/TR/owl-features/, OWL Web Ontology Language
Overview [online], 2004

http://www.w3.org/TR/owl-guide/, OWL Web Ontology Language
Guide [online], 2004

http://www.w3.org/TR/rdf-concepts/, Resource Description Framework (RDF): Concepts and Abstract Syntax [online], 2004

http://www.w3.org/TR/REC-xml-names/, Namespaces in XML 1.0 (Third Edition) [online], 2009

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Recommender Systems. Case iKnow: Composition of the required building blocks for the development of recommender systems

Richting: **Master of Management**
Jaar: **2010**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt
behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -,
vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten
verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Plessers, Ben

Datum: **13/06/2010**