Organizational knowledge access: an approach based on ontology networks

Ajmi, Oumaima

This publication URL: https://archive-ouverte.unige.ch/unige:117088
Publication DOI: 10.13097/archive-ouverte/unige:117088

Organizational knowledge access: an approach based on ontology networks

THESIS

Presented to the Faculty of Economics and Management of the University of Geneva

by

Oumaima AJMI

Under the direction of

Prof. Gilles Falquet
Dr. Claudine Métral

To obtain the title of

Docteur ès Économie et Management
mention Systèmes d'Information

Jury members:

Prof. Dimitri Konstantas, University of Geneva, President of the jury
Dr. Thibault Estier, MER, University of Lausanne
Dr. Lamia Friha, IT Project Manager, STIC, University of Geneva.

Thèse no° 60
Genève, 29 août 2018

La Faculté d'Economie et de Management, sur préavis du jury, a autorisé l'impression de la présente thèse, sans entendre, par-là, émettre aucune opinion sur les propositions quI s'y trouvent énoncées et qui n'engagent que la responsabilité de leur auteur.

Genève, le 29 aout 2018

Le Doyen
Prof. Marcelo OLARREAGA

Impression d'après le manuscrit de l'auteur

Abstract

Knowledge management has increasingly become a big challenge, especially in large organizations (Averweg, 2009) where knowledge resides in many different forms. This is particularly true of explicit knowledge captured in documents, databases, and websites. The known vision of the Semantic Web is to offer intelligent services made possible by facilitative machine understanding of web content, especially by using ontologies. Additionally, an ontology can be used to represent knowledge and define common vocabularies of specific organizations. As ontology creation is known to be a time-consuming and challenging task, we propose a hybrid knowledge management ontology-based approach, which stems from an ontologies-network model capable of representing knowledge from structural and syntactically heterogeneous resources, and of building an organizational knowledge graph based on a network of ontologies. Our approach is composed of a Profile Ontology that defines the shared vocabulary of a specific organization and a set of resource representation ontologies created automatically via a knowledge extraction process that deals with structured, semi-structured and unstructured resource formats. We defined an algorithm to intra-connect the resource representations and to connect these representations to the Profile Ontology. A keyword search process is described which results in a list of relevant resources and the exact position of the items searched. To evaluate our approach and our model, we implemented a framework composed of the different processes using the AML-matcher alignment framework. We applied our system to a specific organization by defining a particular Profile Ontology and building a network of ontologies created from its specific resources. Finally, we present the different points that require improvement and which could enhance the performance of the system if sophisticated extraction methods or additional ones were to be used.

Résumé

La gestion des connaissances est un grand défi, en particulier dans les grandes organisations (Averweg, 2009) où les connaissances se présentent sous de nombreuses formes différentes, en particulier les connaissances explicites représentées par exemple dans les documents, les bases de données et les sites Web. La vision connue du Web sémantique est d'offrir des services intelligents en facilitant la compréhension de contenus Web, en particulier en utilisant des ontologies. Néanmoins, une ontologie peut également être utilisée pour représenter des connaissances et définir des vocabulaires communs à des organisations spécifiques. Etant donné que la création d'ontologies est une tâche fastidieuse et exigeante, nous proposons une approche basée sur un modèle d'un réseaux d'ontologies capable d'extraire des connaissances de ressources hétérogènes et de construire un graphe de connaissances organisationnelles. Mon approche est composée d'une ontologie Profile définissant le vocabulaire partagé d'une organisation spécifique et d'un ensemble d'ontologies représentant les ressources de l'organisation créées automatiquement via un processus d'extraction de connaissances. Cette dernière traite des formats de ressources structurés, semi-structurés et non structurés. J'ai défini des algorithmes pour interconnecter la représentation de la ressource et pour les connecter à l'ontologie Profile.

Un processus de recherche par mot-clé est décrit, ce qui donne une liste des ressources pertinentes et le rôle exact des éléments recherchés dans la ressource d'origine. Pour évaluer notre approche et notre modèle, nous avons implémenté un prototype des différents processus utilisant le système d'alignement AML-matcher. J'ai appliqué mon système à une organisation spécifique qui est la GSEM en définissant une ontologie Profile pour cette faculté et en créant un réseau d'ontologies à partir de ses ressources explicites. Enfin, j'ai présenté les différents points d'amélioration et les différents cas d'application de mon travail.

Acknowledgment

Firstly, we would like to express our sincere gratitude to our research supervisors Prof. Gilles Falquet and Dr. Claudine Métral for their patient guidance, enthusiastic encouragement, immense knowledge and useful critiques of this research work. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisors and mentors for our Ph.D. study.

I would also like to thank our thesis committee: Prof. DimitrI Konstantas, Dr. Thibault Estier, and Dr. Lamia Friha, for their insightful comments and encouragement, but also for the hard questions, which incensed me to widen our research from various perspectives.

My sincere thanks also go to the Geneva School of Economics and Management (GSEM) who provided me an opportunity to join their team, and who gave me access to research facilities. In addition, we thank all our colleagues in GSEM for their support.

Finally yet importantly, we would like to thank our family: our parents, our brother, our sisters, and our husband for supporting me spiritually throughout writing this thesis and our life in general.

To Haroun..

Table of Contents

List of Figures

List of Algorithms

List of Tables

Chapitre 1. Introduction

1.1 Scientific context and research problem

The widespread use of computing technologies has led organizations to store their data and knowledge in databases, documents, spreadsheets, etc., which has led to a large and increasing pool of heterogeneous resources. Consequently, it has become more and more difficult, even humanly unfeasible, to have a complete overview of what the organization "knows."

In the knowledge engineering field and more precisely in the knowledge representation field, scientists focus on designing computer representations that capture information about a particular domain. These representations can be used to solve complex problems by interpreting that knowledge, drawing inferences, and asserting new knowledge. This can be used as a solution for representing organizational knowledge in a way that allows it to be captured and harmonized in an automated system.

In this context, the general problem this research addresses, is how to represent the knowledge embedded in explicit organizational resources, how to manage it, and how to access it in answering the key questions:

- What knowledge do an organization has?
- Where can we find this knowledge?
- How do we access the knowledge? (Which vocabulary should we use? Which conversions should we perform?)

In this thesis, we will confine ourselves to explicit knowledge within the same organization that is represented formally or informally in various resources, namely databases, websites, electronic documents, spreadsheets, glossaries, terminologies. This is contrasted with the tacit knowledge individuals have, such as know-how, judgment, etc.

The approach we propose for the study is one of creating an entirely semantic *organizational memory*, which is ontology-based. We construct our approach using a model of a network of ontologies. The network is based on ontologies created by representing meta-data, structure, and content in structured, unstructured, and semi-structured knowledge sources. To construct the network of ontologies, we define a *Profile Ontology* mapped with resource representation ontologies using an algorithm that we established. To access the knowledge captured in the network of ontologies, we take a semantic keyword approach that re-uses the alignment across the network to answer users' queries.

In this work we discuss two main subjects: on the one hand, the existing models, tools, and methods specific to managing knowledge within an organization, and on the other hand, the different semantic and non-semantic approaches intended to extract knowledge from the resources, as well as approaches to best link the knowledge obtained in various locations, and access to it.

1.2 Research area

Proposing a solution to access the heterogeneous pool of knowledge of a specific domain, requires the specification of several aspects: The particular field's knowledge has to be defined and represented. The strategy behind the construction of links between the various pieces of knowledge, has to be defined. The approach used to access the knowledge has to be defined. To build a semantic knowledge schema and to define a strategy to access it, we identify three central questions:

1. Is there a model to represent explicit knowledge resource contents?

Explicit knowledge can be expressed in different ways and stored in different resource formats. The format can be structured as a database, semi-structured as an XML file, or unstructured as a text file. As no model can express all the content of knowledge resources, many approaches and tools have been designed for a specific representation of the resources, by modeling particular parts of the knowledge resources.

2. How can we link explicit knowledge resources' content?

Knowledge resources' content is expressed with employees' vocabularies, and structured in different ways, even in the same knowledge resources. Therefore, we have specified the following two guidelines:

First, if we need to unify the vocabulary within the conceptual schema, how should we do that? We need to determine if we preserve the semantics of the knowledge resources, or if we need to represent content with different vocabularies.

Second, we need to determine the ideal solution for linking, reusing, and concluding new knowledge forms that have already been extracted.

3. How can we best gain access to knowledge represented as a conceptual schema?

Querying languages have been developed to access data and information within ontologies. In our case, ontologies have different structures and are expressed in different vocabularies. Many approaches have been presented to obtain distributed and Peer-to-Peer (P2P) ontologies, but querying a network of ontologies already related via different mapping methods have not been addressed before.

1.3 Proposed research methodology

The research questions mainly address two aspects, namely (i) organizational knowledge and how we can access and integrate it, and (ii) technologies of the semantic web and how they can benefit this integration. To answer these questions, we decided on the following steps:

1. As the goal is to represent knowledge formally, we investigate the nature of the organizational knowledge, determining types, resources, and how it is managed. We study different structures of explicit knowledge resources. We identify the content that we could integrate automatically.
2. Before identifying the resource types, we investigate the knowledge management systems, their different approaches, and their objectives.

3. We define a model, which is able to represent ontology-based knowledge management approaches. The model allows modeling any resource type in any ontology representation language (e.g. OWL, RDFS.) and manages organizational resources as a network of ontologies.
4. Based on ONM model (see section 1.4), we present our ontologies' network-based approach to represent and integrate the organizational knowledge which is based on the metadata, content, and structure of the explicit organizational knowledge. Our approach is based on a network of ontologies, and represents a resource based on its metadata and its content. It constructs a conceptual schema of the organizational knowledge based on the results of alignment between the resource representation ontologies.
5. After defining our approach for constructing the network of ontologies, we investigate how best we can access the network's content. We study similar strategies based on distributed ontologies, and we finally adopt a solution that re-uses alignment results already used to link the knowledge resources' content.
6. As a concrete example of the usability of our strategy, we implemented a prototype system based on our approach. The prototype contains three modules reflecting the approach. The content representation module illustratively presents methods we implemented to exercise transformation of structured (database), semi-structured (XML, HTML, web services) and unstructured (spreadsheets) resources formats. Any other format can easily be added to the prototype. Then we used a matching tool to construct our network of ontologies based on our algorithm for creating the network links. Finally, we developed a semantic search engine by aligning the query and the Profile Ontology to return a ranked list of documents related to a Natural Language query.
7. We evaluate our prototype considering a real world example, and creating a specific domain ontology to specify the organizational aspect of the GSEM, and we experiment with the different processes of our approach.
8. We compare our results to different approaches, and we analyze the performances of our system to detect possible points to be improved.

1.4 Contributions

Enterprise management systems, information retrieval domains, and knowledge engineering are large research fields with a growing number of projects interested in innovative approaches. Working on the identified research problems led to two major contributions:

1. Ontology Network Model (ONM): A generic model to represent ontology-based knowledge networks. ONM contains a Knowledge Extraction Model (KEM) to represent any resource format and structure.
2. A hybrid ontology-based knowledge representation approach based on ONM.

    a. Profile Ontology: A specific organizational ontology to define a common vocabulary for a specific organization. We defined how to create the PO and how to use it for simplifying the resource ontologies integration and access.
    b. The definition of specific algorithms to build and access knowledge represented via our model
3. We implemented the ONM by creating a JAVA framework. The framework can be instantiated for any specific organization and for any ontology-based knowledge management approach, which presents a concrete example of our work realizability. The source code is available and can be an excellent starting point for any improvement.

1.5 Thesis plan

In chapter 2, we introduce the background to the two major concepts of this thesis, namely organizational knowledge, and semantic web technologies.

In part one of chapter 3, we explain the nature of the organizational knowledge, type of resources and how they can be managed. Thus, we investigate different structures of explicit knowledge resources and their content. We examine tools, methods, and algorithms capable of extracting knowledge from different types of resources in a way that enables representing them in a "standard" way. In the part two of chapter 3, we investigate existing approaches, which allow managing the knowledge extracted, and then give us the possibility to access it.

We dedicate the fourth chapter to different research methods that address distributed knowledge resources. After giving an overview of research approaches on non-semantic knowledge resources, we explicate existing approaches capable of interrogating distributed semantic knowledge resources.

The second part of this thesis presents our contributions. In chapter 5, we describe our model, which is designed to represent a network of ontologies, where ontologies are resource representations of different formats and structures that respect the original resource content and structure. The model allows us to define the entity level connections between the resources and resource level connections. This model enables representing any ontology-based knowledge management approach. Thus, in the second section of chapter 5 we present our approach to ontologies network building based on a semantic knowledge management hybrid ontology-based approach. We describe the "Profile Ontology" which gives the organization's vocabulary that functions as semantic harmonizer of the network. We also present the different processes required to build a network of ontologies from the first step of knowledge extraction based on a design pattern approach through to the final stage of accessing knowledge via an algorithm that we developed.

Chapter 6 contains the details of how we implemented the different processes described in chapter 5. First, we describe examples of algorithms that can be used to extract knowledge from various resource types. As the knowledge extraction problem is not the goal of this work, we demonstrate that our approach can use any tool or method that respects the structure and the content of the original resources to obtain satisfying results. We also describe our "Profile Ontology" implementation process and the way in which the standard organization ontology can be used to facilitate the construction of a specific Profile Ontology. The final section of chapter 6 describes the different algorithms that we defined to build our model of a network of ontologies and the access to knowledge through it.

Chapter 7 is dedicated to evaluating all the processes of our approach, which is based on our model. Thus, we implemented a framework based on algorithms described in chapter 5 and 6, and we set up a use case, which is the GSEM faculty of Geneva University. First, we evaluate the knowledge extraction process by running the operation for five different resource formats. Then, we evaluate our model building approach by implementing ontology-based knowledge management processes, after which we compare the results. Finally, we test the search process, by assessing the performance of our prototype and discussing the possible points of evolution and improvement.

In chapter 8, we conclude our work by answering the questions that we articulated in chapter 1. This demonstrates that the research methodology described in section 1.3 allowed us to find answers to the problems we wanted to address and to detect possible evolution paths.

Chapitre 2. Background

In this chapter, we give an overview of our research domain, thus of semantic web technologies. we also describe the application domain, which is that of organizational knowledge. we define the concepts that will be used later when we explicate the “state of the art” contribution this study makes.

The first section is dedicated to defining *organizational knowledge* and to specifying its characteristics. The second section is devoted to defining semantic web basic concepts.

2.1 Organizational knowledge

2.1.1 What is organizational knowledge?

In the scholarly literature, many definitions have been put forward to distinguish between data, information, and knowledge. (Tsoukas & Vladimirou, 2001) Illustrates the difference between the three concepts clearly:

"Data is an ordered sequence of given items or events (e.g., the name index of a book). Information is a context-based arrangement of items whereby relations between them are shown (e.g., the subject index of a book). Knowledge carries a judgment of the significance of events and items, which comes from a particular context and theory (e.g., the construction of a thematic index by a reader of a book)”.

"Knowledge is organizational simply by its being generated, developed and transmitted by individuals within organizations" (Tsoukas, 2005).

Knowledge, as an organizational asset, has characteristics that distinguish it from other manageable resources (Wiig, Hoog, & Spek, 1997). Knowledge is intangible, and therefore difficult to measure. It is also volatile, and mostly embodied in people. It is not consumed in the process of being used, sometimes it is even increased by being used, and it has a high organizational impact as well. It is more and more evident that sharing and integrating organizational knowledge brings considerable benefits. In addition, sharing and combining knowledge enables people to understand the widespread effect of their actions on improving coordination and fostering synergy (Lubit, 2001).

As proof of the weightiness of knowledge management within an organization, the new ISO 9001:2015 standard introduces the term “knowledge.” As the previous ISO 9001 standard did not address knowledge, the approach to this topic is new and its depth is limited. ISO 9001:2015 defines requirements for the handling of organizational knowledge in the following four phases, which are analogous to the PDCA (Plan Do Check Act). Explicated, this entails that managers have (1) to determine the knowledge necessary for operating processes and for achieving conformity of products and services; (2) to maintain knowledge and make it available to the extent required; (3) to consider the current organizational knowledge and compare it to changing needs and trends; and (4) to acquire the essential additional knowledge.

By introducing the term “knowledge”, ISO 9001:2015 aims to raise organizations’ awareness of the management and linking of know-how to position them for the future (Matsokis, 2010).

> "Clause 7.1.6.: The Organization should determine the knowledge necessary for the operation of its processes and to achieve conformity of products and services. This knowledge shall be maintained and made available to the extent necessary. When addressing changing needs and trends, the organization shall consider its current knowledge and determine how to acquire or access any necessary additional knowledge and required updates. Organizational knowledge is knowledge specific to the organization; it is generally gained by experience. It is information that is used and shared to achieve the organization's objectives. Organizational knowledge can be based on: a) Internal Sources (e.g., intellectual property, knowledge gained from experience, lessons learned from failures and successful projects, capturing and sharing undocumented knowledge and experience; the results of improvements in processes, products, and services); b) External Sources (e.g., standards, academia, conferences, gathering knowledge from customers or external providers)."

Business knowledge generally is of two types, namely *explicit knowledge* that can be written down, transferred, and shared, that is definable and can be protected by the legal system; and *tacit knowledge* that entails know-how, and is by nature difficult to describe, that can be demonstrated but rarely codified, and that resides within its holder. Typically, it is transferred through demonstration and on the job training (Jarrar, 2002). Within this context, knowledge management means the "strategies and processes of identifying, capturing and leveraging knowledge to help the firm compete" (Pillania, 2005). In general, knowledge management is the process of continually managing knowledge of all kinds to meet existing and emerging needs, to identify and exploit existing and acquired knowledge assets and to develop new opportunities (Remenyi, 2007). It is a systematic process of underpinning, observation, instrumentation, and optimization of the firm's knowledge economies. Its overall purpose is to maximize the enterprise's knowledge-related effectiveness and the returns from its knowledge assets, as well as to continually renew their assets (Lehner & Fteimi, 2016).

(Arduin, Grundstein, & Rosenthal-Sabroux, 2015) determines two main complementary approaches to knowledge management: a managerial and a sociotechnical approach. These are used to integrate knowledge as resources in contributing to the strategic vision of the organization. The managerial approach is based mainly on the tacit knowledge in the organization. The technological approach addresses a need for a technical solution based on information and communication technologies. It is based on the codification of the organization's explicit knowledge, using informatics tools and specific techniques for its acquisition and representation (Jarrar, 2002). Knowledge management from the technological point of view is structured around three processes: knowledge codification, knowledge mapping, and knowledge storing. These processes will be described below.

Knowledge codification

This involves three main interrelated processes, which are (a) knowledge externalization, i.e. the process by which tacit knowledge rooted in an individual's action is transformed into explicit knowledge; (b) knowledge representation, i.e. the process which gives explicit knowledge a graphic form, and the possibility to use different information and communication codes (such as language, figures, charts); and (c) knowledge organization, i.e. the categorizing, structuring, and contextualizing of codified knowledge (Jarrar, 2002). Encoding knowledge results in different types of documents, structured in different ways and dispersed in databases, personal computers, and the likes.

Knowledge mapping

This is the process of identifying knowledge assets within the organization and defining ways of accessing them. Often, much of the knowledge people require to solve problems already exists within an organization, but it is not readily available when needed. Knowledge mapping is closely related to knowledge storing as depending on how we store the knowledge, we can define the best way to access it (Jarrar, 2002)..

Knowledge storing

This refers to where and how we store organizational knowledge to be available when needed. It can take the form of either knowledge databases, as has been developed by Ernst &Young who adopted best practices databases to support the activities of consultants worldwide, or by the *yellow pages* that provide links to people with specific know-how as has been developed by Hoffman-La Roche as a part of its overall Drug Approval Process Knowledge map. The latter includes a yellow page catalogue of recent experts that has been arranged according to guidelines of advancement of technologies and Information Systems (IS) associated with the search for success in the competitive market. This leads organizations to seek strategies that assist in acquisition, retention, storage, and dissemination of knowledge in the organization in order to re-use it later, thus preserving its Organizational Memory (OM).

2.1.2 Organizational Memory (OM)

OM supports the development of both the individual and the organization. For the individual, it is provided by aggregating knowledge and learning from the experiences, strategies, and actions the organization takes over time. For the organization, it is provided by using the range of accumulated knowledge that, when associated with the current knowledge of the individual, aids the organization in actions and decision making, as well as in generating new solutions, products, and services (Garvin, 1998). In order for OM to be useful and practical to the organization, it is necessary that the organization steers its strategy toward creating a favorable environment that fosters and encourages collaboration and the sharing of knowledge ideas, experiences and relevant information among its members, in order to feed this OM consistently (Barros, Ramos, & Perez, 2015). It is also essential to ensure that the Information System (IS) which supports its processes facilitates acquisition, retention, and dissemination of this knowledge in the organization (Gonzalez & Martins, 2017).

> “Since the Organizational Memory shows up as a fertile field of research while challenging, the purpose of this essay was to better understand its mechanisms of operation, associating them

> with the Information Systems, given the complexity and scope of such systems, which has as one of its main purposes, the preservation of organizational memory" (Pérez-Ramos, Rodríguez-Calcerrada, Ourcival, & Rambal, 2013).

Capturing knowledge about the history of an organization leads to stocks of resources that need a mechanism to facilitate the organization's employees in making use of it. Otherwise, these resources will have no benefit to the organization (Rexhepi, Ibraimi, & Veseli, 2013). "OM may be thought of as comprising stocks of data, information, and knowledge (the memories) that have been accumulated by an organization over its history. When an individual accesses OM, he performs an act of interpretation on the memory(ies) that is (are) accessed and may or may not act on it (them)" (Barros, Ramos, & Perez, 2015). Organizations create and use knowledge and information. To facilitate the use of knowledge and information, organizations are building and using organizational memory systems. These systems provide processes for capturing, searching and retrieving knowledge and information (Jennex & Olfman, 2004).

2.1.3 Knowledge management and Information management

It is important to distinguish between information management and knowledge management (KM). In information management, information is stored (usually in databases), sorted, and retrieved. Knowledge, on the other hand, requires a system that not only can save the existing knowledge as information, but can also extract and use that information when needed. In this manner, new knowledge can be created from existing knowledge in combination with further information (Edwards, 2016).

Organizational Memory Information Systems (OMIS) emerge to enhance the OM, providing practical support and resources for the organization, assisting in decision making, in the solution of problems, as well as in quality assurance and development of products and services. This article analyses a number of OMIS, selected from a literature review about OMIS features and functionality, to understand the organizations' perspective on these information systems. With this research, we realized that the relationship between OM and IS is still inexpressive even though some cases of success in the use of OMIS have been reported in the literature. The literature reveals that in most organizations individuals' knowledge is not integrated into the information systems management process. Much of this knowledge is generated in the organization, but retained in the individuals themselves. Therefore, there is an obvious need for strategies and mechanisms in the organization to stimulate and provide better knowledge sharing between individuals. When integrated with associated ISs, greater control and practical use of Organizational Memory (Rajpathak, Chougule, & Bandyopadhyay, 2012), know-how, questions, and issues (Jarrar, 2002) become possible. The steps above enable organizations to accumulate knowledge and preserve it over time, which is conducive to the building of Organizational Memory (OM).

2.1.4 Organizational knowledge management systems

There is no exact definition for 'knowledge management system' (KMS) in the relevant literature. However, it is mainly in referring to the technology that stores knowledge. We assume that a knowledge management system is an application designed to capture all the information within an organization and make it readily available to its employees, anywhere, anytime[1]. Knowledge management systems refer to any Information technology (IT) system that stores and retrieves knowledge, improves collaboration, locates knowledge sources, mines repositories for hidden knowledge, captures and uses knowledge, or in some other way enhances the KM process (Terzieva, 2014). Many researchers (Lehner & Fteimi, 2016) use the categorization of KMS proposed by (Shankar & Gupta, 2005). Depending whether it is about the object of the system or a technological strategy, we categorized the organizational systems according to KMS objects and technical approaches.

Groupware Systems (GS) and Collaborative software (CS)

GS and CS are management systems relating to collaborative work within an enterprise, enabling different users at different places to work together simultaneously or sequentially via the Internet (Bafoutsou & Mentzas, 2002). The computer's role in groupware is to provide well-defined mechanisms for interaction. Much of the groupware currently available in the corporate marketplace falls into this category, including many computer-supported cooperative work systems. The most common examples are systems for shared authoring, shared calendars, and project management that regulate interactions within a limited, task-specific focus (Sposito, 2007).

Decision Support Systems

A decision support system (DSS) is a computerized information system used to support decision-making in an organization or a business (Dulcic, Pavlic, & Silic, 2012). A DSS lets users sift through and analyze massive reams of data and compile information that can be used to solve problems and make better decisions. The primary purpose of using a DSS is to present information to the customer in a way that is easy to understand[2]. The benefit to a DSS system is that it can be programmed to generate many types of reports, all based on user specifications. A DSS can create information and output it graphically, such as a bar chart that represents projected revenue, or as a written report[3]. (Averweg, 2009) presents a full study of the history of the decision support system, with detailed classification of DSSs. Examples: MDSS (Strydom, 2005), OPTIMA[4].

Enterprise Content Management

The Association for information and image Management[5](AIIM) defined this system, as "Enterprise Content Management is the systematic collection and organization of information that is to be used by a designated audience – business executives, customers, etc. ECM is "neither a single technology nor a methodology

[1] http://www.dzonesoftware.com/blog/what-is-knowledge-management
[2] https://www.informationbuilders.com/decision-support-systems-dss.
[3] https://www.investopedia.com/terms/d/decision-support-system.asp.
[4] http://www.ess.co.at/OPTIMA/
4

nor a process, it is a dynamic combination of strategies, methods, and tools used to capture, manage, store, preserve, and deliver information supporting key organizational processes through its entire life-cycle"[6]. This category of enterprise management system aims to deal with the content of all resources within an organization; structured ones as databases and semi-structured ones (as invoices, purchase orders) and unstructured ones as (presentation, text documents ...) (Päivärinta & Munkvold, 2005).

Document management systems

According to iDatix[7] "document management encompasses the processes and procedures your organization uses as it pertains to capturing, storing, securing and retrieving information on a daily basis." Document management system aims to combine paper and digital document into a single hub (Smallwood, 2013). An example of a Document Management System: the online banking section of a bank's website that allows users to manage and access their monthly banking statements.

The above categories can be implemented based on one or multiples of technological architecture as:

Intranet

"An infrastructure based on Internet standards and technologies supports sharing of content within a limited and well-defined group"[8]. The intranet act as a platform for groupware applications and document management applications as it is intended to enhance collaboration and act as a repository for embedded knowledge[9].

Warehousing Data

It is "a federated repository for all the data collected by the various business systems of an enterprise"[10]. The goal of this technology is to generate front-end analytic that will support business executive and operational managers[11]. According to the diagram in Figure 2.1, the environment for data warehouses and marts includes the following parts:

- Pre-data warehouse: Source systems that provide data to the warehouse or mart.
- Data Cleansing: Data integration technology and processes that are needed to prepare the data for use.
- Data Repositories: Different architectures for storing data in an organization's data warehouse or data marts.
- Front-End Analytics: Different tools and applications for the variety of users.

---

[5] https://www.aiim.org/What-is-ECM

[7] https://www.docuphase.com/document-management-software

[8] http://www.iorg.com/papers/iw/19981019-advisor.html

[9] https://www.knowledge-management-tools.net/intranet-and-extranet.php

[10] https://searchcloudcomputing.techtarget.com/essentialguide/An-enterprise-guide-to-big-data-in-cloud-computing

[11] https://www.investopedia.com/terms/d/data-warehousing.asp.

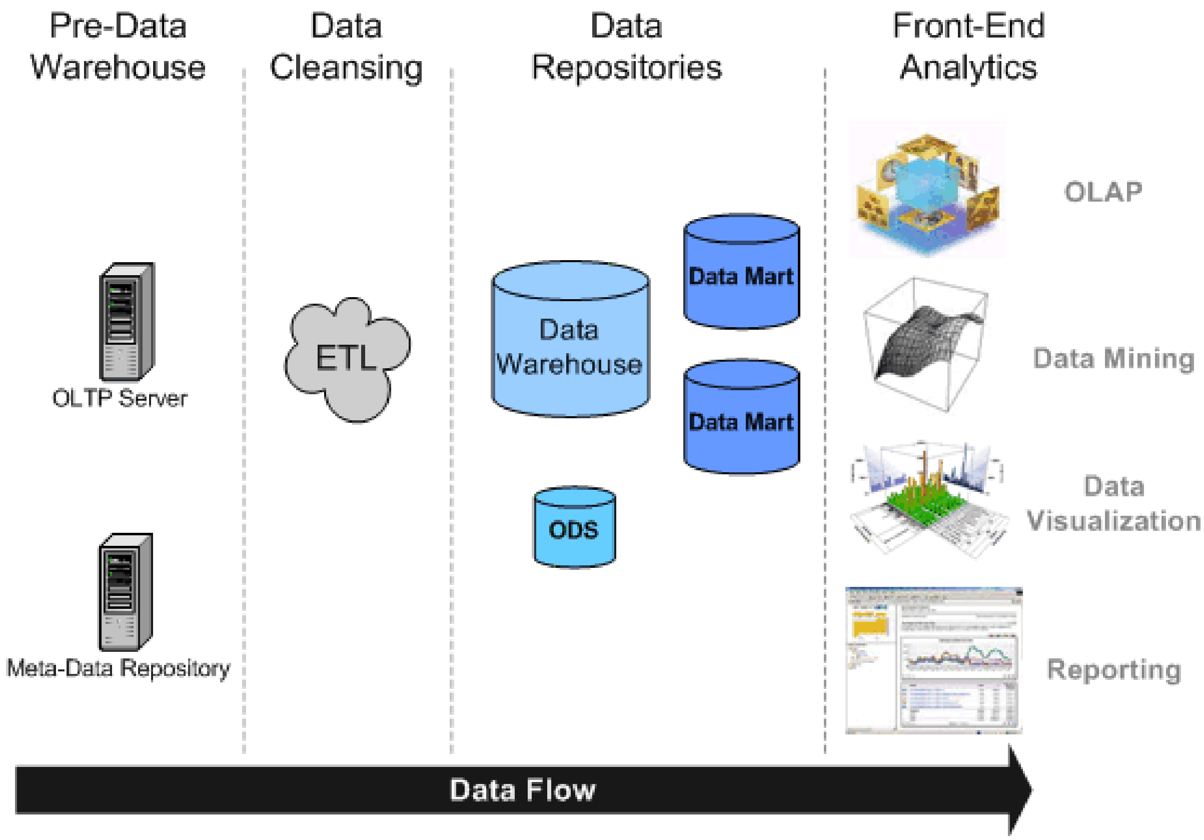

Figure 2.1: Data Warehousing Infrastructure

Data warehouses are expensive to scale and do not excel at handling raw, unstructured, or complex data. However, data warehouses are still an essential tool in the DSS era (Chatterjee, 2010).

Information retrieval systems

Information retrieval (IR) deals with access to information as well as its representation, storage, and organization. The overall goal of an information retrieval process is to retrieve the information relevant to a given request (Amudaria & Sasirekha, 2011). In Figure 2.2 (Ingwersen, 1992), which shows the basic concepts of an information retrieval system, representation is defined as the stored information, matching function as a specific search strategy for finding the stored information and queries as requests to the system for accurate information. Contemporary search engines are optimized for look-up scenarios where the information target is well defined, and human-machine interaction is limited to queries and search-result selections. In this role, the search system serves as a cognitive prosthetic, temporarily enhancing people's mental capabilities to provide access to additional information not known to the searcher or not readily accessible to them (Rajpathak, Chougule, & Bandyopadhyay, 2012). However, this type of support is insufficient for tasks requiring more involved information interaction (e.g., where cross-query learning may be necessary) and situations where information behavior encompasses more than just information seeking (Shah, 2010).

Figure 2.2: A simple model for information retrieval (Ingwersen, 1992)

There are three Information Retrieval (IR) models as follows (Goker & Davies, 2009),

- Boolean model - based on keyword manipulation, on the one hand, a document is represented by a combination of keywords, and on the other hand, a logical expression composed of words that represent a query.
- Vector model - representation of user queries and documents as vectors in the space generated by all the terms.
- Probabilistic model - uses the mathematical model based on the theory of probability.

The difference between data warehousing and information retrieving is that IR does not deal with the transnational update. Database systems deal with structured data, with a schema that defines the data organization. However, IR systems deal with some querying issues not generally addressed by database systems as approximate searching by keywords and ranking of retrieved answers by the estimated degree of relevance.

2.2 Basic concepts of semantic web technologies

The semantic web technologies provide a common framework that allows data to be shared and reused across application, enterprise and community boundaries[12]. The purpose of the Semantic Web is to make explicit the semantic content of resources (documents, web pages) (Uschold, 2003). Computers and agents' software could, therefore, "understand" information contained in these resources and help users to execute and complete the tasks, their requests more automatic and more efficient. The machines maintained could "know" that car is a kind of vehicle and that age of a person is a positive integer and does not have been higher than 150 years... Therefore, they could reason on knowledge already existing to provide more accurate information and systems that are more sophisticated and had better adaption to users' requirements.

[12] https://www.w3.org/2001/sw/

2.2.1 Ontologies

Ontologies are explicit, formal specifications of terms in the domain and the relations among them (Gruber, 1993) and act as a vehicle for seamless data integration and exchange (Rajpathak, Chougule, & Bandyopadhyay, 2012). Thus, they are well suited for the representation and utilization of relations among data and are efficient in reasoning.

Depending on their generality level, different types of ontologies may be identified that fulfill different roles in the process of building Knowledge Base Systems (KBS). (Fensel, 2013) distinguishes the following ontologies types:

- Domain ontology (or domain-specific ontology): it captures the knowledge valid of a particular type of domain and represents specific meanings of terms to that part of the world.
- Meta-data ontology: Like Dublin Core, it provides a vocabulary for describing the content of online information sources.
- Generic or common sense ontology: it aims at capturing general knowledge about the world, providing basic notions and concepts for things like time, space, state, and event. Therefore, they are valid across several domains.
- Others: Like so-called method ontologies, they provide terms specific to particular methods and task ontologies that offers specific terms for a specific task. They give a reasoning point of view on domain knowledge.

Ontologies representation languages: an ontology language is a formal language to represent an ontology. An ontology can be represented merely using:

- RDF: As defined by W3C, RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schema differ, and it specifically supports the evolution of schema over time without requiring all the data consumers to be changed. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a “triple: subject-predicate-object”). Using this simple model, RDF allows structured and semi-structured data to be mixed, exposed, and shared across different applications. RDF has no mechanisms for defining relationships between properties and between resources is the role of RDF Schema (RDFS). Using RDF and RDFS, knowledge can be represented not only at the level of the assertions (or facts) but also at the level of conceptualization (ontology level). The predefined semantics in RDFS makes it possible to reason and deduce new knowledge. The documents in RDFS are often serialized (written) in XML, which makes it possible for several tools already existing for XML by processing the documents in RDFS. For more of readability, other languages are also proposed to represent the models RDFS in formats such as Notation 3 (N3), NTriples, and Turtle. We can use RDFS to represent simple ontologies. The ontologies in RDFS can be serialized into languages such as XML or N3. However, an ontology is used to describe knowledge in an area, so it is necessary to have an expressive language to express them, and RDFS does not meet these needs. For example, using RDFS, we cannot describe the cardinality of a relation or express characteristics of relationships such as transitivity, symmetry or functionality, or to make restrictions for specific classes.

- OWL: Ontology web language: W3C has recommended a more powerful standardized language at the expressive level, which is specially designed to represent ontologies in the semantic Web that allows with ease to create, share and exchange knowledge in the Semantic Web. The OWL language is recommended. It is derived from language DAML + OIL. OWL covers most features of the DAML + OIL language and renames most of its primitives. The OWL ontology language is divided into three sublanguages with a tapering expressiveness power: OWL Full, OWL DL and OWL Lite. The reason concerns the complexity, the computability and the implementation of the language. While the OWL Lite sub-language has the lowest formal complexity and the minimum expressiveness in the family, it is sufficient to represent thesaurus and other taxonomies or classification hierarchies with simple constraints. The tools to support OWL Lite and migrations for the thesauri, the existing taxonomies are also cheaper. OWL DL has more expressiveness while maintaining computational completeness (all conclusions are guaranteed to be computed) and decidability (all calculations will end in finite time). OWL DL corresponds to the variant of the description logic SHOIN (D) (Bruijn, Lara, Polleres, & Fensel, 2005). The OWL DL sub-language is therefore appropriate to represent ontologies needing the power of expressiveness while keeping the computability. Finally, OWL Full is designed for developers, implementers, and users who require maximum expressiveness, freedom of syntax of RDF but no computability guarantee. Currently, no tools or reasoning software are capable of supporting complete reasoning for all the features of OWL Full.

Regarding the compatibility of these sub-languages, the Full OWL sub-language can be considered as an extension of RDF, while OWL Lite and OWL DL can be regarded as extensions of a restricted view of RDF. OWL Full is an extension of OWL DL, and the latter is an extension of OWL Lite. A legal ontology in OWL Lite is also legal in OWL DL and OWL Full. All documents in OWL (Full, DL, and Lite) are valid documents in RDF, and an RDF document is an OWL Full document, but only a few RDF documents are legal documents in OWL Lite or OWL DL (Bruijn, Lara, Polleres, & Fensel, 2005).

Components regardless of the language in which they are expressed, most ontologies describe individuals that are instances of the classes, which are concepts having attributes and related to each other by relations.

- Individuals (instances) are the primary, "ground level" components of an ontology. The individuals in an ontology may include concrete objects such as people, animals, tables, automobiles, molecules, and planets, as well as abstract individuals such as numbers and words (although there are differences of opinion as to whether numbers and words are classes or individuals). Strictly speaking, an ontology needs not include any individuals, but one of the general purposes of an ontology is to provide a means of classifying individuals, even if those individuals are not explicitly part of the ontology (Linckels & Meinel, 2011).

- Concepts that are also called types, sorts, categories, and kinds – can be defined as an extension or an intention. According to an extensional definition, they are abstract groups, sets, or collections of objects. According to an intentional definition, they are abstract objects that are defined by values of aspects that are constraints on being a member of the class. The first definition of class results in ontologies in which a class is a subclass of the collection. The second definition of class results in ontologies in which groups and classes are more fundamentally different. Classes may classify individuals, other classes, or a combination of both.
- Attributes: Objects in an ontology can be described by relating them to other things, typically aspects or parts. These related things are often called attributes, although they may be independent things. Each attribute can be a class or an individual. The kind of object and the kind of attribute determines the sort of relation between them. A relation between an object and an attribute express a fact that is specific to the object to which it is related.
- Relationships (also known as relations) between objects in an ontology specify how objects are related to other objects. Typically, a relation is of a particular type (or class) that defines in what sense the object is related to other objects in the ontology. Much of the power of ontologies comes from the ability to describe relations. Together, the set of relations describes the semantics of the domain. The set of used relation types (classes of relations) and their subsumption hierarchy describe the expression power of the language in which the ontology is expressed. An important type of relation is the subsumption relation (is-a-superclass-of, the converse of is-a, is-a-subtype-of or is-a-subclass-of). These relations define which objects are classified by which class. Another common type of relations is the mereology relation, written as part-of that represents how objects combine to form composite objects. As well as standard relationships, ontologies often include additional types of relations that further refine the semantics they model.
- Axioms: Assertions (including rules: if – the sentence that describes the logical inference) in a logical form that together comprise the overall theory that the ontology describes in its domain of application. This definition differs from that of "axioms" in generative grammar and formal logic. In those disciplines, axioms include only statements asserted as prior knowledge. As used here, "axioms" also include the theory derived from axiomatic statements

Many methodologies, such as TOVE (López, 2002), EO (Dietz, 2006), Methodology (Gómez-Pérez & Suárez-Figueroa, 2009) are described in the literature based on experiences developing specific domain ontologies. These methodologies have similar points, which are starting by identifying the purpose, or specifying motivation scenarios by defining a set of problems encountered in particular enterprise or determining the level of formality at which the ontology should be described as per (Paredes-Moreno, Martínez-López, & Schwartz, 2010). The second step is to produce the specification ether formal or informal to identify the scope of the ontology. The stage of formalization that is coding or implementing the ontology and finally the phase of evaluation, if yes or not the ontology can answer the questions selected in the first and second steps. The mains differences between the three methodologies are that EO methodology apart from the formal and informal levels. Methodology introduces the level of knowledge acquisition via expert interview and all documents specifying the vocabulary used within the same organization, integration which is the step of selecting ontological resources to uniform the ontology, and documentation of the stages of the ontology creation process.

2.2.2 Ontologies' Alignment

Definition

Alignments express the correspondences between entities of different ontologies. Given two ontologies O and O' with associated entity languages Q_L and $Q_{L'}$ and a set of alignment relations Θ, correspondence is a triple: (e, e',r) = Q_L(O)x $Q_{L'}$(O') x Θ expressing that the relation r holds between entity e and e'.

An alignment is a set of correspondences between two ontologies (De Bruijn, et al., 2006). The entity language can be made merely of all the terms or formulas of the ontology language based on the ontology vocabulary. It can restrict them to the named terms or, on the contrary, extend them to all the queries that may be expressed on this vocabulary. Alignments express relations between such entities through a finite set of connections, which are independent of ontology relations (Euzenat, 2014). Ontology alignment is generally described as the application of the so-called match operator (Poli, Healy, & Kameas, 2010). There are many different algorithms, which implement the match operator. These algorithms can usually be classified along two dimensions. There is the distinction between schema-based and instance-based matching (Badr, Chbeir, Abraham, & Hassanien, 2010).

Alignment types

The types of alignments are related to the kinds of resources to be aligned and to the specifications of the matching tool. We identified three categories of alignment resources:

- Formal alignment: matching resources are represented in a logical formalism and using logical relations to express correspondences between their entities. These entities can be simple (node entities) or complex represented as logical expressions using specific constructors (i.e., description logic constructors, first-order logic expressions, etc.) (Suchanek, Kasneci, & Weikum, 2007).
- Terminological alignments: bridging between resources represented in a semantic formalism and using terminological relations (Broader_Term, BroadMatch, Translation, ExactMatch, etc.) to express semantic correspondences between entities (generally terminological or conceptual entities) (Isaac, et al., 2009), (Kefi, Safar, & Reynaud, 2006), (Noy, et al., 2001).
- Linguistic alignments: expressing correspondences between two syntactic structures generally for multilingual resources. It is usually expressing equivalency between two parse trees to match different n-grams in two sentences (Pang, Knight, & Marcu, 2003) or directly maps of translation links between pairs of linguistic entities.

2.2.3 Annotation

Here are the definitions of (Oren, Moller, Scerri, Handschuh, & Sintek, 2006)*:*

Definition 1 (Annotation)

An annotation is a tuple (a_s, a_p, a_o, a_c), where a is the subject of the annotation (the annotated data) a_o is the object of the annotation (the annotating data), a_p is the predicate (the annotation relation) that defines the type of relationship between a_s and a_o, and a_c is the context in which the annotation is made. The annotation subject can be formal or informal.

For example, when we put a note in the margin of a paragraph, the informal convention is that the note applies to the section, but that pointer is not formally defined. However, if we use a formal pointer such as a URI to point to the section then the subject is specified officially.

Definition 2 (Formal annotation)

A formal annotation A_f is an annotation A, where the subject a_s is a URI, the predicate a_p is a URI, the object a_p is a URI and the context a_c is a URI.

Definition 3 (Ontological annotation)

An ontological annotation A_s is a formal annotation A_f, where the predicate a_p and the context a_c are an (arbitrarily complex) ontological term, and the object a_o conforms to an ontological definition of a_p.

2.2.4 Semantic querying languages

RDF querying languages

The SPARQL[13] query language for RDF is designed to meet the use cases and requirements identified by the RDF Data Access Working Group. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs. SPARQL has facilities to i) Extract RDF subgraphs, ii) Construct a new RDF graph using data from the input RDF graph queried, iii) Return “descriptions” of the resources matching a query part, iv) Specify optional triple or graph query patterns (i.e., data that should contribute to an answer if it is present in the data queried, but whose absence does not prevent an answer being returned), and v) Test the absence, or non-existence, of tuples.

Ontology querying languages

Query languages for OWL are still in their infancy compared to those for RDF (Bailey, Bry, Furche, & Schaffert, 2009). OWL query language (Fikes, Hayes, & Horrocks, 2004) is a well-known language for querying OWL data and is an updated version of the DAML Query language (Fikes, Hayes, & Horrocks,

[13] https://www.w3.org/TR/rdf-sparql-query/

2004). Its design targets the assistance of query-answering dialogues between computational agents on the Semantic Web. Unlike the RDF query languages, it focuses on the querying of schema rather than instance data. An RDF language such as SPARQL may, of course, be used to query OWL data, but it is not well suited to the task as it is not being designed to be aware of OWL-semantics (Haizhou & Kemafor, 2011).

2.3 Conclusion

As showed, many approaches and tools have been developed for different tasks to optimize the management of the knowledge within an organization with different strategies to avoid the heterogeneity of knowledge types with the goal to maximize the exploration and exploitation of knowledge resources content. We also present the architecture of information retrieval approach, which presents a potential methodology to develop many types of knowledge management systems as document management and enterprise content management systems. In the second section, we detail the principal notions of the semantic web and its available technologies and resources and explain why the nowadays-semantic web is the solution to give mining to information and to deal with heterogeneity what we consider a promising solution to give exact mining to resources content and to develop semantic management systems for organizations. In the next chapter, we present the state of the art of semantic knowledge management solutions based on different approaches proving that ontology-based approaches are a promising solution not only to deal with heterogeneity but also to give meaning and to contextualize knowledge.

Part I: State of the art

Chapitre 3. Semantic knowledge management

3.1 Introduction

As seen in chapter two, the knowledge developed during the activities of a company can be expressed explicitly in acts, documents, or experiences. All these types of knowledge and resources, as well as the information crucial to the company, are called “business memory” or “organizational memory” (section 2.2.1). To facilitate and improve access, sharing, and re-use of corporate memory, and to create or deduct new knowledge, it is necessary to have the means and tools to materialize the memory and to index the content of this corporate memory (Dieng, Corby, Giboin, & Ribière, 1999). By drawing an analogy between the resources of corporate memory and those of the Web, we can use Web technologies for the management, representation, and exploitation of knowledge in an organization by implementing and using the Intranet within the organization (section 2.2.2). With the evolution of the Web to the Semantic Web, new technologies for the latter can also be applied in the organization: corporate memories are constructed in the form of the corporate semantic Web (Dieng, Corby, Giboin, & Ribière, 1999). Research of the past decade has started to refer to *next generation frameworks* based on the Semantic Web to mainstream enterprise applications (Dau, 2011): "The emerging Semantic Web will require us to dramatically rethink traditional notions of how business, data/information, application, and technology architectures are conceptualized and realized within an enterprise"[14].

3.2 Knowledge representation models

In this section, we present different models and languages created to handle the task of knowledge representation. Generally, each model represents a specific aspect of the resources without covering all their types: ontological, terminological, lexical, textual, documentary, etc. It is more difficult to find models representing a variety of knowledge resources of different kinds.

The major's transformation approach from knowledge resources to resource representation are:

1. Tbox transformation (Caracciolo, Heguiabehere, Sini, & Keizer, 2009): it transforms the resource content into an ontology schema. This transformation approach tries to enforce a formal semantics to the re-engineered resources, even at the cost of changing their structure. The requirements for this transformation are:
    - Full conversion: the resultant ontology has all the information that is present in the original resource. In other words, all queries that are possible on the source should also be possible on the ontology generated.

[14] https://www.cutter.com/journal/rise-semantic-enterprise-487151.

- Conversion on the semantic level: it implies that the schema translation interprets the semantics of the data. In other words, the conversion should not avoid possible interpretations.

2. Abox transformation (Caracciolo, Heguiabehere, Sini, & Keizer, 2009): it transforms the resource schema into an ontology schema, and the resource content into ontology instances. This transformation approach leaves the informal semantics of the re-engineered resources mostly untouched. The requirements for this transformation are
   - Full conversion: it is the same condition than for the TBox transformation. Again, this implies that all queries that are possible on the source should also be possible on the ontological version.
   - Structure preserving translation: it is the opposite of the second requirement of the TBox transformation. The translation should respect as much as possible the original structure of the resource; in other words, the conversion should avoid possible interpretations.
3. Population: it transforms the resource content into instances of an ontology. The requirements of the transformation are full conversion, the same need for the TBox and ABox change. The ontology instances generated should respect the target ontology structure as strictly as possible. In this case, the class structure of the ontology already exists and is extended with instance data. In other words, the ontology instances must conform to the already existing ontology schema.

Explicit digital knowledge resources are composed of content and information about the content and the context of the support, which are the metadata. To represent these two components, we investigate models capable of serving these types of knowledge.

3.2.1 Metadata representation models

Metadata can be defined as structured and machine understandable collections of lexical items to be used for expressing assertions about the organization and the contents of some sets of digital or non-digital documents (Zarri, 2015).

There exists a wide variety of metadata formats for digital resources. They range from relatively simple ones, like the Dublin Core basic scheme, to more detailed forms like the Text Encoding Initiative [15] and MARC [16]. Dublin Core has recently evolved into the Dublin Core Metadata Initiative DCMI [17]. The “DCMI Metadata Terms” is a broad set of metadata elements that includes the fifteen ones of the original Dublin Core along with several related properties and classes (Coyle, 2012). Following the publication of the RD-compatible DCMI Abstract Model [18], Dublin Core represents now one of the vocabularies most commonly used in a W3C/SW/RDF context (Figure 3.1).

[15] http://www.tei-c.org/index.xml
[16] http://www.loc.gov/marc/uma/pt1-7.html
[17] http://dublincore.org/
[18] http://dublincore.org/documents/2007/02/05/abstract-model/

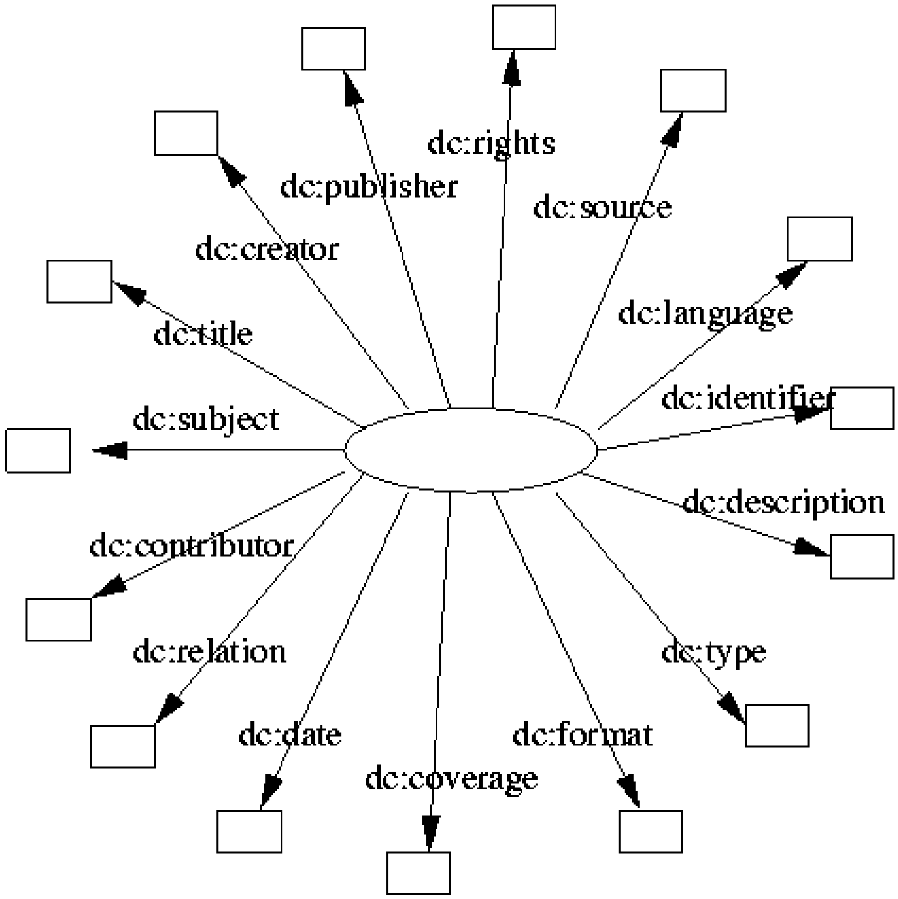

Figure 3.1: Unqualified DC: The hedgehog model (DCMI)

The Dublin Core Metadata element set[19] is composed of:

- Contributor: An entity responsible for making contributions to the resource
- Coverage: The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant.
- Creator: An entity primarily responsible for making the resource. Comment: Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.
- Date: A point or period associated with an event in the life cycle of the resource. Comment: Date may be used to express temporal information at any level of granularity.
- Description: An account of the resource. Comment: Description may include but is not limited to an abstract, a table of contents, a graphical representation, or a free-text account of the resource.
- Format: The file format, physical medium, or dimensions of the resource.

[19] http://www.dublincore.org/documents/dcmi-terms/

- Identifier: An unambiguous reference to the resource within a given context. Comment: Recommended best practice is to identify the resource using a string conforming to a formal identification system.
- Language: A language of the resource.
- Publisher: An entity responsible for making the resource available. Comment: Examples of a Publisher include a person, an organization, or a service.
- Rights: Information about rights held in and over the resource. Comment: Typically, rights information includes a statement about various property rights associated with the resource, including intellectual property rights.
- Source: A related resource from which the described resource is derived.
- Subject: The topic of the resource. Typically, the subject will be represented using keywords, key phrases, or classification codes. Recommended best practice is to use a controlled vocabulary.
- Title: A name given to the resource.

Some models, such as NoRMV[20]which is a meta-data vocabulary describing non-ontological resources, provide a categorization of non-ontological resources. NoRMV (Figure 3.2) is an extension of OVM[21] that provides a vocabulary describing ontological resources (Hartmann, et al., 2005). Other approaches focus on representing the knowledge within resources by specifying a model or transforming rules for each type of resources, trying to identify what knowledge can be extracted using the available technologies.

[20] http://mayor2.dia._.upm.es/oeg-upm/index.php/en/ontologies/177-normv/index.html

[21] http://mayor2.dia._.upm.es/oeg-upm/index.php/en/downloads/75-omv/index.html

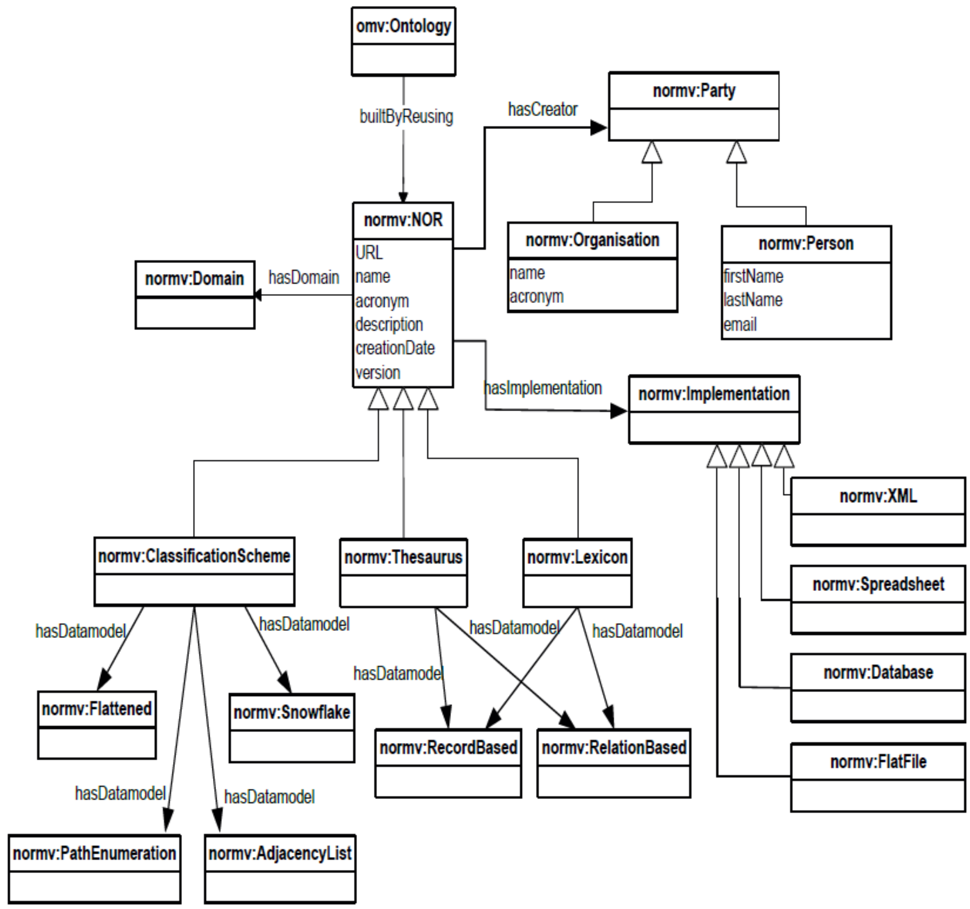

Figure 3.2: NoRMV: Nonontological resource metadata vocabulary (Villazón-Terrazas, 2012)

3.2.2 Content representation models

Depending on the initial format of a knowledge resource, we investigate the tools aiming to map non-ontological resources to ontological resources. Non-ontological resources can be structured as databases and XML-based schema resources or unstructured as spreadsheets and text document.

Structured resources transformation

Relational data base (RDB) to OWL

Researcher initiatives related to RDBs and ontologies transformations proliferate and are addressed continuously. Considering RDF, OWL ontologies and RDBs, several frameworks, mapping approaches and tools exist. These approaches and tools are referenced and analyzed in the W3C (World Wide Web

Consortium): Virtuoso RDF View[22]; D2RQ[23] ; Triplify[24]; RDBToOnto (Krivine, Nobécourt, Soualmia, Cerbah, & Duclos, 2009) ; R2O (Barros, Ramos, & Perez, 2015), SBRD Automapper (Fisher & Dean, 2008); RDB2OWL[25] ; ERONTO (Fahad, 2008); OWLFromDB (Zhang, Di, & Feng, 2012). (Chbihi Louhdi, Behja, & El Alaoui, 2013) presents a comparative study of several methods of transforming relational databases (RDB) into the ontology. Based on this survey (Table 3.1), we compared the methods based on the source of the ontology transformation: conceptual model (1) or relational model (2), and the criteria of availability and integrability (3) that we assume very useful in scientific context, as in our case, we need existing tools to implement our approach. Thus, we investigate if the methods deal with one-to-many relations (4), many-to-many relation (5), many-to-many relation with attributes (6) and n-ary relation (7).

Methods	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**
(He-ping, Lu, & Bin, 2008), (Santoso, Haw, & Abdul-Mehdi, 2011)	no	no	yes	yes	no	no	no
(Gagnon, 2007) (Cerbah, 2008) (He-ping, Lu, & Bin, 2008)	no	no	yes	yes	yes	no	no
(Xu, Cao, Dong, & Su, 2004) (Astrova, Korda, & Kalja, 2007)	no	yes	no	yes	yes	yes	yes
(Myroshnichenko & Murphy, 2009)	no	no	yes	yes	yes	yes	yes
(Fahad, 2008)	no	yes	no	yes	yes	yes	yes
(Li, Du, & Wang, 2005), (Telnarova, 2010)	no	yes	no	yes	yes	no	yes
(He-ping, Lu, & Bin, 2008), (Ahmed, Aslam, Shen, & Yong, 2011), (Cullot & Yétongnon, 2007), (Hazber, Yang, & Jin, 2010), (Xu, Cao, Dong, & Su, 2004)	no	yes	no	yes	yes	no	no
(Chbihi Louhdi, Behja, & El Alaoui, 2013)	no	yes	no	yes	yes	yes	yes

Table 3.1: Comparative study overall several transforming RDB to ontology methods (Chbihi Louhdi, Behja, & El Alaoui, 2013)

Even if the source code is not available and no API is created to be integrated into future work, we consider that the explanations given over various papers are sufficient for possible implementation.

[22] https://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html

[23] http://d2rq.org/

[24] https://www.w3.org/2001/sw/wiki/Triplify

[25] http://rdb2owl.lumii.lv/

XML to OWL

The issued to transform XML to OWL has been addressed from the problem of converting metadata schema to ontologies. These approaches are generally generic approaches dealing with the problem of giving an arbitrary metadata document (e.g., XML file) how it can be automatically transformed to an ontology representation (e.g., OWL or pdf file) (Munir & Sheraz Anjum, 2017). For example, issues that are an often addressed in the conversion are how XML components, such as XML sequences, elements/attributes, identifiers, should be reported presented in RDF language (Battle, 1999); (Bohring & Auer, 2005).

The generic approaches usually convert metadata records on the syntactic level, in the sense that they consider mapping between XML and RDF structural components. It has been concluded that there is no complete approach, which focuses on a transformation from ”legacy” XML instance documents to OWL ontologies (Svátek, Sváb-Zamazal, & Vacura, 2010). In (Ferdinand, Zirpins, & Trastour, 2004) the authors even assume, that a suitable automatic mapping between XML and RDF is impossible because XML does not contain any semantic constraint. It is claimed that XML represents the document structure, but does not include any information about the meaning of the content. Contrary, Melnik approach (Klein, 2002) assumes that there is some semantics in the XML documents, which can be discovered out of the document structure, it detects semantics in XML instance documents and maps them to RDF documents with a simplified syntax assuming that every XML document has an RDF model. (Reif & Gall, 2009) proposes an automatic mapping from XML contents to RDF metadata by using an ontology, which was created from the corresponding XML Schema. This ontology contains the model but has no instances. The XML data will not be mapped to its OWL equivalents. By now, the mapping from the XML Schema to the OWL ontology is created manually. Our aim is it to be able to generate the mapping automatically. (Thuy, Lee, & Lee, 2009) describes mappings from XML to RDF as well as from XML Schema to OWL, but these mappings are independent of each other. That means, that OWL instances have not necessarily to suit to the OWL model, because elements in XML documents may have been mapped to different elements in the OWL. Furthermore, this approach does not tackle the question of how to create the OWL model, if no XML Schema is available. (Yahia, Mokhtar, & Ahmed, 2012) proposes a strategy of how OWL ontologies may be generated automatically out of existing XML data. This transformation has to be done by establishing suitable mappings between the different data model elements of XML and OWL. Figure 3.3 is a comparative study done by (Hacherouf, Bahloul, & Cruz, 2015) that demonstrates a broad set of approaches designed to transform the XML document into an OWL representation. These approaches apply to convert HTML document to OWL, but they need to be modified/instantiated to deal with the particular structure of HTML files.

Approaches	Inputs	Outputs	OWL Type	Generation of ontology and individuals	Rules	Integrity constraints	Existing ontology
OWLMAP	XML schema +XML instance	OWL schema +RDF graph	OWL-DL	Sequentiel	Automatic	YES	NO
XML2OWL	XML schema +XML instance	OWL schema +individual	OWL-DL	Sequentiel	Automatic	YES	NO
XS2OWL	XML schema	OWL schema	OWL-DL	n/a	Automatic	YES	NO
XSD2OWL	XML schema +XML instance	OWL schema +individual	OWL-DL	Sequentiel	semi-Automatic	NO	YES
X2OWL	XML schema	OWL schema	OWL-DL	n/a	Automatic	NO	NO
JANUS	XML schema	OWL schema	OWL2-RL	n/a	Automatic	YES	NO
EXCO	XML schema +XML instance	OWL schema +individual	OWL-DL	parallel	Automatic	YES	NO
(Yahia, Mokhtar, & Ahmed, 2012)	XML schema	OWL schema +individual	OWL-DL	n/a	Automatic	YES	NO
DTD2OWL	DTD+XML instance	OWL schema +individual	OWL-DL	Sequential	Automatic	YES	NO

Approaches	Inputs	Outputs	OWL type	Generation of ontology and individuals	Rules	Integrity constraints	Existing ontology
OWLMAP [18]	XML schema + XML instances	OWL schema + RDF graph	OWL-DL	Sequential	Automatic	Yes	No
XML2OWL [12]	XML schema + XML instances	OWL schema + individual	OWL-DL	Sequential	Automatic	Yes	No
XS2OWL [13]	XML schema	OWL schema	OWL-DL	n/a	Automatic	Yes	No
XSD2OWL [1]	XML schema + XML instances	OWL ontology + individual	OWL-DL	sequential	Semi-automatic	No	Yes
X2OWL [14]	XML schema	OWL schema	OWL-DL	n/a	Automatic	No	No
Janus [15]	XML schema	OWL schema	OWL2-RL	n/a	Automatic	Yes	No
EXCO [16]	XML schema + XML instances	OWL schema + individual	OWL-DL	parallel	Automatic	Yes	No
Yahia et al. [17]	XML schema	OWL schema + individual	OWL-DL	n/a	Automatic	No	No
DTD2OWL [19]	DTD + XML instances	OWL schema + individual	OWL-DL	Sequential	Automatic	Yes	No

Figure 3.3: Survey: Transforming approaches from XML to OWL (Hacherouf, Bahloul, & Cruz, 2015)

Semi-structured resource transformation

HTML to OWL

HTML stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages, it describes the structure of Web pages using markup. Its elements are the building blocks of HTML pages and are represented by tags, which label pieces of content such as "heading," "paragraph," "table", and so on. Browsers do not display the HTML tags but use them to render the content of the page (introduction to HTML).

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

The <!DOCTYPE html> declaration defines this document to be HTML5
The <html> element is the root element of an HTML page
The <head> element contains meta information about the document
The <title> element specifies a title for the document
The <body> element contains the visible page content
The <h1> element defines a large heading
The <p> element defines a paragraph

Figure 3.4: HTML pages structure

As there is no specific tool or methods that describe the transformation from HTML files to OWL and as the difference between XML and HTML is over the syntax, which is more rigorous in XML (and thus in XHTML 1.0) than in HTML, we investigate tools and methods for transforming XML to OWL.

Web services transformation

Web services are considered as self-contained, self-describing, module applications that can be published, located, and invoked through the Web (Dargam, et al., 2014). Several languages have emerged, to add semantic description features to the web services standards as OWL-S (formally DAML-S) which is a revision of DAML+OIL (Wali & Gibaud, 2012). OWL-S (formerly DAML-S) is an ontology of services that allows user and software agent to discover, invoke, compose and monitor web resources offering particular services and having specific properties with a high degree of automation (Satapathy, Avadhani, & Abraham, 2011).

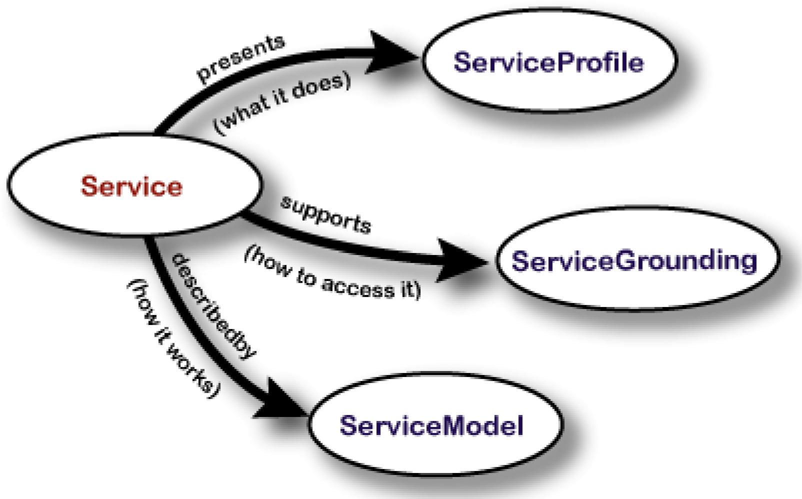

Figure 3.5: The principal OWL-S classes (Daniel, Papadopoulos, & Thiran, 2013)

The overall structure of the ontology and its three main parts are the service profile for advertising and discovering services; the process model, which gives a detailed description of a service's operation; and the grounding. This structure provides details on how to interoperate with a service via messages. The structure is described and recommended by the Web-Ontology Working Group[26] at the World Wide Web Consortium. Web services adapting OWL-S ontology are directly usable and integrable in any semantic-based tool.

[26] https://www.w3.org/2001/sw/WebOnt/

Unstructured resources transformation

Spreadsheets transformation

A variety of systems has been developed to map spreadsheet content to RDF. The earliest systems include Excel2RDF[27] and Convert2RDF[28]. Both systems support necessary mappings from entity-per-row spreadsheets. RDF123 system (Han, Finin, Parr, Sachs, & Joshi, 2008) supports less restricted data models. However, it is still primarily row-centric. While complex mapping conditions are supported, the mapping language still fundamentally assumes entity-per-row storage. The XLWrap[29] system attempts to address this shortcoming. It allows data to be organized in essentially arbitrary ways and supports an expressive mapping language for generating RDF content. Other primarily metadata-based systems include MIT's Simile project[30], Aperture/Nepomuk from Semantic Discovery Systems (Grimnes, Sauermann, & Bernardi, 2009), and Cambridge Semantic's Anzo for Excel[31]. Some systems use an XSLT[32]-based approach to map automatically-generated XML representations of spreadsheets to RDF. However, these approaches can be very cumbersome and are generally used for only a small range of simple mappings. A related, higher-level approach is to use importation tools to generate OWL or RDF tabular representations of spreadsheet data and then to map these tabular representations to domain ontologies using a rule or scripting languages. For example, TopBraid[33]Composer's SPARQL Motion[34] provides a range of scripting modules for generating RDF from tabular data imported from spreadsheets. The authors have used a similar approach with a data importation tool called DataMaster (Nyulas, O'Connor, & Tu, 2007) that uses SWRL[35] rules to map spreadsheet data imported by DataMaster (Nyulas, O'Connor, & Tu, 2007) to domain-level constructs. While these approaches provide great flexibility, a multitude of rules or mapping scripts can quickly accumulate, which can be challenging to manage and debug. Recent approaches on mapping information contained in spreadsheets to OWL suffer from a variety of limitations, including assuming well-formed spreadsheets reminiscent of a single relational database table and verbose syntaxes for expressing mapping rules. To overcome these limitations (O'Connor, Halaschek-Wiener, & Musen, 2010) developed a mapping language, M2, which is based on an extension of the OWL Manchester Syntax that supports arbitrary spreadsheet cell references. This mapping language provides a compact, user-friendly approach for expressing mapping rules for arbitrary spreadsheets. The language also supports syntactic transformations of cell contents, as well as inline OWL axioms involving classes, properties, and individuals extracted from cell contents.

---

[27] http://www.mindswap.org/~rreck/excel2rdf.shtml
[28] http://www.mindswap.org/~mhgrove/ConvertToRDF
[29] http://xlwrap.sourceforge.net/
[30] http://simile.mit.edu/
[31] https://www.w3.org/2001/sw/wiki/Anzo
[32] https://www.w3.org/TR/xslt/
[33] https://www.w3.org/2001/sw/wiki/TopBraid
[34] http://sparqlmotion.org/s
[35] https://www.w3.org/Submission/SWRL/

Below a comparative study where we specified several criteria:

1. Row centric approach: which shows if the method concern simple transformation or a complex one.

2. Well-formed: If the spreadsheet needs to follow a template or not.

3. Direct mapping: if the method requires more than one-step to do the transformation (complex)

Methods	**row-centric**	**well-formed spreadsheets**	**direct mapping**	**embeddable**	**available**
ExceltoRDF[36] Convert2RDF[37]	yes	yes	yes	no	no
RDF123 (Han, Finin, Parr, Sachs, & Joshi, 2008)	yes	no	yes	no	no
XLwarp[38]	no	no	yes	no	no
Motion [39] /DataMast (Nyulas, O'Connor, & Tu, 2007)	no	yes	no	no	no
M2 (O'Connor, Halaschek-Wiener, & Musen, 2010)	no	no	yes	yes	yes

Table 3.2: Comparative study from Spreadsheets to OWL

Conclusion

Different models and approaches were designed to extract knowledge from different types of resources. From this section, we can conclude that no approach or tool can automatically parse any knowledge resource independently from its kind or structure to an ontology. This is because of the different structures of knowledge resources. In addition, each method aims to extract a specific content that it considers useful for the approach. We assume that an embeddable personable tool is still needed to execute resources content extraction tasks. However, we can use algorithms described in several studies to implement transformations methods.

36 http://www.mindswap.org/~rreck/excel2rdf.shtml
37 http://www.mindswap.org/~mhgrove/ConvertToRDF
38 http://xlwrap.sourceforge.net/
39 http://sparqlmotion.org/

3.3 Ontology-based integration of information

Ontology has shown promise in removing interpretation problems by computationally capturing the semantics of concepts, ensuring their consistency and thus providing a correct and shared understanding across multiple domains (Usman, et al., 2013). In nearly all ontology-based integration approaches, ontologies are used for the explicit description of the information source semantics. Nevertheless, there are a different way of how to employ the ontologies in each part of an ontology-based integration system.

As seen in section 2.2.2, an information retrieval system is composed of three parts: representation module, integration module (or matching function part) and query module that we detail in the next chapter. Each part of the system is defined by the ontology-based approach that we can choose and use to build our system.

3.3.1 Ontology-based knowledge representation approaches

Single ontology approaches

Single ontology approaches (Caracciolo, Heguiabehere, Sini, & Keizer, 2009): one global ontology provides a shared vocabulary for the specification of the semantics. All information sources are related to one global ontology. A prominent approach of this kind of ontology integration is SIMS (Arens, Chin Y, Chun-Nsn, & Knoblock, 1993). The global ontology can also be a combination of several specialized ontologies. A reason for the combination of several ontologies can be the modularization of a potentially sizeable monolithic ontology. Ontology representation formalism supports the mixture, i.e., importing other ontology modules (cf.Ontolingua (Farquhar, Fikes, & Rice, 1997)). Single ontology approaches can be applied to integration problems where all information sources to be integrated provide nearly the same view on a domain. However, if one information source has a different perspective on a field, e.g., by providing another level of granularity, requiring the minimal ontology commitment becomes a problematic task, e.g., SIMS (Arens, Chin Y, Chun-Nsn, & Knoblock, 1993).

SIMS (Arens, Chin Y, Chun-Nsn, & Knoblock, 1993): A domain model provides the general terminology for a particular application domain. Each information source is incorporated into SIMS by describing the data supplied by that source regarding the domain model. This model is contained in the mediator. SIMS uses a global domain model that also can be called a global ontology. Then any change in the information sources will affect directly the global ontology (Buccella, Cechich R., & Brisaboa, 2005). This is also the case of DBpedia (described in 3.4)

Multiple ontology approaches

In multiple ontology approaches (Caracciolo, Heguiabehere, Sini, & Keizer, 2009), each information source is represented by its ontology. For example, in Observer (Mena, Kashyap, Sheth, & Illarramendi, 1996) the semantics of an information source is described by a separate ontology. Observer defines a model for dealing with multiple ontologies avoiding problems about integrating global ontologies. The different ontologies (user ontologies) can be described using different vocabularies depending on the user's needs. To solve the problem of heterogeneous vocabularies used to describe the same information, Observer uses "IRM shared repository" which is a catalog of the semantics of the system (Buccella, Cechich R., & Brisaboa, 2005).

Hybrid approaches

Hybrid approaches were developed to overcome the drawbacks of the single or multiple ontology approaches. Similar to multiple ontologies approaches the semantics of each source is described by its ontology. However, to make the source ontologies comparable to each other, they are built upon one global shared vocabulary (Caracciolo, Heguiabehere, Sini, & Keizer, 2009). The shared vocabulary contains basic terms (the primitives) of a domain. To construct complex terms of source ontologies, some operators combine the primitives. Because each term of a source ontology is based on the primitives, the terms become easier comparable than in multiple ontology approaches. Sometimes the shared vocabulary is also an ontology (Stuckenschmidt & Harmelen, 2006). The advantage of a hybrid approach is that new sources can easily be added without the need of modification in the mappings or the shared vocabulary. It also supports the acquisition and evolution of ontologies. The use of a shared vocabulary makes the source ontologies comparable and avoids the disadvantages of multiple ontology approaches.

Examples:

- DOME (Cui & O'Brien, 2000) deals with structured databases and their application programs where the data sources are legacy information systems. The system defines two separate kind of ontologies: resource ontologies, which describe the terminology used by specific information resources; and application ontologies, which represent the vocabulary of a user - group or client - application. Once the ontologies have been described, they are stored in the Ontology Server (Buccella, Cechich R., & Brisaboa, 2005).
- KRAFT (Gray, et al., 1997) also defines two kinds of ontologies: a local ontology and a shared ontology. The difference between DOME and KRAFT is that KRAFT contains a shared ontology. When a modification or addition is made in some source, the local ontology (which represents this source) has to be changed, and the mappings between the local and the shared ontology have to be performed (Buccella, Cechich R., & Brisaboa, 2005).

3.3.2 Inter-ontologies' mapping

Many of the existing information integration systems such as (Mena, Kashyap, Sheth, & Illarramendi, 1996) or (Gray, et al., 1997) use more than one ontology to describe the information. The problem of mapping different ontologies is a well-known problem in knowledge engineering. We defined in section 2.3.2 the alignment operation and its different types. In this section, we present the general approaches using the alignment operation to interconnect ontologies. (Caracciolo, Heguiabehere, Sini, & Keizer, 2009) discussed these approaches that are used in information integration systems.

- Defined mapping: A standard approach to the ontology-mapping problem is to provide the possibility to define mappings. Karma system (Gupta, et al., 2015) uses an approach that allows excellent maneuverability, but it fails to ensure the preservation of semantics: the user is free to define arbitrary mappings even if they do not make sense or produce conflicts.
- Lexical relations: An attempt to provide at least intuitive semantics for mappings between concepts in different ontologies. The approaches extend a common description logic model by quantified inter-ontology relationships borrowed from linguistics. Relationships used are a synonym, hypernym, hyponym, overlap, covering and disjoint. While these relations are similar to constructs used in description logic, they do not have a formal semantics. Consequently, the subsumption algorithm is somewhat heuristic than formally grounded.
- Top-level grounding: To avoid a loss of semantics, one has to stay inside the formal representation language when defining mappings between different ontologies (e.g., DWQ (Calvanese, Giacomo, & Lenzerini, 2002)). A straightforward way to keep inside the formalism is to relate all ontologies used to a single top-level ontology, which can be done by inheriting concepts from a common top-level ontology. While this approach allows establishing connections between concepts from different ontologies regarding common super-classes, it does not create a direct correspondence that might lead to problems when exact matches are required.
- Semantic correspondence: An approach that tries to overcome the ambiguity that arises from an indirect mapping of concepts via a top-level grounding is the attempt to identify well-founded semantic correspondences between concepts from different ontologies. These approaches have to rely on a common vocabulary for defining concepts across different ontologies to avoid arbitrary mappings between concepts,

Alignment based on complementary resources

As per (Staab, 2006), the majority of the approaches use a combination of terminological and structural methods, where the lexical overlap is used to produce an initial mapping, which is subsequently improved by using the structure of source and target. Thus as they assume that (1) sufficient lexical overlap exists between the source and the target ontology, (2) source and target ontology have sufficient structure. In the case of lightly structured ontologies, and ontologies using different vocabularies, these assumptions do not hold (Aleksovski, Klein, Kate, & Harmelen, 2006).

Many works concern the use of complementary knowledge called "background" or support, most often represented in the form of a 3rd ontology (Safar, Reynaud, & Calvier, 2007) to overlap these issues.

Matching ontologies using a background knowledge shows significant results in particular, when the source and the target ontology are of poor semantics, such as flat lists, and where the background knowledge is of rich semantics, providing extensive descriptions of the properties of the concepts involved (Aleksovski, Klein, Kate, & Harmelen, 2006). These background knowledge resources spans from a thesaurus, lexical resources, linked open data, one or several ontologies or a full repository of ontologies. The significant difference between the works is how to select the background knowledge.

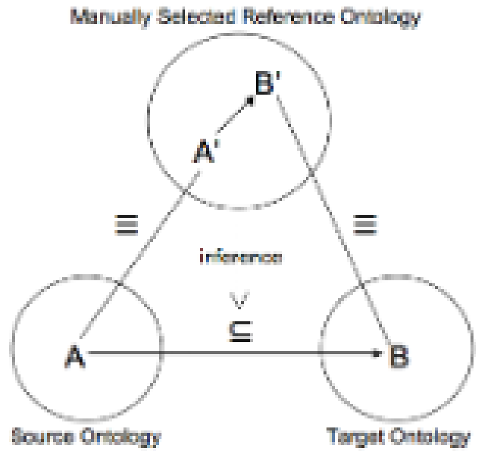

Figure 3.6: Basic approach using a Background ontology (Safar, Reynaud, & Calvier, 2007)

(Safar, Reynaud, & Calvier, 2007) presents a clear general presentation of the works done on this purpose considering that each ontology O comprises only a set of concepts C and a set of relations R between these concepts. The ontology alignment process aims at matching the concepts of one of the ontologies, known as the source ontology (OSrc), with the concepts of another ontology, known as the target ontology (OTar). The process is composed of two main steps as shown in Figure 3.6, Another approach is to rely on a reference domain ontology as a semantic bridge between two ontologies: Terms from the two vocabularies are first mapped to so-called anchor term, and then their mapping is deduced based on the semantic relation of the anchor terms (Sabou, D'Aquin, & Motta, 2006) (Faria, Pesquita, Santos, Cruz, & Couto, 2014).

Ontologies Alignment systems

Every year since 2003, the Ontology Alignment Evaluation Initiative[40] (OAEI) publishes a list of ontology alignment systems participating in the evaluation conference (Ajmi, Ghoula, & Falquet, 2012). Some systems are active and often with the open-source code as LogMap (Jiménez-Ruiz & Grau, 2011) which is a highly scalable ontology matching system with 'built-in' reasoning and diagnosis capabilities. LogMap can deal with semantically rich ontologies containing tens (and even hundreds) of thousands of classes.

[40] http://oaei.ontologymatching.org

Based on the latest result within 2017[41], AML system (Faria, et al., 2013) is one of the best four systems from the ten systems participating this year. AML is a lightweight ontology matching system specialized on the biomedical domain but applicable to any ontologies.

Many types of formalism represent these kinds of alignment resources. Each sort of alignment have its semantics and use specific relations to link resources' entities. The matched entities are specifically of a certain type, for instance, a terminological alignment uses only terminological relations to bridge between terminological entities (Thiéblin, Haemmerlé, Hernandez, & Trojahn, 2018).

rar2	2016			2017		
Tool	Precision	F1-measure	Recall	Precision	F1-measure	Recall
ALIN	0.87	0.4	0.26	0.86	0.41	0.27
AML	0.78	0.69	0.62	0.78	0.69	0.62
LogMap	0.77	0.66	0.57	0.77	0.66	0.57
LogMapLite	0.68	0.56	0.47	0.68	0.56	0.47
XMap	0.8	0.65	0.55	0.78	0.65	0.55

Figure 3.7: OAEI 2016- 2017 evaluation results

Furthermore, the AML system can be considered as an open source toolbox that can easily be integrated and personalized according to the need of ontologies' experts. It also combines a powerful, flexible and extensible framework with a comprehensive user interface that enables alignment visualization and manual editing.

3.3.3 Conclusion

The ontologies present a solution to remedy the structural heterogeneity of the resources of a company, so several ontologies can coexist without being expressed with the same vocabulary. By examining the three ontology-based knowledge integration approaches, we can note that:

- All of them are based on a collection of ontologies and a set of alignments between these ontologies.
- Systems built based on these approaches are using different formalisms to concept their ontologies. They are also using different alignment techniques to better connect ontologies instances.

In the next section (section 3.4), we describe different systems based on semantic web approaches for information integration and some systems in different domains using semantic web for knowledge management systems.

[41] http://oaei.ontologymatching.org/2017/results/index.html

3.4 Semantic organizational knowledge representations

TOVE project (López, 2002), EO (Dietz, 2006) aimed at the development of a set of integrated ontologies for modeling all kinds of enterprises, The REA enterprise ontology (Sedbrook, 2012) is based on elements of the REA (Resource Event-Actor) model (Sicilia & Mora, 2010). The e3-value ontology (Weigand, 2016) was introduced as a tool to help explaining the concepts used to represent e-commerce ideas. The ontology provides concepts for describing economic exchange among partners. Enterprise Ontology (EO) (Dietz, 2006) is a collection of terms and definitions relevant to business enterprises. It was developed as part of the Enterprise Project, with the aim to provide a framework for enterprise modeling. EO is divided into five parts: i) terms related to processes and planning, ii) terms related to organizational structure, iii) terms related to high-level planning, iv) terms relating to marketing and sales and v) terms used to define the terms of the ontology together with terms related to time. Our research focuses on the organizational ontologies, as to unify the semantics used on the ontologies' network; we need to use the same vocabularies.

3.5 Semantic knowledge management systems

This section presents some research projects presenting approaches for creating semantic knowledge management systems. In this section, we describe ontology-based information systems regardless their specific domains to investigate the approaches behind the systems.

In addition to the examples cited in section 3.1.1, (Kim, Manley, & Yang, 2006) proposed the assembly relation model (ARM) using ontologies. An ontology model of the Product Data and Knowledge Management Semantic Object Model has been developed, with the aim of implementing ontology advantages and features into the model (Matsokis, 2010).

(Dartigues, Ghodous, Gruninger, Pallez, & Sriram, 2007) describes an ontological approach to integrate computer-aided design (CAD) and computer aided process planning (CAPP). Also, many knowledge integrations and sharing methods have been developed with ontologies (Staab, 2006); (Jiang & Tan, 2009); (Lee & Suh, 2008)).

These studies prove that ontology-based methods have powerful ability to integrate multiple data resources, facilitate the consideration of the complex relations among concepts, and provide logic reasoning in decision-making. However, few studies focus on providing comprehensive coverage for information integration and sharing within domains of the part, process, and tooling. This lack of consideration makes it impossible not only to integrate and share valuable information effectively but also to satisfy the information requirements of related collaborators.

The semantic knowledge management system OBSA (Gu, Chen, Yang, & Zhang, 2004) manages the integration of XML resources using a globally shared ontology and defining mapping mechanisms based on direct, subsumption composition and decomposition mapping to integrate resources ontologies into the mediator ontology. In order to query the system, the authors use the XML query rewriting approach so the semantic adapter module accepts the query and transforms it to a form that the local information source can take. The local information source executes the query and the semantic adapter transforms the result to the way in which the global information site needed with XSLT technology.

SLACKS (Zhao & Qian, 2017) and ISCPTD (Premkumar, Krishnamurty, Wileden, & Grosse, 2014) are based on a suite of ontologies for a specific domain; they define the mapping between the available ontologies and they are using querying languages or Protégé framework to manage them. SKMSGR (Tello-Leal, Rios-Alvarado, & Diaz-Manriquez, 2015) propose an architecture based on three tiers as per Figure 3.8. The knowledge is extracted using NLP and matched with WordNet, and then the knowledge databases are populated.

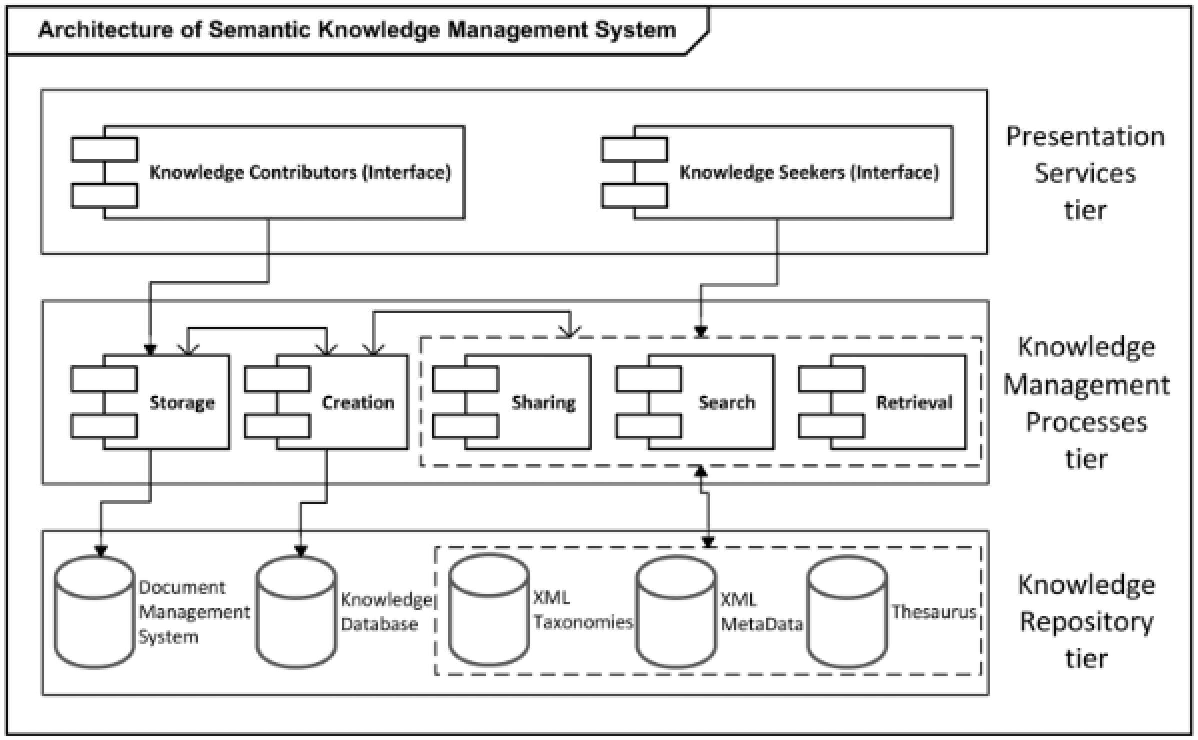

Figure 3.8: SKMSGR (Tello-Leal, Rios-Alvarado, & Diaz-Manriquez, 2015)

DBpedia[42]: DBpedia (Astrova, Korda, & Kalja, 2007) is a crowd-sourced community effort to extract structured content from the information created in various Wikimedia projects. This structured information resembles an open knowledge graph (OKG) which is available for everyone on the Web. The DBpedia Ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used info-boxes within Wikipedia (Pattuelli & Rubinow, 2013).

42 http://wiki.dbpedia.org/

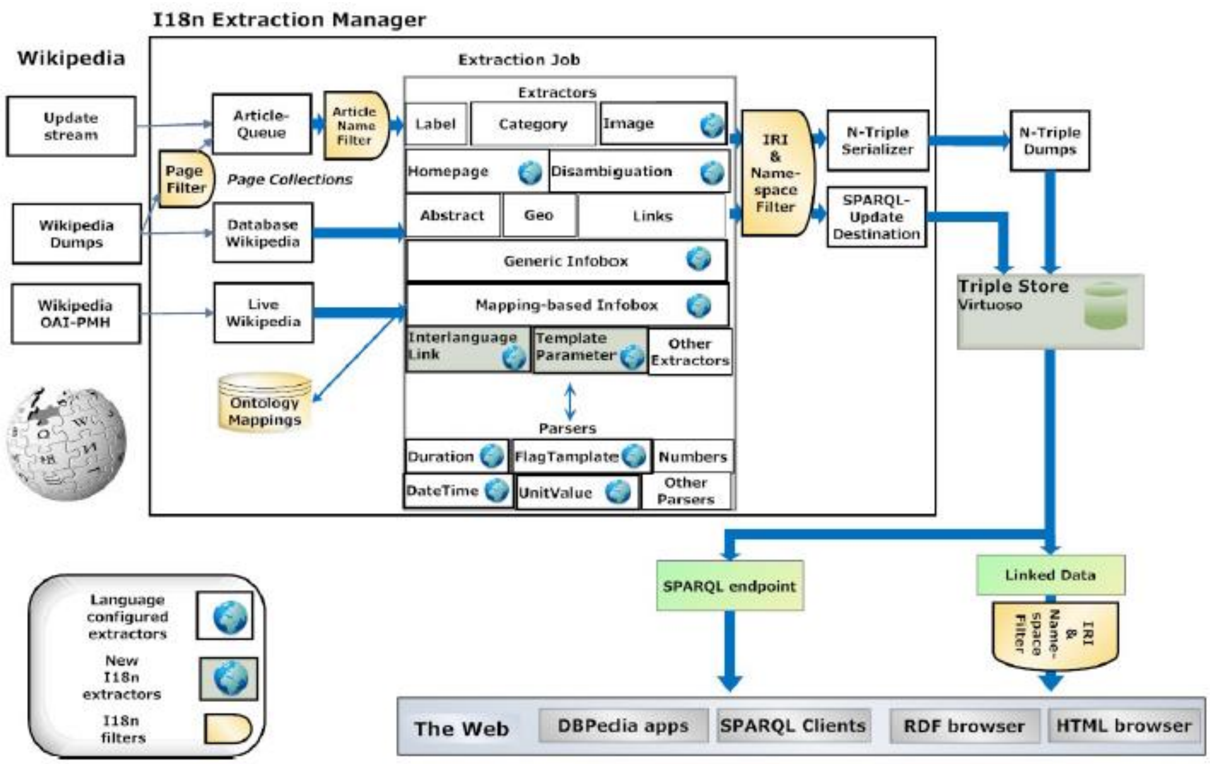

Figure 3.9: DBpedia Extractor

The semantic extraction of information from Wikipedia is accomplished using the DBpedia Information Extraction Framework (DIEF)[43]. The DIEF can process input data from several sources provided by Wikipedia: The Wikipedia dumps are used to produce the static DBpedia versions[44]. In total, 279 language-specific Wikipedia editions exist. An article update stream is used by the DBpedia Live Extraction (Morsey, Lehmann, Auer, Stadler, & Hellmann, 2012) to provide instant updates. As soon as an English Wikipedia article is edited, the extraction is performed on the latest revision, and the DBpedia Live SPARQL endpoint is updated via SPARQL Update queries (Kontokostas, et al., 2012).

3.6 Conclusion

Building an ontology for a specific area or organization is an expensive exercise regarding time and expertise. To simplify this task, several strategies attempt to automate the process by specifying a fixed structure of the knowledge to be extracted to populate an upstream meta-ontology. In this context, The DBpedia Ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used info-boxes within Wikipedia. The main disadvantage of this approach is to limit itself

---

43 https://github.com/dbpedia/extraction-framework/wiki

44 version 3.5.1 in March 2010, 3.6 in October /November 2010 and 3.7 in August 2011

to predefined classes, properties description based on the Wikipedia info-box values that will be difficult to expand once the process is launched, and new concepts added will not be at the same population level as the old ones. Other works describe and implement strategies to represent non-ontological knowledge resources as ontological resources. Researches are being done to unify organizational vocabularies. But as far as we know, there is no complete solution or an approach that specify the process of extracting knowledge from different resources formats to a definition how we can best access this knowledge. The next chapter will be dedicated to the state of the art of strategies and system to access knowledge. It is the same problem as for representing knowledge approaches to access knowledge usually defined depending on the nature of resources formats (Databases, files, text...).

Chapitre 4. Knowledge access

4.1 Introduction

As seen in chapter 2, in a domain as an organization, the knowledge exists in different formats and different types. Approaches and systems are presented and developed to encode best and store the organizational knowledge. The object is to access it better, reuse it and reason on it to find the answer to questions such as “where can we find resources related to a precise question?” Querying relational databases using schema-cognizant languages like SQL and querying document collections by typing arbitrary keywords are the extreme ends of the continuum between structured and unstructured data access (Amudaria & Sasirekha, 2011). Thus, many approaches have been provided to avoid querying languages to the non-expert user.

In this chapter, non-semantic and semantic approaches are proposed to “better” access to knowledge that can be stored and managed in different ways depending on the architecture and the capabilities of the system.

.

4.2 Overview on knowledge access approaches

4.2.1 Indexing and keyword search

Keyword research has been developed since the 1960s to search for information in the body of texts (Tsoukas, 2005). It consists of representing the texts indexing and queries according to a formal model (vector model, logic, language, latent semantic, etc.) and defining a function of pairing between texts and queries (e.g., cosine calculation in the case of the model vector). The documents then constitute the answer with the best matching (or relevance value) with the request. These search techniques have been adapted to hypertext to create search engines on the web. Other adaptations were made to index and search by keywords in the databases.

Keyword search engines are developed in several ways, It can apply a Boolean search (e.g., red AND/OR car) or phrase search (“red car”) and it results in a ranked list using as criteria the number of occurrences of keywords in the document, the popularity of the document or as Google using a pre-computed ranking algorithm Teoma a ranked list of topic-specific pages (Choudhary & Burdak, 2012). If we use the keyword search in a corpus of text, we need to predict the exact words used by the writer, in indexed databases, no matter what terms are used in the document, the result will be all document indexed under the subject term.

The best-known system in this category is Swoogle[45] It is therefore possible to index all the knowledge resources of an organization and to carry out research by keywords on this corpus. Several existing systems (e.g., Exalead[46], Sinequa[47], Verticrawl[48]: with 220 file formats of different kinds).

4.2.2 Query Answering (QA) approach

The goal of QA systems, as defined by (Hirschman & Gaizauskas, 2001), is to allow users to ask questions in Natural Language (NL), using their terminology, and receive a concise answer. This approach first emerged in the 1980s encompassing data mining, data prospecting, and extraction of knowledge from data (Cur & Blin, 2014). It is a family of statistical methods for dealing with a vast number of data and identifying the most interesting aspects of the structure of these data. Some techniques help to highlight the relationships that may exist between the different data and to derive statistical information from them, which allows a more succinct description of the main information contained in the data. The goal of QA systems, as defined by (Hirschman & Gaizauskas, 2001), is to enable users to ask questions in NL, using their terminology, and receive a concise answer. There are several QA systems; the first is BASEBALL (Green, Wolf, Chomsky, & Laughery, 1961) and LUNAR (Minaee & Liu, 2017) until arriving at Watson[49]from IBM. Each year dozens of systems are evaluated in conferences (e.g., NTCIR[50], TREC[51] or CLEF[52]) which are natural language interfaces to expert systems, tailored to specific domains. This approach facilitates the querying of database knowledge and the response to intelligent queries in database systems (Keng, 2011). According to (Ajami & Bagheri-Tadi, 2013), for other resource formats, it seems that, because of the quality of data in specific areas (and despite the industry's interoperability efforts), these systems are not performing as well.

4.2.3 Search approaches for distributed resources

Heterogeneity in databases can be manifested in different ways; they may run on different hardware, use different network protocols, have different software to manage their data stores and have different data models. Two major approaches have been presented to remedy these difficulties.

The database federation approach was presented several years back by first trying to solve the problem of interoperability. As detailed in (Parent & Spaccapietra, 2000), through two approaches, one at the system architecture level and another at the intermediate level, the user is provided with a multi-database access language (SQL) but without considering the unification of the semantics of data from different sources that

[45] https://www.w3.org/2001/sw/wiki/Swoogle

[46] https://www.exalead.com/search/

[47] https://www.sinequa.com/

[48] http://www.verticrawl.com/

[49] https://www.ibm.com/watson/services/personality-insights/

[50] http://research.nii.ac.jp/ntcir/index-en.html

[51] https://trec.nist.gov/

[52] http://www.clef-initiative.eu/

is the case of federated database systems. The advantage of this approach is to give a unified view of a distributed system and the impression of having a centralized database.

IBM has made a significant investment in federation technologies that have resulted in a set of federation capabilities throw a variety of IBM products including InfoSpher[53], DB2®[54], InfoSphere Warehouse[55], and IBM Enterprise Information Portal (EIP)[56]. IBM's product is based on a technology created by IBM's Almaden Research Center: "Garlic"[57] that is a mediator. SAP created NetWeaver platform[58] to federate distributed portal installations both SAP and non-SAP to provide single portal access.

Mediator-type approaches have emerged to provide more flexibility. These are mainly two types of strategies, schema mediation that is a direct extension of the federated approach, and the mediation of contexts based on semantic distance and unification of backgrounds (Goh, Madnick, & Siegel, 1994). Taking the example of Garlic, it acts as a mediator that searches for information from different data sources. After finding all the requested information, it brings it together and presents it to the requester in a unified way, as if it all came from a single source of data. Which is based on a query processor that gives the user a unified view of all the data gathered by the system? It connects to the data sources via a wrapper. It consists of a set of interfaces, for such tasks as describing data, invoking operations on data, planning searches that tell the wrapper what part or parts of a query it can handle, and executing searches, which tell the wrapper the data source native language.

While database integration approaches have reached a certain maturity, it should be noted that they apply, of course, only to data represented in databases and therefore do not deal with other resources such as texts and hypertexts, or spreadsheets. Even though the researchers explained that their work dealt with all types of resources, we can figure it out from the examples they gave that these resources must be stored in databases in order to be retrieved.

4.3 Ontology-based search

Semantic search based on ontologies enables to define concepts and relations representing knowledge in specific domains, which make search engines more intelligent by adding meaning and structure to search process. Recently, some ontology-based searches have been published, even that their application area and its realizations are different; their based common goal is mostly the same. Since accessing semantic knowledge needs ontologists and specialists of query languages syntax, an easy to use interface is required. To solve the problem, many solutions have been described:

[53] http://www-03.ibm.com/software/products/fr/ibminfofedeserv

[54] https://www.ibm.com/analytics/us/en/db2/trials/

[55] https://www.ibm.com/support/knowledgecenter/en/SSEPGG_9.5.0/com.ibm.dwe.welcome.doc/dwev9welcome.html

[56] ftp://ftp.software.ibm.com/software/partners/pdf/IBMEnterpriseInformationPortal.pdf

[57] http://www.almaden.ibm.com/cs/garlic/

[58] https://www.sap.com/products/netweaver-platform.html

4.3.1 Form-based approach

These systems provide a user interface (e.g., web form) that allow users to specify a query associated with a concept, property or values in a semantic knowledge base. This approach is much related to faceted navigation (Hearst & Stoica, 2009), like taxonomic navigation, guides users by showing them available categories (or facets), but does not require them to browse through hierarchies that may not reflect their search intentions or map well to how they want to explore the data. The output of the selection or the analyses may be displayed as tables, charts, or summaries of aggregated data such as TreeMaps (Shneiderman, 1992) which may be updated dynamically based on user's manipulation of provided control mechanisms (Ahlberg & Shneiderman, 1994). Others searchers may also seek answers from other searchers either as part of a collaborative search task (Morris & Factors, 2007) or in question-and-answer (QA) scenarios where questions are posed broadly to online communities such as Yahoo! Answers[59]or a group of knowledgeable (expert) answers via a more focused medium such as instant messaging (Horowitz & Kamvar, 2010).

The Shoe search[60] engine is an example of a form based engine search, which offers to the user concepts from a backend ontology. This approach is easy to implement but users cannot use their vocabulary to refine the search. As in Figure 4.1, a Google Custom search[61] has been added to the Uwe as Shoezam website cannot provide it.

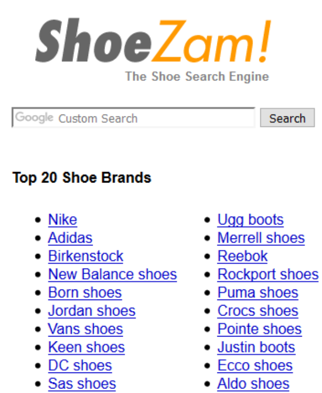

Figure 4.1: ShoeZam! an application of Shoe Search Engine

4.3.2 Semantic keyword search

This semantic search approach enhances the performance of keyword search technique by transforming a keyword into a formal ontology-based query, e.g., in RDQL[62], RQL (Chardin, Coquery, Pailloux, & Petit,

59 answers.yahoo.com

60 http://www.shoezam.com/

61 https://cse.google.com/cse/

62 https://www.w3.org/Submission/RDQL/

2014), SPARQL[63], etc. automatically. The benefits of semantic keyword search systems are that they provide an easy search interface that users are familiar with by hiding the ontology structure and the complexity of the formal semantic query from users.

Semsearch[64] proposes the use of predefined query templates to construct formal queries. The models are a combination of all possible entity types from the knowledge base. The indexed semantic entities, including classes, properties and instances are constructed to support the mapping of keywords to semantic entities. The indexing input is the RDF data with the ontology schema. The query templates and then the indexed semantic engine are computed in the pre-processing time. At the run-time, each keyword term is mapped onto entities. All the mapped entities then are matched to the templates to construct the formal query.

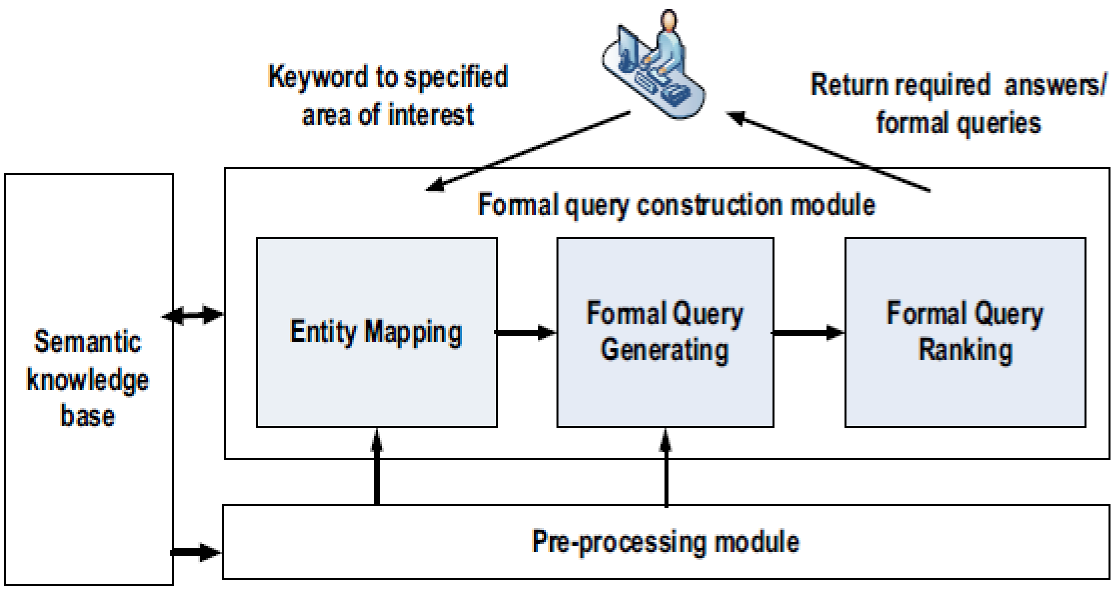

Figure 4.2 The standard architecture of semantic keyword search

Quick (Rosé & Hall, 2004) proposes the same process to build the predefined query. It starts with a user entering keywords; then a new interface is provided for supporting users to produce a semantic query gradually, which is not suitable for users with no knowledge about graphs, ontology concepts and relations.

Spark (Zhou, Wang, Xiong, Wang, & Yu, 2007) is based on adapting keywords to querying the semantic web which automatically translates the keyword queries into formal logic queries using a string comparison techniques and semantic mapping with WordNet[65] so that end users can use keywords to perform a semantic search. A probabilistic query-ranking model is proposed to select the most likely SPARQL query. A minimum spanning tree algorithm is used to construct the query graph for each query set.

63 https://www.w3.org/TR/rdf-sparql-query/

64 http://technologies.kmi.open.ac.uk/semsearch/

65 https://wordnet.princeton.edu/

(Tran, Cimiano, Rudolph, & Studer, 2007) suggests another approach for interpreting a word query using a semantic knowledge base. This approach uses a transversal graph algorithm to construct a query graph. After a user has entered the keywords with corresponding entities, the query graph is built by traversing the RDF knowledge base and finding the neighboring entities of each mapped entity within a limited range. Possible sub-graphs are then extracted from the whole graph and transformed into formal semantic queries that will be presented to the users to choose the one that fits its needs.

Q2semantic (Wang, Zhang, Liu, Tran, & Yu, 2008) propose a different approach as it supports keyword search on schema-less RDF data graphs and keyword search that do not include an initial graph. The method uses the Clustered Rack Graph (Bechhofer, 2008) technique to infer the ontology structure. An algorithm is then used to generate top-k query graphs by exploring the ontology structure. The top-k query graph will be transformed to a list of top-k SPARQL queries, which allow selecting the suitable one for the user.

SKengine (Supnithi, Yamaguchi, Pan, Wuwongse, & Buranarach, 2015) is designed for expert discovery tasks also using a graph-based technique to construct formal queries. The pre-processing module consists of indexing entities to a keyword and relationships between the ontology classes. The query graphs will be built from the mapped entities to produce all possible query graphs. To fix the root note, the approach uses the fixed root node algorithm to avoid distinct rood nodes as it may generate irrelevant roots that are not related to the expert concept. The process ends with ranked possible graphs following a selection of criteria. The highest ranked graph will be then transformed into a SPARQL query.

All the above approaches describe ways to deal with transforming keywords into formal querying languages queries and then propose several ways in order for the user to choose the suitable one to query the knowledge base. Another approach has been described to detect users profile and then introduce suggestions from an interest score assigned to ontology domain concepts. The method (Sieg, Mobasher, & Burke, 2007) proposes a spreading activation algorithm for maintaining the interest score in the user profile based on user ongoing behavior.

The approach presented in (Tran, Cimiano, Rudolph, & Studer, 2007) is based on translating keywords queries to Description Logic (DL) conjunctive queries using background knowledge available in ontologies. The keywords are processed by Lucerne search engine. The exploration process takes the ontology entities returned by the engine. The output of this approach is a visualization of sub-graphs connecting matched entities and highlighting entities matched with the keywords.

The IBM China Research Lab (Zhong, Zhu, Li, & Yu, 2002) proposes a conceptual graph matching approach that defines the semantic similarity between concepts, relations and conceptual graph. It uses a matching algorithm to calculate similarities between a resource conceptual graph and a conceptual query graph.

```
1    get user query and the central word set by user.
2    parse the query and generate query CG using ALPHA.
3    get the entry of query graph E and locate it in WordNet
4    for (each resource CG indexed by E and its sub-concept in the
            domain ontology)
5    {    // the beginning of the recursive process
6     calculate the similarity between entry pair
7     For (each relation directly associated with entry in query CG)
8     {
9      For (each relation directly associated with entry in
              resource CG)
10       {
11        calculate the similarity between these two relations and
          calculate the similarity between two subgraphs induced by
          the two relations recursively(line 5 to 15). Each time,
          The concept in current subgraph associated with the
          relation which induces the subgraph will serve as the
          entry of the subgraph.
12       }
13    }
14   find the best match from the above combinations of subgraphs
     using Bellman-Ford algorithm and sum up the similarity between
     entry pair and the similarity between each subgraph pair
     according to their respective weights as the similarity between
     the resource CG to the query CG
15   }    // the end of the recursive process
16   Rank the results and return answers back to user in proper order
```

Figure 4.3: Conceptual graph matching algorithm (Zhong, Zhu, Li, & Yu, 2002)

As the approach is based on a maximum subgraph matching, which is an NP-complete problem, many assumptions have been taken to avoid cycle in the graph that will lead to an unending recursion and will be fatal to this algorithm. Furthermore, as the query graph is based on ALPHA (Li, Zhang, & Yu, 2001), the user keyword need to have a centered word that will be considered as the root node for the query-graph.

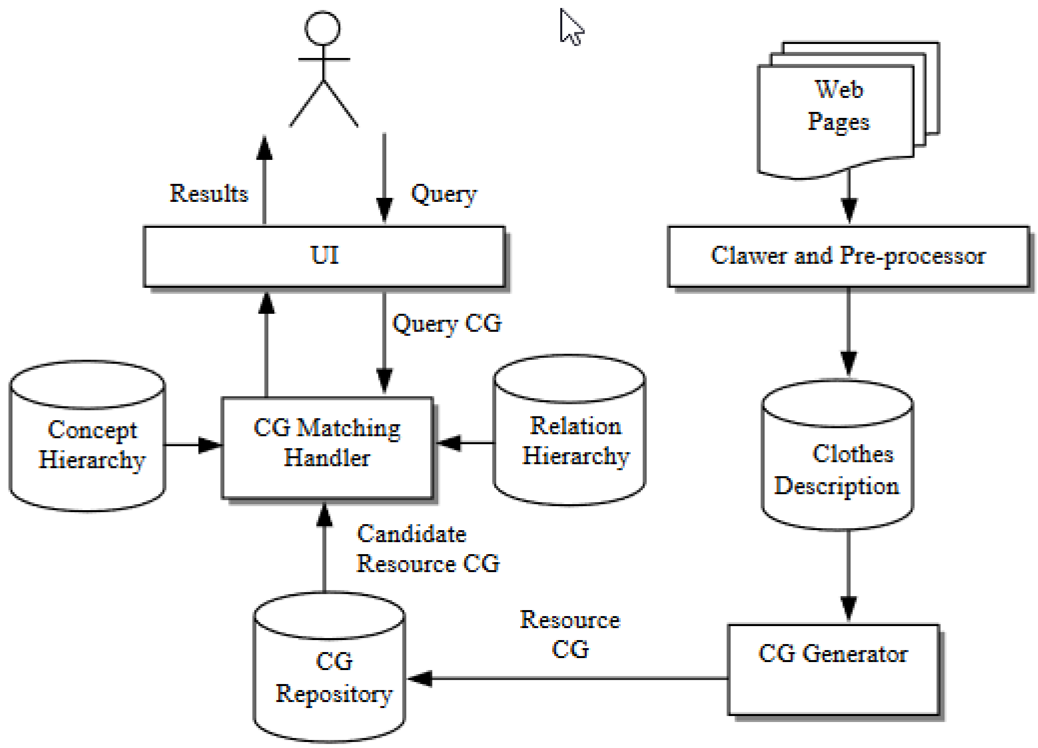

Figure 4.4: Overview of the whole architecture of search engine (Zhong, Zhu, Li, & Yu, 2002)

This approach has been proposed in 2002 and with the progress in the field of ontologies alignments, the idea can lead to the best performance approach.

4.3.3 Semantic Question Answering (QA)

As with semantic keywords search, researchers in semantic QA try to hide the complexity of semantic querying language using several methods,

QACID (Bondiombouy & Valduriez, 2016) proposes preparing a collection of Natural Language (NL) queries for a given domain and their correspondence in SPARQL. Therefore, when the user chooses an NL query, he is selecting a SPARQL query translated in a comprehensive language for him. ORAKEL (Cimiano, Haase, Heizmann, Mantel, & Studer, 2008), which is an NL interface, translates factual wh-queries (who, where, what...) into F-Logic or SPARQL and evaluates them concerning a given knowledge base (Bouziane, Bouchiha, Doumi, & Malki, 2015).

AquaLog interface (Lopez, Uren, Sabou, & Motta, Is Question Answering fit for the Semantic Web?, 2011) allows the user to choose an ontology, and then NL queries have to respect its vocabulary. It uses JAPE [66]

[66] https://gate.ac.uk/sale/tao/splitch8.html

grammar to identify a grammatical instance to convert the NL query to a triple query. All of these approaches use external lexical resources such as WordNet.

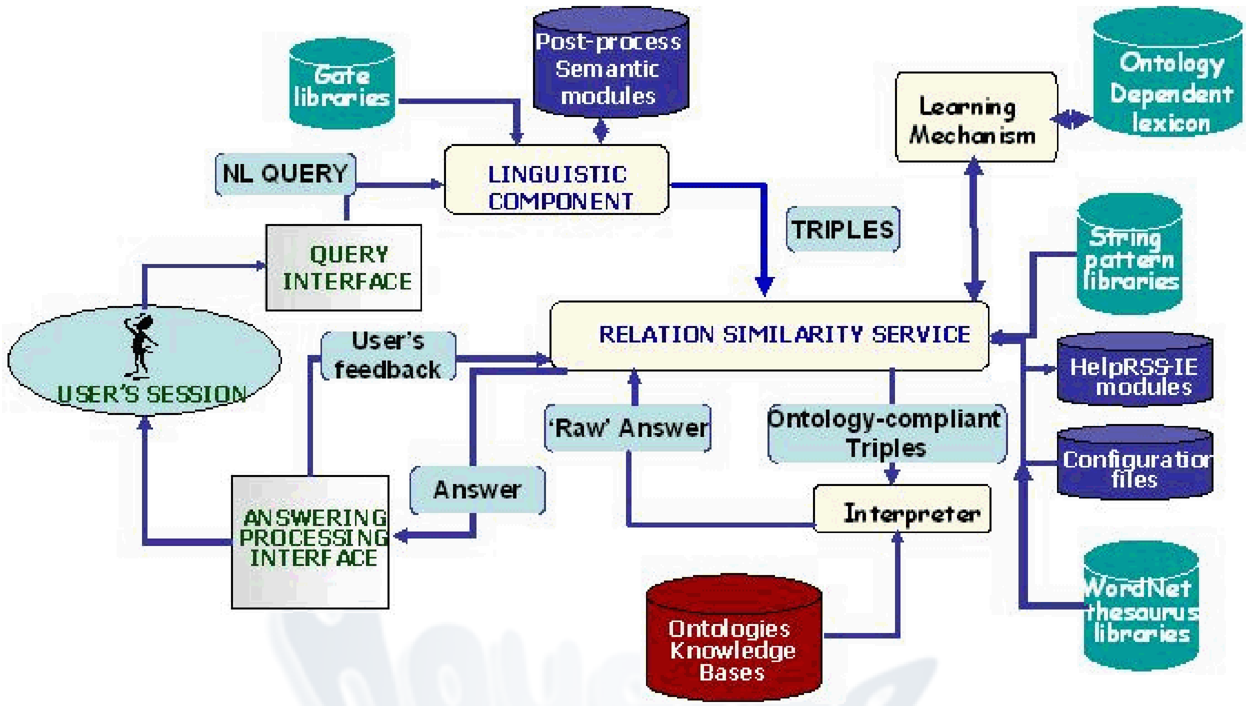

Figure 4.5: AquaLog process architecture (Lopez, Uren, Sabou, & Motta, 2011)

The main challenge in the development of the current version of AquaLog (Lopez, Uren, Sabou, & Motta, Is Question Answering fit for the Semantic Web?, 2011) is to deal with complex NL queries efficiently. Thus, the system is supported by user's feedback to improve its results as well as the FeryA system (Damljanovic, Agatonovic, & Cunningham, 2010).

Figure 4.6 presents a summary about the technics presented in this section that shows that many techniques are used to interpret best the user's needs by trying to automatically transform their question to a semantic query with preservation of the original meaning and their success rate.

Ontology-based QA systems	Subset of NL			Customization		Ontology-independent		
	Guided NL	Bag of words	Full shallow grammar	Domain grammar / collection	Domain lexicons	User learning	Relation (Triple) based	Pattern-matching (structural lexicon)
QACID		+		+	+			
ORAKEL			+	+	+			
e-Librarian			+		+			
GINSENG	+							+
NLPReduce		+						+
Querix			+				+	+
AquaLog			+			+	+	(entity lexicon only)
PANTO			+				+	+
QuestIO		+					+	+ (gazetteers)
FreyA			+			+	+	+ (gazetters)

Figure 4.6: Performance results of some ontology-based question answering systems (Bouziane, Bouchiha, Doumi, & Malki, 2015)

4.3.4 Semantic approaches for distributed ontologies

Federation and mediation approaches used for distributed databases have been reused for distributed ontologies. Many systems have been developed using a different strategy to find a solution to two main problems. The first one is to hide semantic complexity for the user, and this is much managed over query rewriting as well as as for question answering and keyword search. The second problem that is specific to distributed resources is to the ability to deal with the heterogeneity of the ontological resources.

Observer goal (Mena, Kashyap, Sheth, & Illarramendi, 1996) is also to find an answer to an ontology query over heterogeneous ontologies. In observer, IRM (Inter-Ontology Relationship Manager) manages the semantic mapping among different ontologies for query rewriting. However, the scope of the semantic mapping is restricted to the equality among concepts (i.e., synonym relationship) and it is based on query rewriting. As a follow-up, Observer (Mena, Kashyap, Sheth, & Illarramendi, 1996) considers the query rewriting using similar concepts (i.e., hypernyms or hyponyms) when there is no synonym relationship between ontologies. Such a rewriting cannot preserve the semantics of the original query. Thus, (Mena, Kashyap, Sheth, & Illarramendi, 1996) addresses the information loss of the retrieved query results caused by such a rewriting. Observer proposes a set of ontologies from where the user can select the vocabulary that he want to use for querying the other ontologies. Then, the system rewrites all the queries based on it. In our research, we do not consider query rewriting and we propose a new approach not based on SPARQL querying language, but we rely on ontologies matching to find knowledge resources matched through the concepts announced via the user query as the semantic mappings preserve the semantics of the original query.

Other researches focusing on query processing over distributed ontologies have been performed. (Ensan & Bagheri, 2010) suggests a global data summary for locating data matching query answers in different sources and the query optimization. However, (Ensan & Bagheri, 2010) assumes that all distributed ontologies can be accessed in a uniform way like a global schema. In other words, the heterogeneity of schema of the distributed ontologies is not considered. Besides, many tasks are concentrated on the mediator. As well as query scheduling, the merge (i.e., join) of all local query results is also executed in the mediator. Thus, when the mediator receives requests for many queries at the same time, the bottleneck on the mediator is inevitable.

Most researches on query answering over distributed ontologies are based on the P2P architecture. Edutella (Nejdl, 2002) uses an unstructured P2P network, which has no method to route a query to the relevant ontologies. Instead, the query is broadcasted to the entire network. Thus, a considerable amount of unnecessary network traffic incurs. As a successor of Edutella, to provide better scalability, (Nejdl, 2002) presents a schema-based query routing strategy in a hierarchical topology using the super-peer concept. (Nejdl, 2002) also suggests a rule-based mediation between two different schemas to collect results from many peers using heterogeneous schemas. SOMERDF (Adjiman, Goasdoué, & Rousset, 2007) supports the semantic mapping between two atomic concepts and between the domain (or range) of a property and a class. Piazza (Halevy, et al., 2005) proposes a language (heavily relying on XQuery/XPath) to describe the semantic mapping between two different ontologies. In these works, for distributed query answering, a peer reformulates a query by using the semantic mapping and forwards the reformulated query to another peer related by the semantic mapping. Drago (Serafini & Tamilin, 2005) focuses on a distributed reasoning based on the P2P-like architecture. In Drago, every peer maintains a set of ontologies and the semantic mapping between its local ontologies and remote ontologies located in other peers. A reasoning service is performed by a local reasoner for the locally registered ontologies, and the reasoning is propagated to the other peers when the local ontologies are semantically connected to the other remote ontologies. The semantic mapping supported in Dragon is only the subsumption relationship between two atomic concepts. Besides, it does not support the ABox reasoning. Kanop2p (Wang, et al., 2007) also suggests the P2P-like architecture for query answering over distributed ontologies. Kanop2p supports more extensive semantic mapping which describes the correspondence between views of two different ontologies, where a conjunctive query represents each view. For the distributed query answering, it generates a virtual ontology including a target ontology to which the query is issued and the semantic mapping between the target and the other ontologies. Then, the query evaluation is performed against the virtual ontology. Several recent studies based on the P2P architecture also consider the semantic mapping between two different ontologies. However, the scope of the semantic mapping is still not sufficient for the distributed ontology environment. They do not consider the semantic mapping associated with more than two ontologies

By exploring different system dealing with different architecture of distributed ontologies systems, we can conclude that rewriting user queries to fit resources formats, languages and models is the main strategy deployed by the methods.

4.4 Systems evaluation's approaches

Traditionally in information retrieval (IR), there has been a strong focus on measuring system effectiveness: the ability of an information retrieval system to discriminate between documents that are relevant or not relevant for a given user query (Crestani, 1995). The efficiency can be measured with the metrics such as precision, recall, precision, F-measure and mean average precision. The value of precision and recall lies between 0 and 1, and the maximum value is 1.

The efficiency of an information retrieval system has also been assessed, e.g., measuring how long the system takes to return results or memory/disk space required to store the index. Measuring the effectiveness and efficiency of an information retrieval system has commonly been conducted in a laboratory setting, with little involvement of end users and focused on assessing the performance of the underlying search algorithms; therefore, this is commonly referred to as system-oriented evaluation (Ioannidis, et al., 2016).

(Peltonen & Kaski, 2011) explains that the user object has to be clear to optimize precision and recall results. Thus, because of its emphasis on the role of human intelligence in the search process, IR requires different evaluation models. One example of this is (Koenemann & Belkin, 1996), where the authors examine different levels of transparency in query suggestion. Other IR researches, such as Borland's IIR evaluation model, applies a methodology more reminiscent of human-computer interaction, focusing on crucial issues such as the characteristics of users and the details of experimental design (Borlund, 2003). Other research on the evaluation of IR systems has shifted the focus toward the measurement of non-traditional, but critical, aspects of the search process such as user engagement (O'Brien & Toms, 2010), search satisfaction (Fox, Karnawat, Mydland, Dumais, & White, 2005), time-based gain (Smucker & Clarke, 2012), learning as a function of exploration time (Rao & Card, 1994), or the rate of grasping variable properties or inter-variable relationships during exploratory data analysis (Pirolli & Rao, 1996).

In practice, it is common to utilize various evaluation approaches that will be used throughout the development of an information retrieval system. From using test collections to develop, contrast and optimize search algorithms; to conduct laboratory-based user experiments for improving the design of the user interface; to evaluate carried out in situation as the information retrieval system is used in a practical setting (Clough, 2015).

One of the most used methodologies for conducting experiments that can be repeated and performed in a controlled laboratory-based setting is test collection-based evaluation (Robertson, 2008), (Sanderson, 2010), (Järvelin, 2011),[(Harman, 2011)). It is often referred to the Cranfield approach or methodology (Rowley & Hartley, 2017). The approach uses test collections: re-useable and standardized resources that can be used to evaluate information retrieval systems concerning the system. The main components of an information retrieval test collection are the document collection, topics, and relevance assessments. These, together with evaluation measures, simulate the users of a search system in an operational setting and enable the effectiveness of an information retrieval system to be quantified. Evaluating information retrieval systems in this manner allows for the comparison of different search algorithms and the effects of altering algorithm parameters to be systematically observed and quantified.

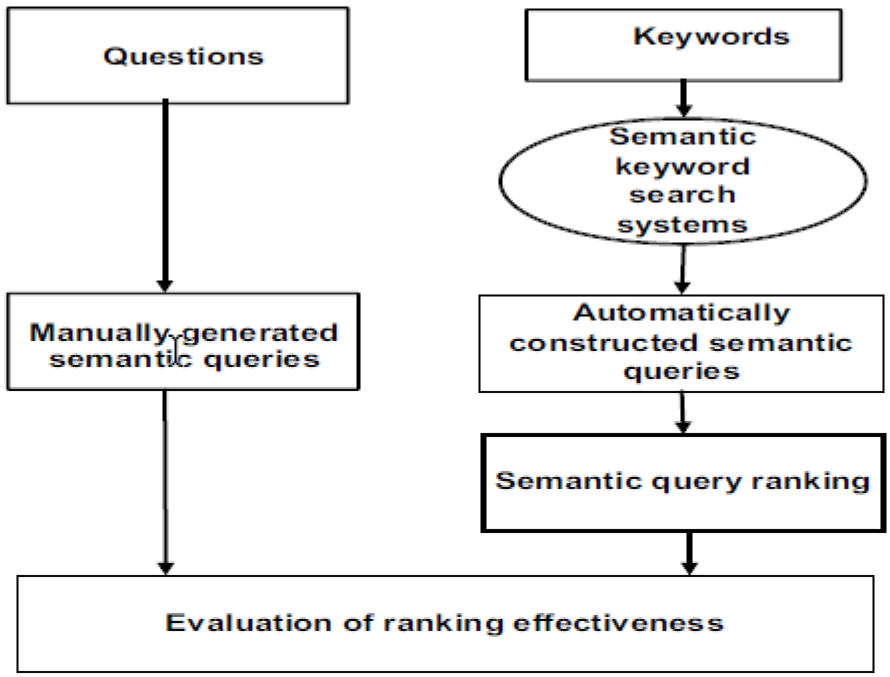

Figure 4.7: Methods of ranking effectiveness evaluation

For example, to evaluate keyword approaches, a set of keyword phrases (i.e., the input of systems) is provided. Keyword phrases may be extracted from problem scenarios or questions presented by experts/participants in the ontology domains. The manual semantic queries corresponding to the questions is created as a golden standard. The constructed semantic queries from systems are evaluated with the golden-standard manual queries. If the constructed semantic query is semantically equivalent to the golden standard query then it is ”acceptable”. The order of the acceptable query in the ranked list is for evaluating the capability of ranking component.

4.5 Conclusion

After describing the basic concept behind different querying approach, we present how the introduction of semantic technologies contributes to better discover knowledge within a knowledge base based on different approaches. Semantic technologies make a simple search engine smarter by introducing the meaning and context to a word used to be After having described the basic concepts behind different querying approaches, we present how the introduction of semantic technologies contribute to better discover knowledge within a knowledge base based on different approaches. Semantic technologies make a simple search engine smarter by introducing the meaning and context to a word used to be managed as a set of characters. A common example is when we search for the word “mouse,” and the traditional search engine cannot understand if we are looking for the “computer mouse” or “the animal mouse” or “Mickey Mouse.” Figure 4.8 presents the example of a keyword search using DuckDuckGo[67]search engine and Google search engine[68]for the word “FAST,” which is the electronic fuel injection leader. We can find the relevant answer in the semantic search engine at the five answer, but all the first pages of google are representing fast definition as the opposite of “slow.”

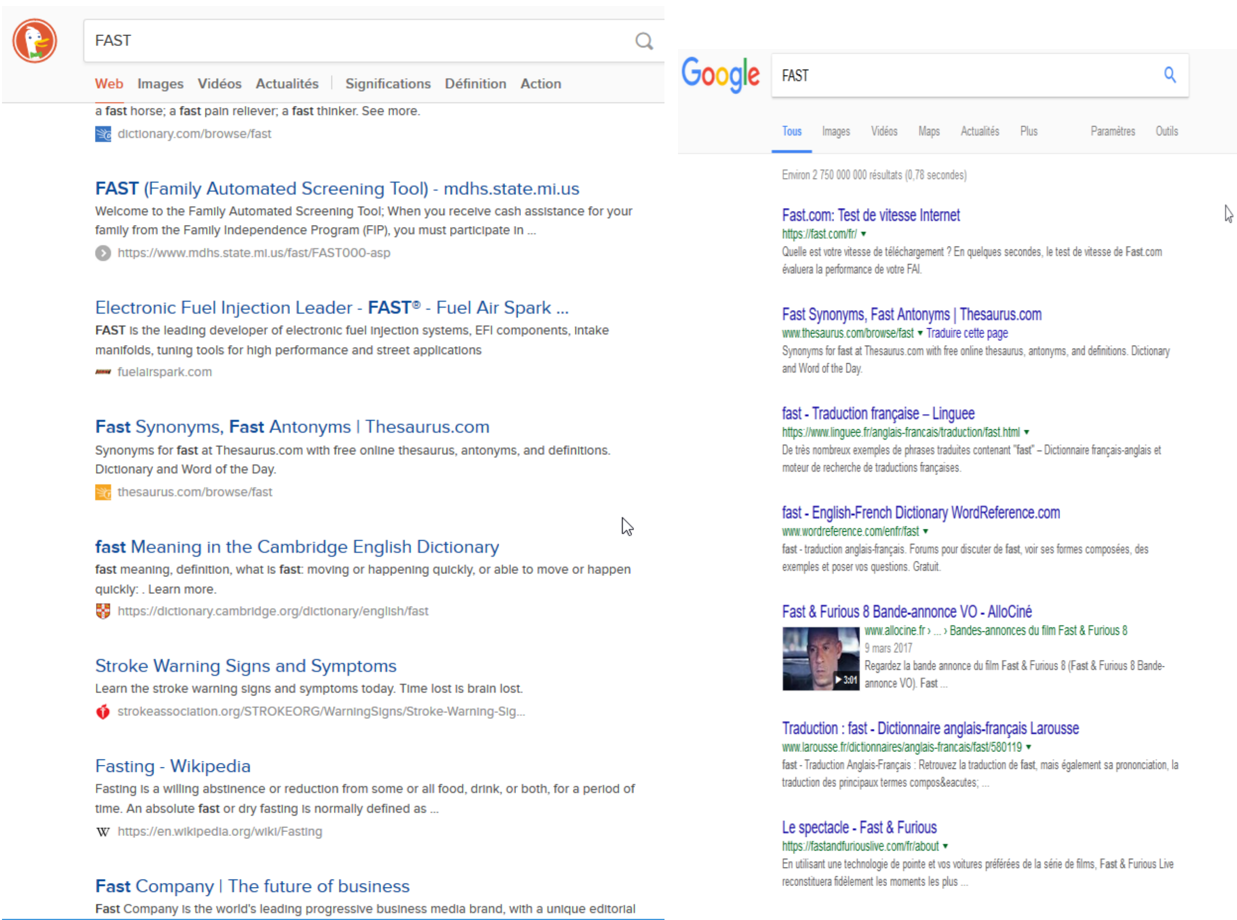

Figure 4.8: Results for the query “FAST” with a semantic and nonsemantic search engine

[67] https://duckduckgo.com/

[68] https://www.google.com/search

Section 4.3 was dedicated to details different approaches of how to enhance a search engine with semantics while hiding the semantic query languages difficulties. For semantic keyword search and semantic QA search, we observe that the primary strategy is to rewrite keywords, user-questions, and predefined questions automatically with semantic query languages. Query re-writing approaches are cumbersome to achieve, as they need many steps to be done in the run-time that is why many approaches suggest a predefined query, which is not the optimal solution. We find the idea behind the graph matching approach very interesting, as matching ontologies using the actual matching systems lead to interesting results. The next part of this thesis is dedicated to describing our approach. This approach manages organizational knowledge over a network of ontologies and reuses the arcs of the network of ontologies, which are the result of alignments, to find the knowledge represented over the network's nodes.

Part II. Our approach

Chapitre 5. Organizational conceptual schema: ontologies' network-based approach

Our work does not aim to represent all the content of the sources in an integrated kind of knowledge base, but to create only a representation of each resource.

The representation of the resource is composed of a metadata representation and a structure representation. In the case of a website, for example, the description should include the entity types represented in the site and their relationships (e.g., Teacher, Course, Subject, Teacher-Teach-Course, etc.), in fact, an ontology of the website.

We refer to the resource representation as an "ontology," as defined on section 2.3.1 "Ontologies are explicit, formal specification of terms in the domain and the relations among them" (Gruber, 1993). The terms and relations, together with metadata and document structure form a collection of a specific domain. It is composed of feature classes represented in the knowledge resource and their relationship, which is described as a set of triples. The set of ontologies will then be connected in a network of ontologies. To achieve the construction of the ontological network, we described our approach to represent metadata and knowledge resource structures in a way that enables tracing the concepts in their original resources.

Our approach is intended to build a conceptual knowledge network of a domain or an organization through representations of its explicit knowledge resources, as well as a generic description of the field that we call the "Profile Ontology." This plays a central role in the construction of the network.

With the term "conceptual knowledge network" we refer to the information extracted from the explicit resources that represent the nodes of the graph (concepts) linked by semantic relations representing the arcs.

Our model makes it possible to represent the explicit resources, as well as the entities, via a representation of their metadata and structure. This allows linking them to both original resources and the progress of knowledge. Bindings are modeled through a management operator that defines the alignment operation, the linked entities, and the semantic similarity measures between related entities. The meta-model of the network is then composed of linked entities, as well as the results generated by the link manager.

In this chapter, we describe our approach for creating an organizational, conceptual schema. In the first part, we describe the abstract model, which constitutes the main thrust of our strategy. In the second part, based on our model, we present our approach to building a network of a conceptual representation of knowledge resources.

5.1 Ontologies' network model

Our model aims to represent the domain resources and the semantic connection between them. To do so, we need to describe the resources

1. form an entities level to specify the connected concepts between two specific resources
2. from a resource level to be able to identify if two particular resources are connected or not.

After identifying and categorized the resources and investigating their structure, we followed an iterative process by trying different abstract models, testing their ability to represent the connection between the resources and how can we best use the model to reuse the connections to identify the semantic links between the resources.

To adequately describe and use the resource representations we need to define their life cycle within the network (see Figure 5.1):

Resource_representations (RR): is a class representing the entities extracted from a knowledge resource. This model represents the meta-data and the resource structure (described in section 5.2)

Network_Management_Operations (NMO): is a class representing the results of the set of operations done to match resources entities. For each operation, it takes as input two resource_representations (RR) and produces as output a collection of the entities semantically similar. This output will be saved via the Network_Manager (NM).

Network_Manager (NM): Is the class representing all the sets of alignment results done by the (NMO).

Ontology_Network (ON): Is the class representing all the resources within the network and all the connections between them.

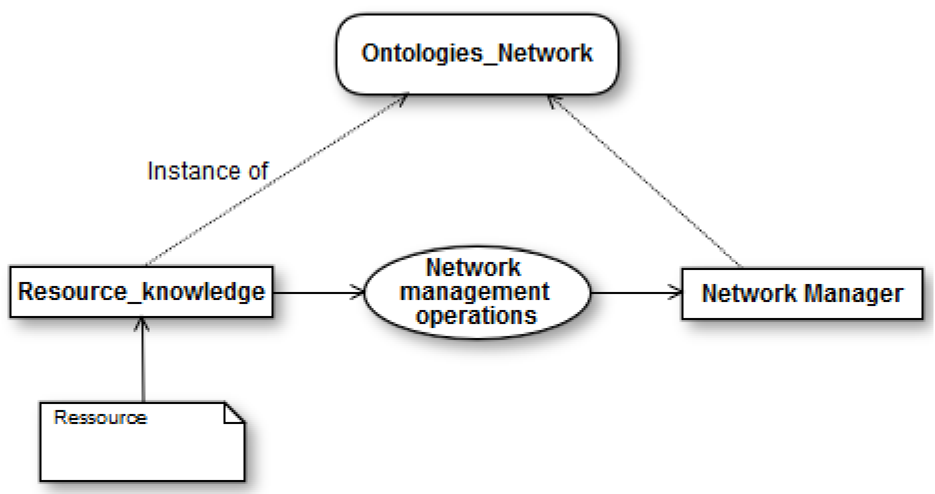

Figure 5.1: General approach

Let's R be a resource and O_i^R a representation of R. The NMO is the set of operations done to identify the resources representations containing correspondences with O_i^R . The output of the NMO will represent an instance of the NM class. The collection of the instances of the NM class represent an instance of the ON.

5.1.1 Resources representation

This part of our model aims to be able to describe the maximum of entities extracted for a different type of knowledge resource. Once integrated into the network, each resource will have a specific namespace. At this point, our model is different from the repository approach when the resources have a standard terminology. As for us, the resources need to be independent; the links between them are explicitly defined via the semantic similarity. Since there exist many different (and incompatible) ways to express knowledge in resources, it is difficult to create a single expressive, decidable and consistent, universal representation model for the content of resources. Different aspects represented by different formalism, cannot be reduced to even one (Kutz, Hois, Bao, & Grau, 2010). Consequently, after trying different design alternatives and applying several improvements and more expressiveness in the meta-model, we propose to start from a generic model that supports multiple representations for the content of a resource (Ghoula, Falquet, & Guyot, 2010). This model allows to create a model representing heterogeneous resources (different representation models) and to perform operations that involve several resources. Starting from the TOK_Model, we applied several modifications as this model was initially proposed for ontological resources, and we aim to represent ontological and nonontological resources (more details in the implementation chapter).

Metadata representation

We consider Meta-data as important as the content of the resource. Many representation models have been presented the structure of metadata elements, as most of them are standard for a broad set of format. we identified (see Figure 5.2) a set of critical features for describing knowledge resources and imported all the possible useful metadata representations from NoRMV and Dublin core (see section 3.2.1). Multiple resources format use Dublin core to represent meta-data elements of electronic resources. We also consider NoRMV to include extra meta-data elements. To design our model, we first identified and categorized the resources that we represent. Then, for each type of these resources, we investigated their structure, entities, and semantics. As shown in Figure 5.3, in to describe the different types of resources, we extended the NoRMV model using Dublin core metadata and we specified additional classes. For defining and validating our model, we followed an iterative process by trying different abstract models, testing them on resources from each category and then refining them or modifying them depending on their capacity of representing all the types of resources. Using the Tok_model for representing meta-data, combining with Dublin core and NoRmv meta-data elements and conducting several changes, we identified the structure of the resources representation model (Figure 5.3).

A knowledge resource has multiple metadata (Chapter 3.2) as described by Dublin core we kept from NoRMV:

- normv:Implementation :An implementation of non-ontological resource

1. has sub-classes: normv:Database c, normv:FlatFile c, normv:Spreadsheet c, normv:XML
2. is in range of normv:hasImplementation

From OMV model, we took the classes: omv: Person and omv: organization. We consider the omv: Person as a super_class of dc: Creator and dc: Contributor.

To the subclasses of implementations, we add OWL-S: Service to describe web services resources. OWL-S has subclasses: OWL_S: ServiceProfile, OWL_S: ServiceGrounding and OWL_S: ServiceModel.

We added an additional sub_class ont:HTML: an Html implementation. To normv: Spreadsheet we add two sub_classes: ont: Rows and ont: Columns and to normv: DB we add ont: Tables and ont: Relations as sub_classes and ont: Attributes as subclasses of ont: Tables.

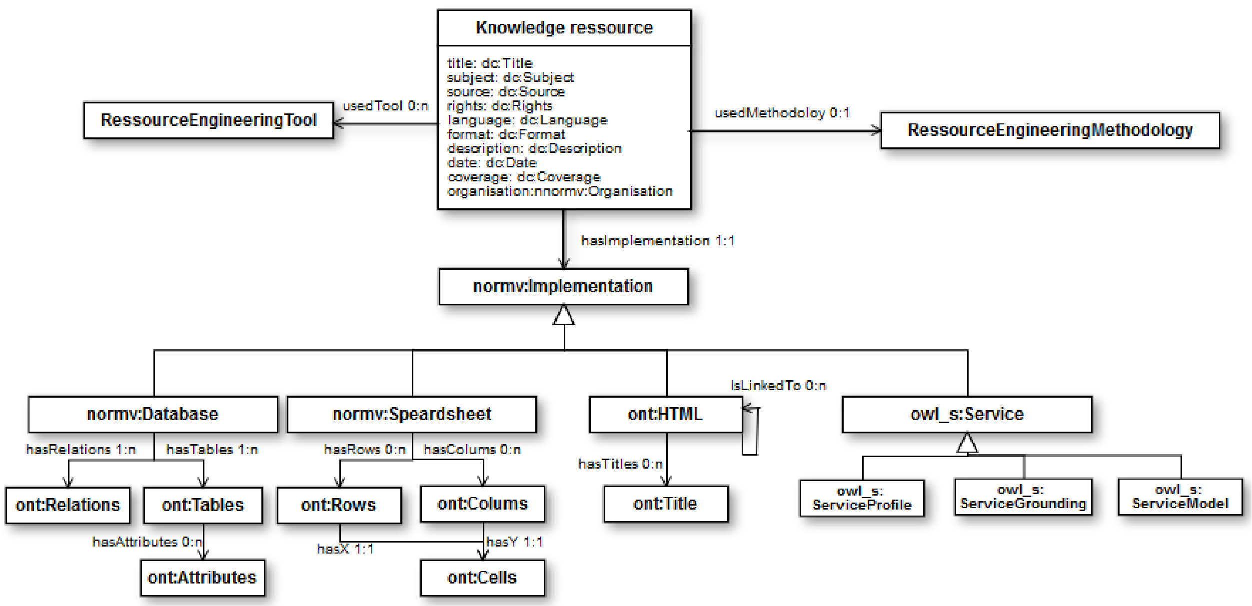

Figure 5.2: Knowledge Extraction Model (KEM)

To keep track of the life cycle of the resources, we consider additional classes as:

RessourceEngeneeringMethodology

This class describes the methodology used to transform the resource from its original format. The method can be characterized by its name, acronym, description, and documentation.

RessourceEngeneeringTool

It describes the tool used to parse the knowledge resource.

Content representation

As shown in Figure 5.3, the content of the resources is implemented in a different way depending on the resource format (tables, attributes and relations for databases, columns, and rows for spreadsheets format).

We intend to preserve the originality of all resources and their representation language to have less information loss (to keep the resources independent while begin aligned or to derive easily new representations or views from the original formalism). To handle multiple content representations, the ontology contains upper-level classes that can be refined using classes from specific representation models that we call “Patterns” (described in implementation chapter). A resource is defined using the class Resource_representation (described in 5.1.1.); its metadata are represented using the model in Figure 5.2.

Unlike the Tok_Model, we do not separate the representation of an entity form a resource. If the same entity belongs to two different resource, the relation will be modeled within the network by representing the link between the two entities and the two resources containing the entities. Thus, our model represented the entity types of different resources as subclasses of Resource_Entity. The content of a resource is expressed using the class Resource_Entity, which is an upper-level descriptor of the generic aspects of a resource’s item.

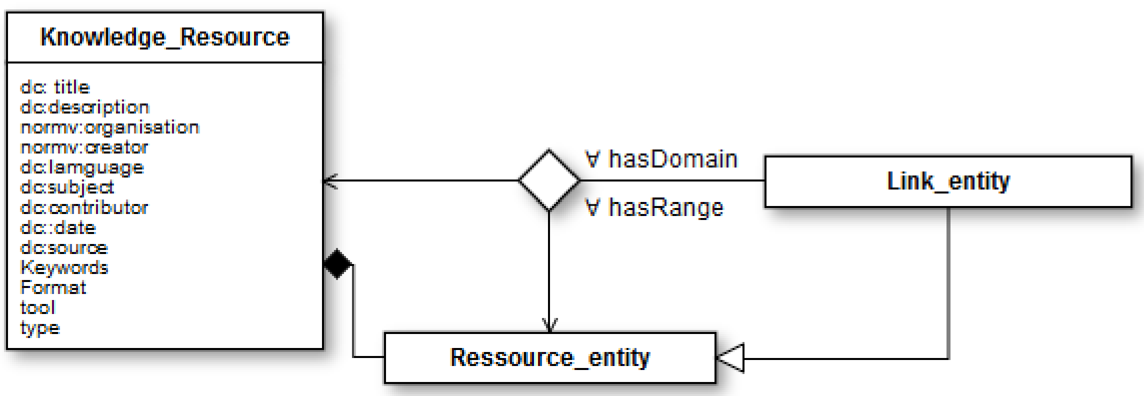

Figure 5.3: Excerpt of the content representation model of knowledge resources

For instance, a resource’s entity can be a sub-entity of, associated to or disjoint with another resource’s entity. The representation of a resource is completed through the description of its entities.

A Link_Entity is the crucial element of content description; it allows expressing relationships between a resource’s entities or with other entities. A linked entity can be a property, a predefined relation within a specific namespace or any type of relation. It connects different elements, which can be resources or entities.

For each type of knowledge resource (ontology, database schema, hypertext ...) there are specific types of link entities. An ontology may have link entities such as Concept_To_Concept, Concept_To_Term, Concept_To_Axiom... whereas a relational data schema may have Database_To_Table, Table_To_Attribute, Table_Primary_Key, Table_Foreign_Key, etc. These types correspond to subclasses of Link_entity.

If we take the example of a database: Considering the database “Student_Program.” The table “Student” has attributes: “age.” As a representation, we have: 1. all the metadata of the database, 2. the Resources_entities: “Student_Program,” Student,” “Age” linked using the Link_entities specific to the database format as below:

In RDF/Turtle syntax (without prefixes):

```
Student_Program a Knowledge_Resource .
Student a Resource_Entity .
Age a Resource_Entity.
student_program_to_student a Database_To_Table)
    hasDomain Student_Program ;
    hasRange Student .
Student a Table_To_Attribute)
    hasDomain Student ;
    hasRange age .
```

In this section, we describe how our model is designed to link the resource representation 1. to the original resource and 2. other resource representations within the network.

5.1.2 Model architecture

After multiple versions and conducting several changes, we identified the structure the conceptual knowledge schema represented over our network of ontologies Model (ONM) (Figure 5.4).

The approach we propose of representing knowledge resources within a network of knowledge is divided into three level.

1. The upper level represents the meta-meta-elements by defining abstract classes that describe the ontologies ’network model.
2. The second level refines the first one using a specific representation model that describes the model from a resource level. In this level, we can define the connections between the resources representations.
3. The third level represents the connection between the resource representations from an entity level.
4. The fourth level represents knowledge resources.

This approach is based on the MOF4 (Meta-Object Facility) standard, and it proposes to unify the representation of heterogeneous resources in a common formalism (Ghoula, Falquet, & Guyot, 2010). The details of the levels of the MOF-based representation in the context of our model are as follows: 1. Level M3: is the meta-meta-model (first layer) that describes the abstract elements required to represent knowledge resources within the network; 2. Level M2: meta-models that are related to resources representations formalism (e.g., generic model for representing resources, in our case OWL meta-model,); Level M1: is the model representing the resources via their entities, and the connection between the resources form an entity level. M0, represent an instance of knowledge resource that we represent in our model.

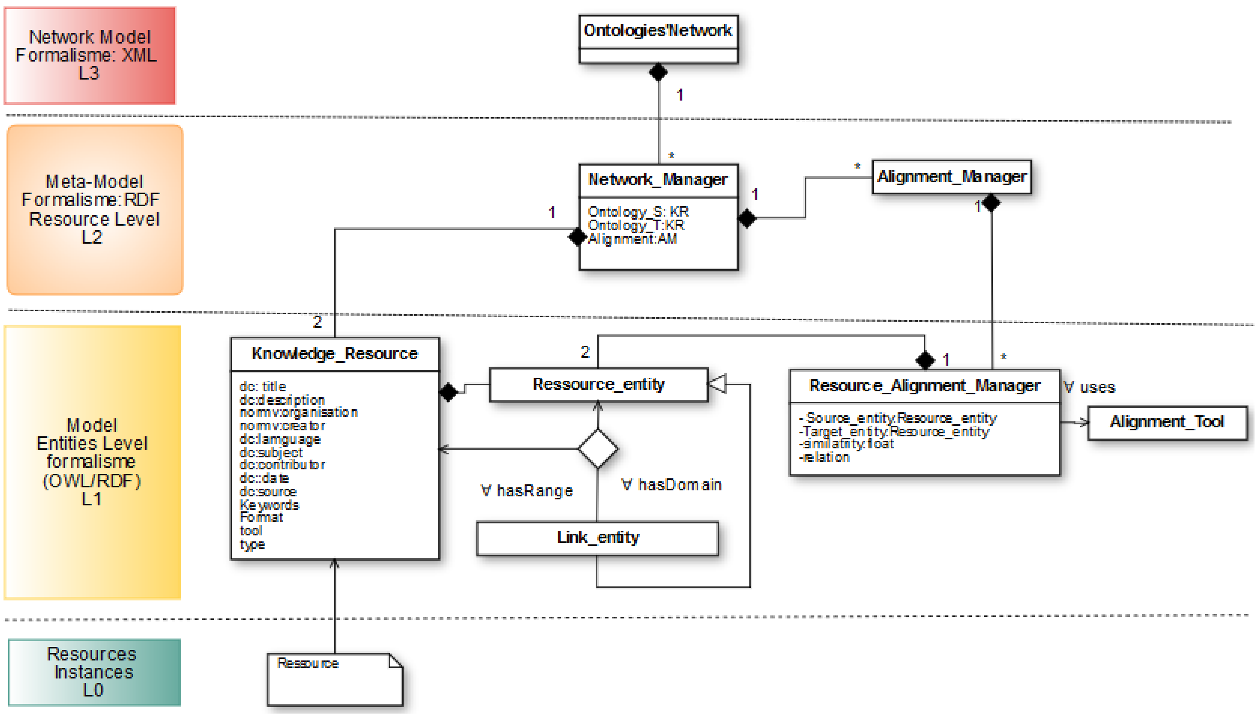

Figure 5.4: Ontologies 'network model (ONM)

This model is an aggregation of all instance of the Network Manager (described in section 5.3.2). To get these instances, the Alignment Manager get the set of Alignment resource management instance. Which manage the resources at the entities level using alignment operations to construct links between resources entities are represented using a dedicated model and may have multiple implementations and different kinds of input and output (described in section 5.3.3).

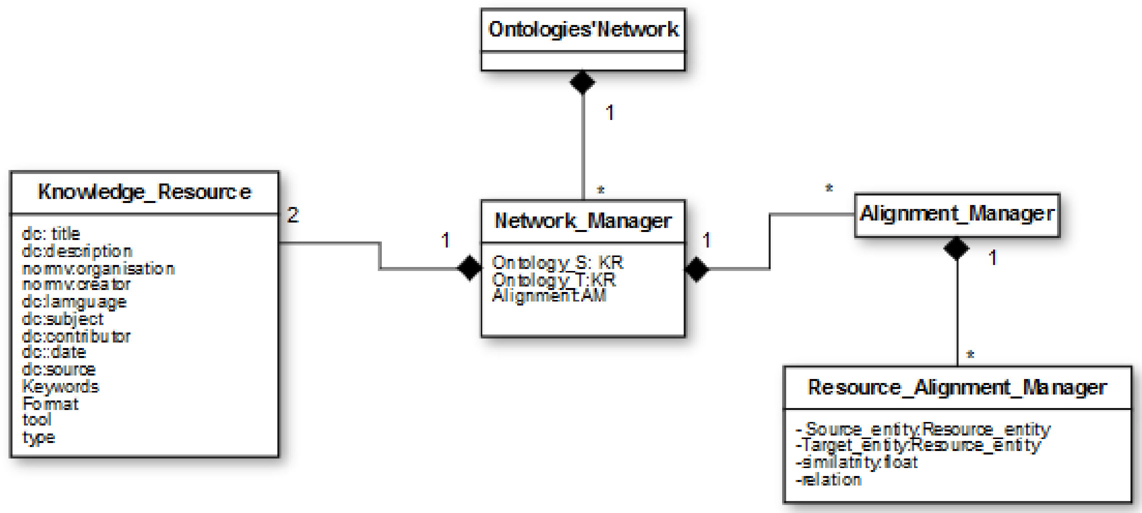

Figure 5.5: Description of Network management

Resource level representation

The class Ontologies'Network is composed of the set of the Network_Manger instances. Thus, it represents all the resources representations and how it is connected to each other.

The Network_Manager allows representing the relation between different resource representations. If two resources are connected, which means that their alignment results in a least one correspondence, we can find via the Network_Manager:

- Ontology_S: Knowledge_Resource: the namespace of the source knowledge resource representation,
- Ontology_T: Knowledge_Resource: the namespace of the target knowledge resource representation
- Alignment: Alignment_Manager: the path of the alignment resource, which is provided by the Alignment Manager.

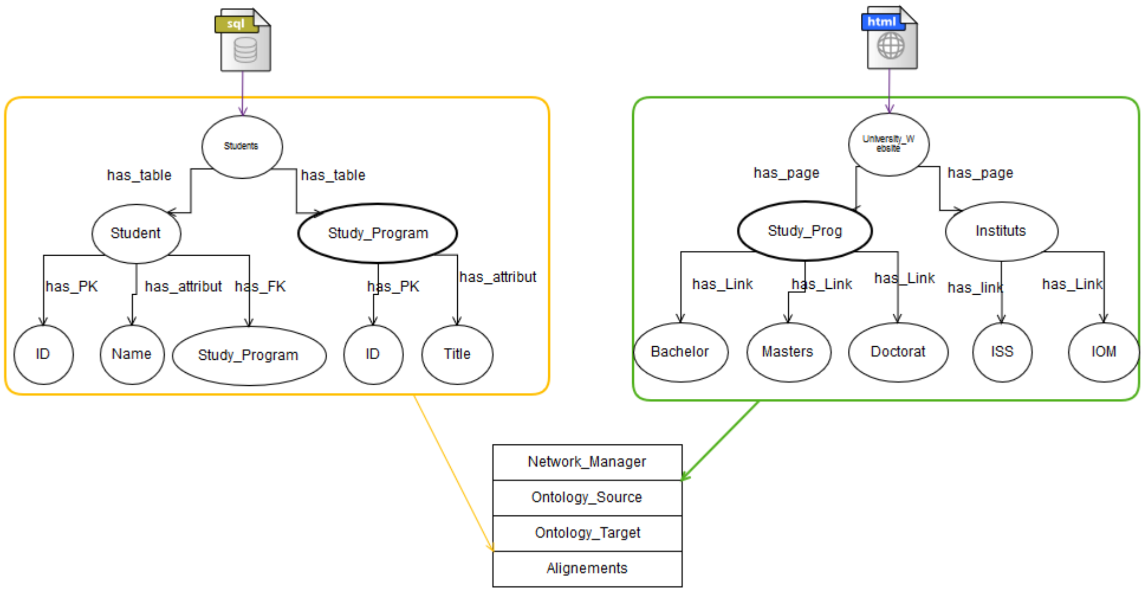

Figure 5.6: Example of a Network Manager instance

The Alignment Manager class contains all the Resources_entities aligned between two specific Resources_representations. AM represent the link between the Resource level and the entities level in our model.

Assuming that we have two resource representations as showed in Figure 5.6, the first one represent a student database, the second one represent the web site of the university. Assuming that the alignment process detect that these ressources are alignnebale, we then create an instance of the network manager class that will save the information about the source ontology, the target ontology and the set of alignements found.

Entity level representation

In this section, we explain how we model the connection between different resources representation form an entity level. The Resource_Alignment_Manager class contains two resource_entities form two different resource representation instances and use an alignment tool to measure their similarity. If a relation is found between the two resource_entities, then a new Resource_Alignment_Manager instance will be created containing, the:

- Source_entity: Resource_entity
- Target_entity: Resource_entity
- Similarity_ measurement: float
- Relation: {"=",">","<"}

The Alignment_Manager will contain the set of Resource_Alignment_manager instances for each two specific resource representations. The absence of RAM instance for two specific Resource_Representation and consequently of Alignment_Manager instance means that there is no connection between them.

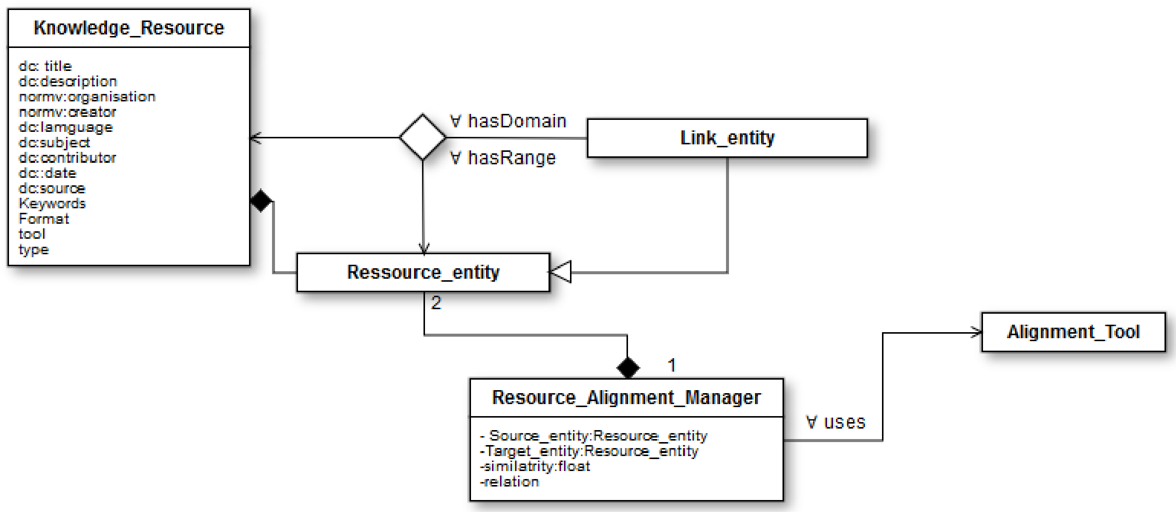

Figure 5.7: Description of Alignment management operation

Taking the example of the last section (Figure 5.8), for each alignment found, a ressource_alignement_manager instance will be created to save the information from an entity level, which means, the entity from the source ontology, the entity from the target ontology, the similarity score and the relation between the entities. On this diagram, the nodes represent Resource_Entities and the labelled edges represent Link_Entity's. The labels represent specific subclasses of Link_Entity that are used for this type of resource.

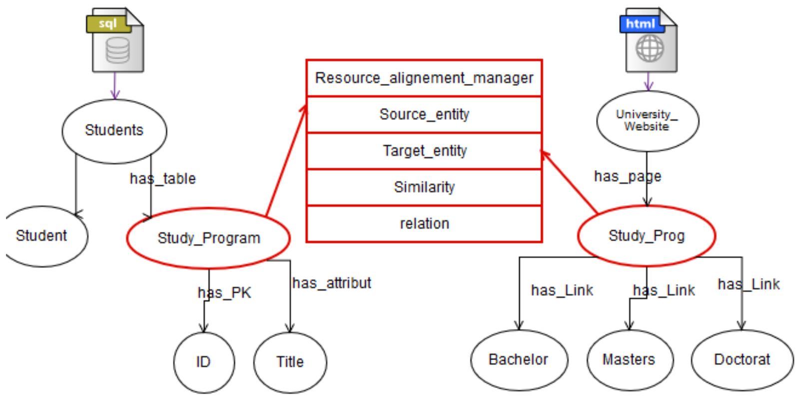

Figure 5.8: Example of a Ressource_alignement_Manager instance[69]

5.1.3 Conclusion

The ontologies' network model is a solution to the syntactic and structural heterogeneity of the resources by providing an abstract resource model able to represent any resource. In the next section (5.2), we describe our approach, which use the abstract model to instantiate it to a specific use. Thus, we describe our solution to resolve the problem of syntactic heterogeneity by introducing the Profile Ontology (section 5.2.1). Then, we present our methodology to transform explicit knowledge resource into resource representation. We also present our strategy to link the different compounds of our approach. The implementation of the different process and algorithms are detailed in the next chapter.

5.2 Ontologies' network-based approach

The Ontologies' Network Approach (ONA) allows using explicit knowledge resources to construct a conceptual knowledge schema using the Ontologies' Network model. The main idea behind our approach (see Figure 5.7) is to explain how we transform the knowledge resources into knowledge resources representation and how we built conceptual knowledge resource based on the ontologies' network model.

Many strategies have been tried to better use the content of the resources and the potential of alignment operation to link them and to cross resources to enrich the discovery of new knowledge.

The main idea behind our approach is to be able to create a representation for local knowledge resources and the link these representations to construct a network by finding the semantic relations between their concepts (Figure 5.2). To do so, two significant steps are needed:

1. For each resource and after investigating its format and we create a representation for it. The representation includes the meta-data that we can extract from the specific format of the resource, and its structure. Thus, for each format that we consider, we should prepare a pattern. The use of

[69] On this diagram, the nodes represent Resource_Entities and the labelled edges represent Link_Entity's. The labels represent specific subclasses of Link_Entity that are used for this type of resource.

the pattern leads to easy annotate the entities extracted with their role in the original resource. For example, considering that the original resource is a database, using the specific pattern for databases, we can annotate an extracted entity automatically if it is a “table” or “an attribute” or “a relation” (described in section 5.2.1)

2. We annotate the concepts of the knowledge representations with the concepts of the “Profile Ontology” (PO).

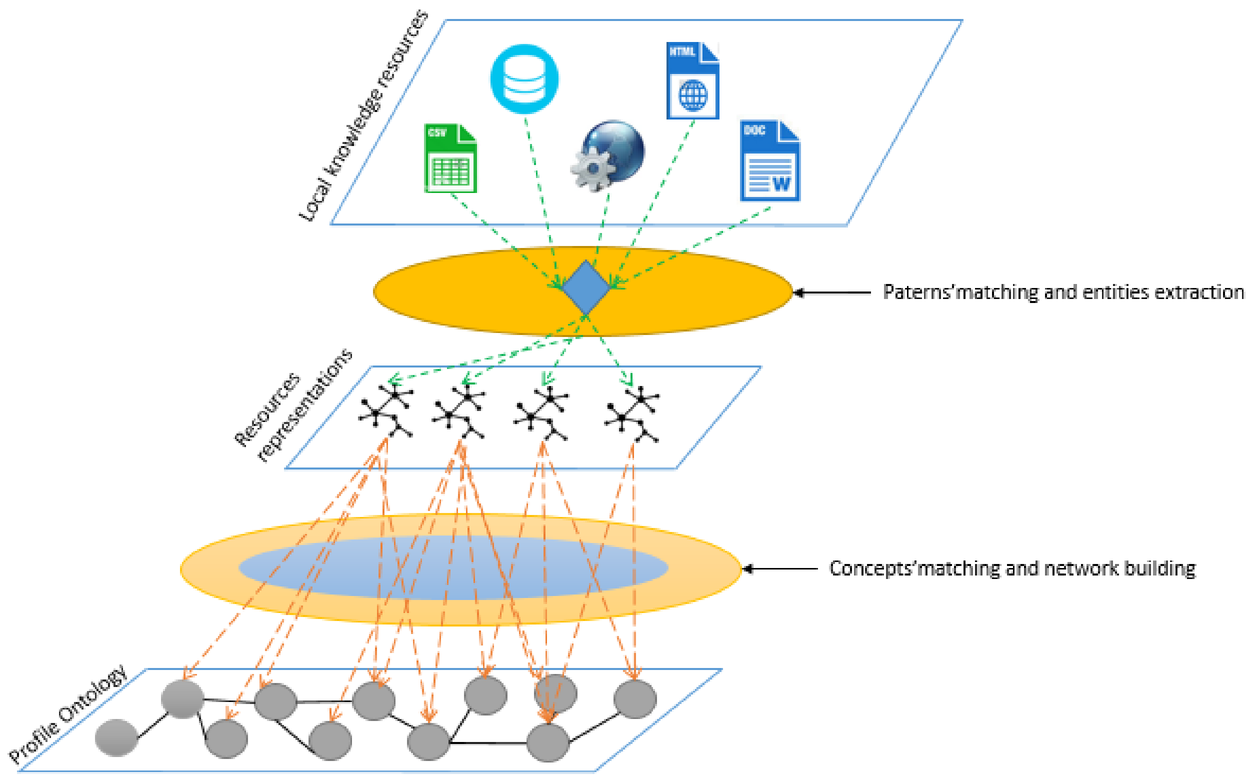

Figure 5.9: Ontologies’ network approach

Our approach is composed of two major operations:

1. Transform knowledge resource in a specific format to an instance of a resource representation class (described in section 5.2.1)
2. Build the network using connection classes of ONM (described in section 5.2.2)

5.2.1 Knowledge resource transformation process

As discussed in the section (3.2), the knowledge resources are implemented in different ways depending on the resource format. To use the resource representation model, we use a pattern approach to select the right algorithm for a specific resource format when we need to integrate a new resource into our network. The transformation operation includes:

1. The extraction of the elements required by the resource representation model

2. The annotation of the different entities extracted with their role in the original resource.

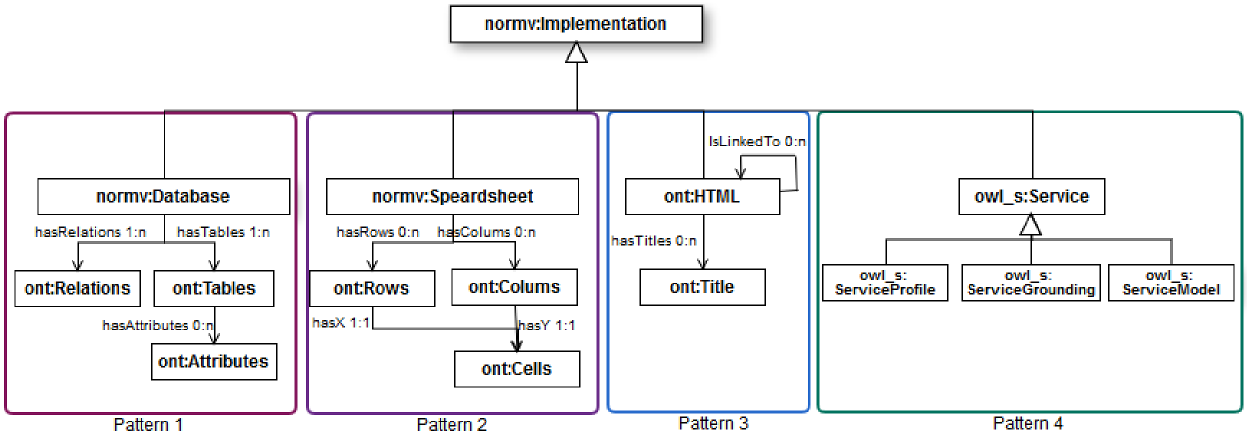

Figure 5.10: Resources transformation patterns

In chapter 6, we describe the various algorithms that implement the different patterns presented in the model we described in (1.3) (see Figure 5.8).

5.2.2 Ontology network construction approach

Our model offers the tool to represent the knowledge resources and their connections. The question that we answer in this section is which strategy we adapt to link the resource representations better.

Our idea is to add a semantic layer on the top of the network for two specific reasons:

1. The semantic layer will lead to unifying the vocabulary of the network. Thus, we annotate the resource representations with an ontology that we called "Profile Ontology."
2. This approach will avoid us to connect all the resource representations systematically together, which can be fatal in the case of an extensive network.

Profile Ontology

The Profile Ontology is a particular domain ontology, which aims to represent a generic representation of the specific organization using our system. The system can be used by any organization and any domain, the only parameter to change will be the PO. Among the specificity of a domain, applications are that the:

- Vocabulary set is limited.
- Semantic ambiguities are rare.
- Terms and jargon of the domain appear frequently.

Accordingly, the following assumptions could be derived:

1. The use of synonyms is very often
2. It is relatively easy to build a domain ontology that includes terms or jargon used in a specified domain.

3. Relations used in a specific domain are limited.

These assumptions take their full extent in the case of a specific organization. Thus, due to hypothesis 1, we can consider that an approach based on a network of ontology constructed using ontology matching techniques will work well in deciding the similarity between the meanings of concepts extracted from knowledge resources produced within the same organization.

In our project, WordNet is employed as a background ontology to enhance the detection of synonyms. We use WordNet as a background ontology to align ontologies from a specific domain gave pertinent results as per (Safar, Reynaud, & Calvier, 2007). This work is feasible considering assumption 2 and three as the Profile Ontology will be built manually to extend the Organizational ontology to a specific organization to specify the "jargon" used within the particular organization and hierarchical relations between the terms of its vocabulary.

The Profile Ontology provides an ontological description of an organization. Several studies have addressed the issue; research has been focused on manly four perspectives namely: functional, behavioral, organizational and informational. These perspectives underlie separate yet interrelated representations for analyzing and presenting processes.

Figure 5.11: Profile Ontology

Recently the W3C published the organization ontology[70]: an ontology designed to enable publication of information based on organizations and organizational structures including governmental organizations. It is intended to provide a generic, reusable core ontology that can be extended or specialized for use in particular situations. This coverage corresponds to the type of information typically found in organizational charts. The W3C organization ontology will be used as a vocabulary to define a specific ontology for a

[70] https://www.w3.org/TR/vocab-org/

particular case of the user. Since that, the construction of the ontology representing the profile will be done in flowing the above steps:

1. We identified the purpose of the ontology: be able to describe functional, behavioral, and organizational and information aspect. Which mean be able, ones we add the knowledge resources produced by the employees of the enterprise to find the specific department from where we can ask for more information.
2. We identified documents from where we can extract the precise definitions and vocabulary used within the company.
3. We interviewed the different services of the company to specify the process.
4. We identified extra ontological resources that can enrich the enterprise ontology.
5. We determined the rules (axioms) relating the different concepts of the ontology.
6. We implement the ontology using Protégé
7. We then evaluate the ontology based on our final objective, which can connect the knowledge resources and present a full vocabulary to describe the organization.

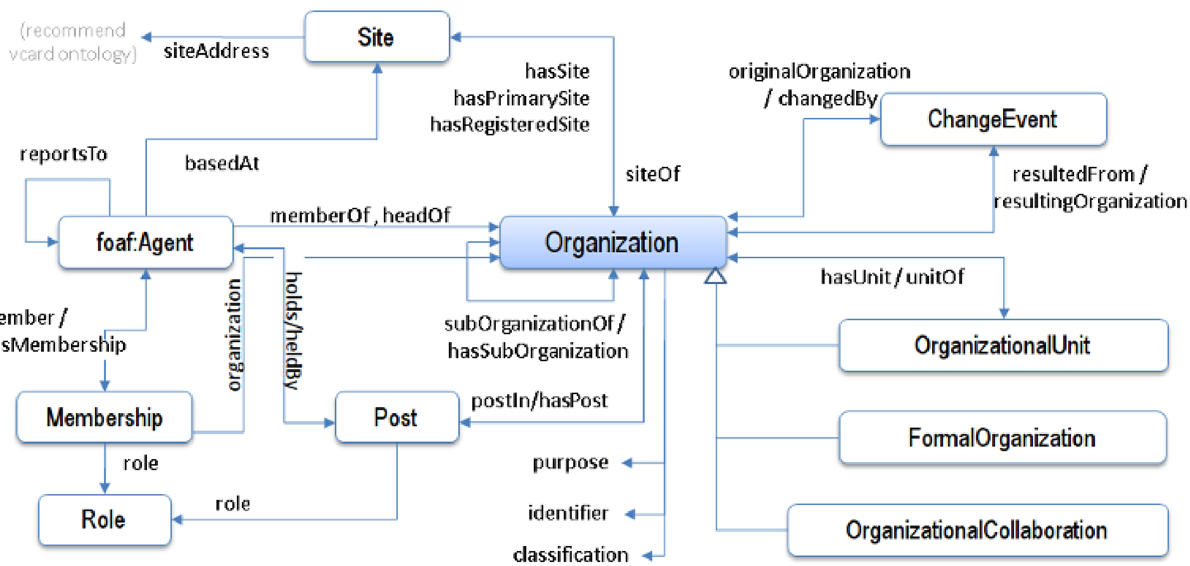

Figure 5.12: Organization ontology description- W3C

Since that, the organizational ontology is a meta-ontology describing the based concepts of an organization. This ontology helps as a start point to define a specific ontology for a particular organization. One specific ontology has to be implemented by the organization describing functional, behavioral, organizational and informational aspects of it.

A full description of the ontology will be detailed in chapter user case, as the specific ontology is specific for each organization. Here we present the implementation steps to specify the purpose of having such an ontology in our approach.

The Network building process

Networks of ontologies are made of a collection of logic theories, called ontologies, related by alignments. They arise naturally in distributed contexts in which ontologies are developed and maintained independently, such as the semantic web (Euzenat, 2014). A network of ontologies: Formally, a network of ontologies $(\boldsymbol{O}^r, \boldsymbol{A})$, is made of a finite set $\boldsymbol{O}^r$ of ontologies and a set of alignments between these ontologies. We denote by $\boldsymbol{A}\{O, O'\}$ the set of alignments in A between O and O' (Euzenat, 2015).

Considering the formal definition, we define an algorithm to specify our construction strategy. Our strategy is based initially on the Hybrid approach presented in the state of the art section (3.3.1.3). We are using the Profile Ontology as a specific ontology that describes the Organization vocabulary. After being parsed, we align the new resource with the Profile Ontology. If the alignment process results in maps of similar concepts, then we save the concepts aligned using the Resource_Alignment_Manager, and we end up the process. This point of our approach means that the resource added contains entities expressed with the vocabulary of the organization or that the knowledge expressed is from the general concepts expressed within the Profile Ontology.

In case of no correspondence between the Profile Ontology and the representation of the new resource, we align the new resource representation with the resource representations of the network. This case should mean that new “knowledge” that was not specified in the Profile Ontology is added to the network which constitutes a way of the evolution of systems based on our approach that we discuss in the last chapter of this work.

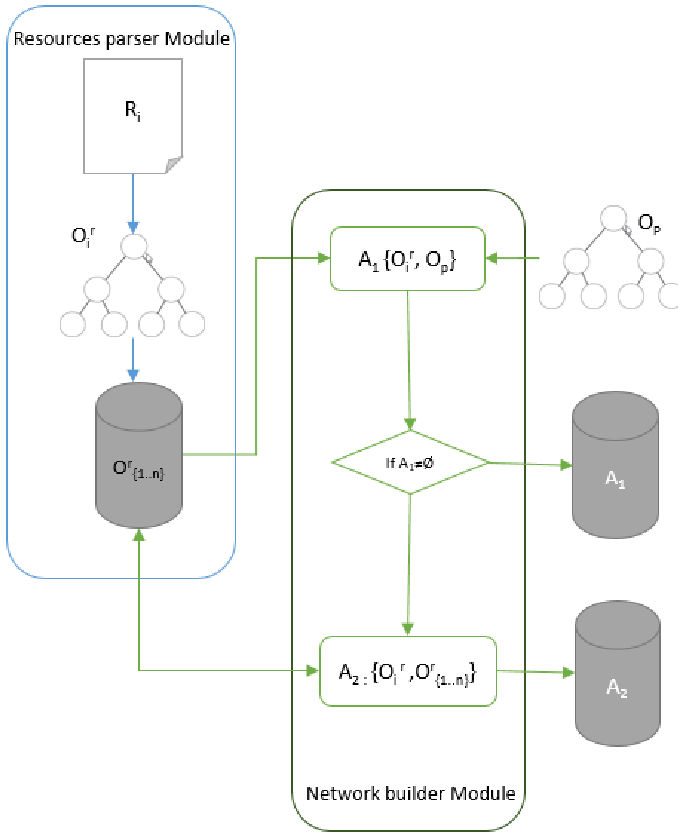

Figure 5.13: Ontologies network building approach

5.3 Ontology network querying approach

This approach has been created to allow users to access to knowledge within the conceptual schema based on ontologies' network schema. We defined a new approach to access the knowledge schema using a collection of keywords. We described the approach form Keyword-based search perspective. However, our approach steps can be easily used for navigation based search or a combination of both of them. Thus as the final result of our approach is a list of pertinent knowledge resources for specific keywords and the

5.3.1 Approach steps

Our approach is composed of three steps illustrated in Figure 5.11:

1. From keyword-based search to concept-based search: The first step is to create an ontology of concepts from the user query O_{q^L}. As we assume that the user does not have any idea about the conceptual schema, we do not put any restrictions about the natural language that he can use, or the terms he can enter. Formally, we can represent the operation as below:

***Input*:** $\boldsymbol{Q^L}$= $\{T_1, T_2... T_n\}$ (where T= Term, L= natural language, Q= user Query, n= number of terms entered by the user)

***The output*: O_q^L** $= \{C_{T^1}, C_{T^2},.., C_{T^n}\}$ where each C_{T^k} is a concept labeled with the corresponding term T_k.

Note that this ontology is extremely simple (it does not have any axiom). Its only purpose is to be aligned with Op.

2. Expressing the Query ontology with the vocabulary of the network: The goal of this step is to map the query terms, represented by the query concepts (the C_{T^k}'s) to the organization's concepts. This is done by aligning the query ontology Q_{q^L} with the organization ontology O_p (S2 Figure 5.11). and assuming that we can find correspondence between the three terms of the query and the concepts of the Profile Ontology Op:

***Input*: O_q^L** $= \{C_{T^1}, C_{T^2},.., C_{T^n}\}$

Output: A set of concepts $\{D_1, D_2, ..., D_k,\} \in O_p$ such that there is an alignment $A(O_{q^L}, O_p) \supseteq \{(C_1, D_1, s_1), (C_2, D_2, s_2), ..., (C_k, D_k, s_k)\}$, the s_i's are similarity scores that are greater than a fixed threshold

3. Querying results: At this step, we will query the network ontologies using the Op's selected concepts from the step 2. This third step contains two sub steps:
   a. Sub step 1: Selecting the resources' ontologies O^r whose alignment with O_p contains the selected concepts from step (2) ($S3^1$ Figure 5.11)

      Input: $\{D_1, D_2, ..., D_k,\} \in O_p$

      Output: A set of concepts E= $\{\{E_1^1, E_2^1,.. E_{n1}^1\}, \{E_1^2, E_2^2,.. E^2_{n2}\},\ldots,\{E_1^m, E_2^m,.. E^m_{nm}\}\} \in O_x = \{O_x^1, O_x^2,\ldots, O_x^m\} \in O^r$ (O^r = the set of ontologies composing the network) such that there is an alignments $Aj(O_p, O_j^r) \supseteq \{(D_1, E^r_1, s^x), (D_2, E_2^r\ s^x{}_2), ..., (D_j, E^r_{nm}, s^x{}_V)\}$, the $s^x{}_V$'s are similarity scores that are greater than a fixed threshold

This first sub-step outputs the first list of resources' ontologies with a direct link with the profile ontology, which is means, that these resources have concepts with a strong link to the domain.

   b. Sub step 2: For all the ontologies selected in sub step 1, we take the concepts aligned with the profile and find in the inter-network O^r not chosen in supstep1, which their alignment contains correspondences with concepts selected from substep1.($S3^2$ Figure 6.11)

      Input: E

      Output:* *B $= \{\{B_1^1, B_2^1,.. B_{n1}^1\}, \{B_1^2, B_2^2,.. B^2_{n2}\},\ldots,\{B_1^m, B_2^m,.. B^m_{nm}\}\} \in O_y = \{O_Y^1, O_Y^2,\ldots, O^m_Y\} \in O^r$ such $O_y \not\subset O_x$ and $(O_y \cup O_x) \subseteq O^r$.

      The output of this step is a list of ontology with a non-direct link with the profile ontology.

The output of the approach is: $O_y \cup O_x$.

5.3.2 Example

To better explain the results of our search approach, we give two examples on 1. how we use the alignment between the resources representations to find Knowledge resources corresponding to a user query and 2. how our resource representation model allows us to see precisely the role of a concept selected on a knowledge resource.

Assuming that we have:

1. a Profile Ontology of a specific university $O_P \supset \{C_a$: AcademicService, C_c: Study_Prog}

2. a resource representation of Student database

3. a resource representation of the University website

4. a query Q = "Study Program."

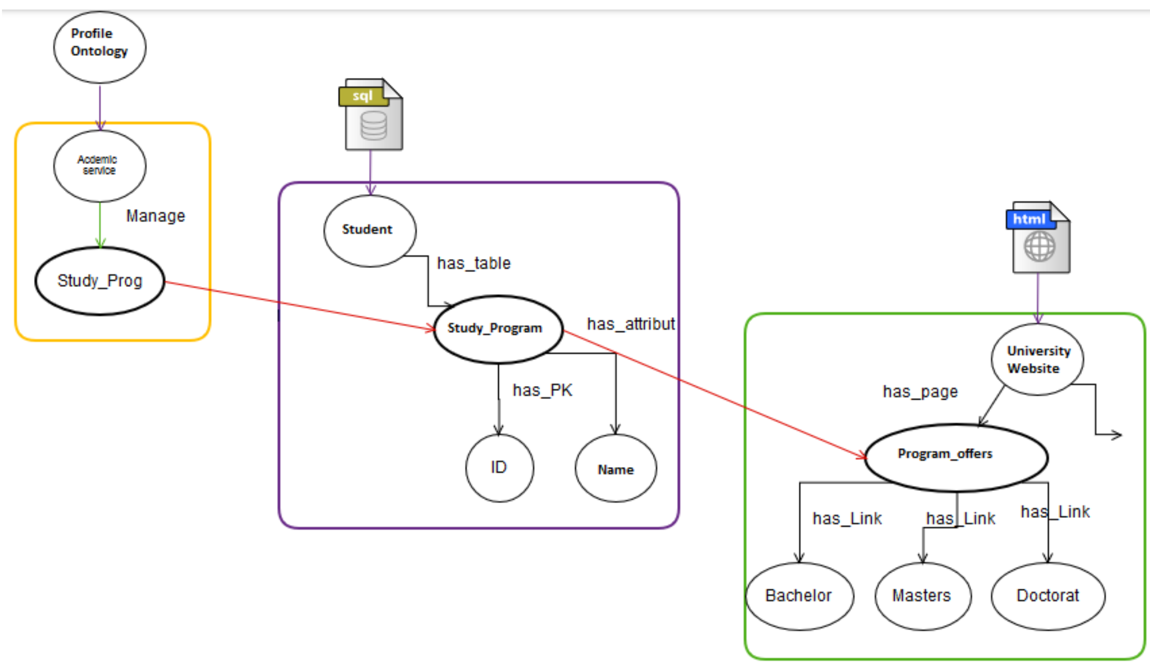

Figure 5.14: Ontologies' Network Querying approach

Knowledge Resources detection

1. Step: creation of the O_q: $O_q = \{C_1$: Study_Program}.
2. Step: A(O_q,O_p)={(Study_Program,Study_Prog,0.8)}

3. Step: Let O_r^a be the set of ontologies whose alignment contains correspondence with {C_a: Study_Prog }.
    1. 3.1: Query the conceptual schema using the vocabulary of O_p, and find O_r^a, which is, in our example, the resource representation of the database "Student."
    2. 3.2: Let O_r^b be the set of ontologies that have not been selected in step 3.1. In our example, the resource representation of the University website.

Structure Detection

Now we have the two resources selected:

The resource representation of the Student database and the resource representation of the University website. As shown in Figure 5.13, we can easily note that the concept selected from the first resource representation is a table in the Student database, and the concept selected from the second resource representation is a title of a page in the University website.

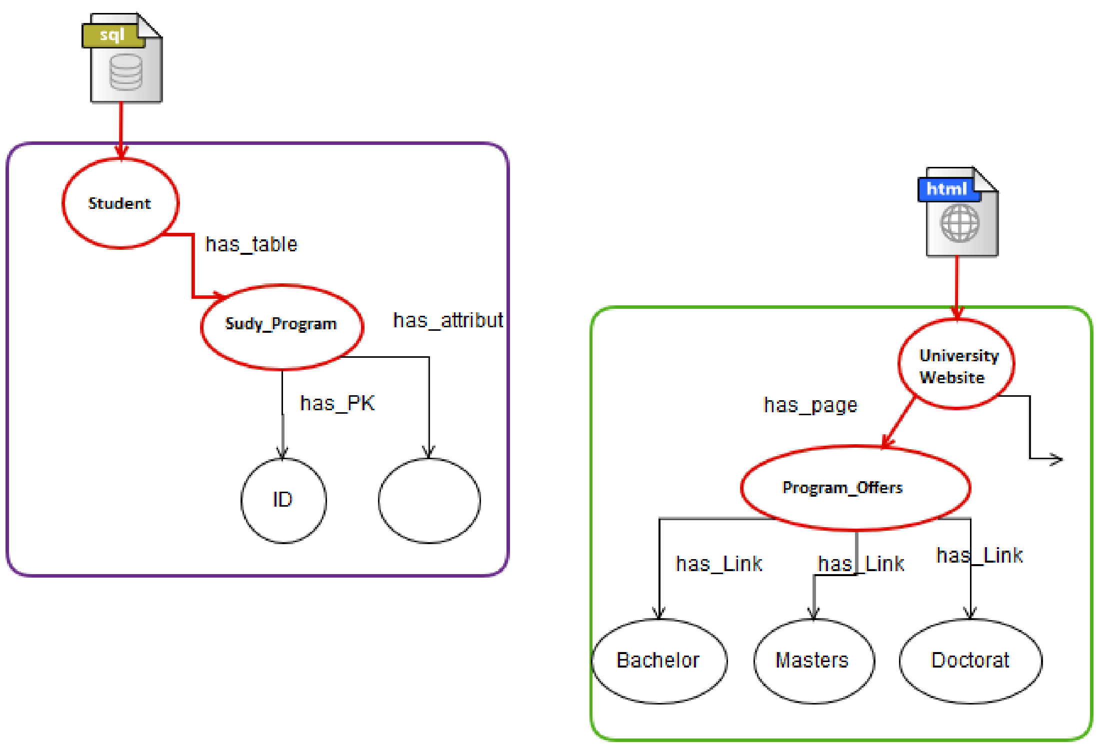

Figure 5.15: Example – resources' structures

5.4 Discussion

In this chapter, we first described our model to represent and manage knowledge resource within a network of ontologies. Our model handles representations of knowledge resource based on their metadata and structure and maintains their semantic correspondence that is considered as the connection elements of our knowledge network. The resource level connections and the entity level connections are the different way to link in the first hand, the resource to the network and the original knowledge resource on the other side.

In the second section of this chapter, we presented our approach to using our model to represent heterogeneous structural resources and we described how we add a semantic layer to deal with their semantic heterogeneity. The third section was dedicated to explaining, how can access to the knowledge within our model and we gave to examples to answer the question: Can we find the pertinent resources and the exact role of specific user keywords,

As we explain in the previous sections, our approach is general, as it can be used for any organization. For that, we will only need to create a specific Profile Ontology to specify the organization vocabulary. In addition, as we use Design pattern approach for knowledge resources transformation, we can easily add new patterns for additional resources formats.

In the next chapter, we present the different algorithms we created and used to implement the different patterns for knowledge resources transformation, and we describe how we applied the various procedures of our approach.

Chapitre 6. Implementation and algorithms

6.1 Introduction

Our approach for building a network of resources based on our model can be implemented in several ways[71]. In this chapter, we give an example of the implementation process of our approach to demonstrate its feasibility. Thus, for the first part of our approach that is the transformation process from knowledge resources to resource representations, we describe several algorithms for several resources formats capable of transforming a knowledge resource to a resource representation as per our model implemented as OWL ontologies. In the second part of this chapter, we present an implementation of the network construction process. Finally, we present an implementation of a search process to access to knowledge within the network.

6.2 Resources ontologies

Resources ontologies are the representation of the organizational resources following our model (described in chapter 5.2.2). As specified in state of the art, there are many systems each of them has been implemented to extract information for a specific task. As explained in the section (5.2.1), the resource transformation is based on the design pattern approach. Thus, for each resource format we can add a transformation pattern into the Knowledge_loading module (Figure 6.1). In this section, we describe the algorithms of four resources formats transformation (represented in Figure 5.2) as examples of structured and unstructured format to be able to demonstrate that our model is capable of representing different formats. After analyzing the results of state of the art (section 3.2.2), and as the resource transformation methods are not the goal of our theses, we decided to implement simple transformation algorithms to demonstrate the feasibility of our approach. Further possibilities for future works will be discussed in the last chapter of this thesis.

[71] discussed in chapter 7

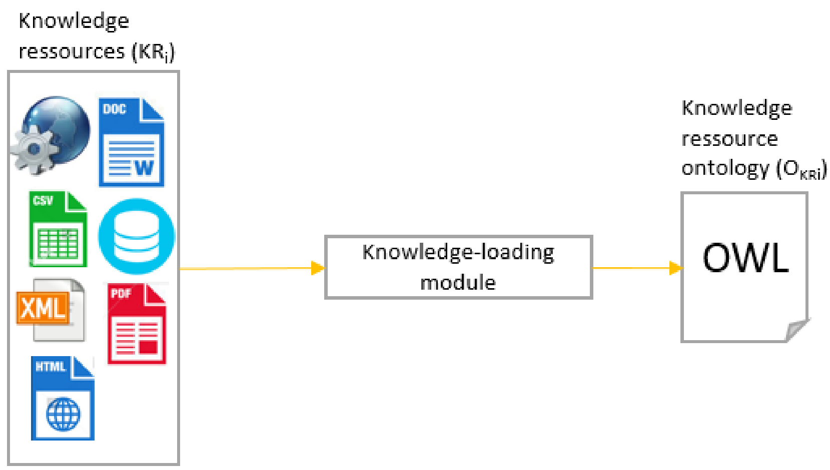

Figure 6.1: Resource transformation process

In this section, we demonstrate how the resources ontologies will be implemented. Thus, for each of the four resource formats selected, we present an example of the different operations and the output of the transformation process.

6.2.1 Databases knowledge extraction

As per the resources model represented in section 5.1.2, also of the meta-data of a specific database, we need to extract information as the names of the database tables, their different attributes, and relations. The final resource representation needs to contain the identical structure of the database to be able to find the exact role of any concept selected from this resource. As presented in state of the art (Table 3.1), different tools and algorithms can represent such information, but no one of them is available. Therefore, to explain the feasibility of our approach for DB resources, we can take the example of the model transformation algorithm described in (Alaoui, Hajjamy, & Bahaj, 2014).

Moving from a representation with Resource_Entity and Link_Entity to a representation with classes and properties is one of possible implementations of our model depending of the ontological language that we want to use.

As we choose OWL, we represent Resource_Entity as Classes because OWL alignment systems are generally more performant aligning classes than instances.

```
Input: A knowledge resource D that represents a relational database schema

Output: An OWL ontology that represents the same database schema

Create a class D (if it does not exist yet)

for each Resource_Entity T that is linked to D through a Database_To_Table link

       Create a class T

       for each Resource_Entity A that is linked to T through a Table_To_Attribut link

              Create a class A (if it does not exist yet)

              Create a property has_attribute_T_A

              Create an axiom  has_attribute_T_A domain T

              Create an axiom  has_attribute_T_A range A

              if A is a primary key of T (linked to T through Table_to_Primary_Key)

                     Create a property has_primary_key_T_A

                     Create an axiom  has_primary_key_T_A domain T

                     Create an axiom  has_primary_key_T_A range A

              if A is a foreign key of T to T' (linked to T through Table_to_Foreign_Key)

                     Create a property has_foreign_key_T_A

                     Create an axiom  has_ foreign _key_T_A domain T

                     Create an axiom  has_ foreign _key_T_A range T'
```

Algorithm 6.1: From a database representation to OWL

The Algorithm 6.1 is a simple example of how using our model we can transform a knowledge resource to a resource representation. We create four Object properties (Figure 6.5): has_table, has_attribute, has_primary_key and has the _foreign key. We create a Class for the database name.

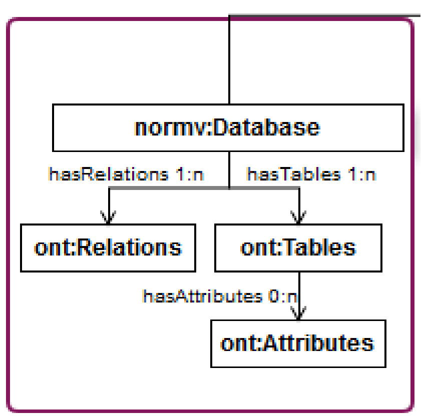

Figure 6.2: Database transformation pattern

For each table of the database, we create a class, and we set the domain of the Object property the class representing the name of the database, and as a range, the class representing the name of the table. Then, for each attribute, if it is a primary key we set the Object_property: Primary Key, if it is a foreign key, we set the object property foreign key else we set it with the object property Has_attribute.

To explain the transformation process from a database to a database representation, we give an example (Figure 6.6):

In this database, we have two tables: "Student" and "Study_Prog" in each table we have a list of attributes and one relation between the two tables. Using the algorithm 6.1, we obtained:

Etudiants.acces class, which represent the database name. The tables' names of the DB have been represented as Concepts (Student, Study_Prog). The Object Property (Has_table) was used twice to link the database name to the tables' names. For each table, the attributes were represented as Concepts and linked by the Object Property (Has _Attribute) to their specific table. The ID which represents a Primary Key for the "Studentt" table has been linked to the table "Student" via the Object Property (has_Primary_Key) and the attribute "Student" has been linked to the table "Study_Prog" via the Object_property "has_foreign_key" (Figure 6.4).

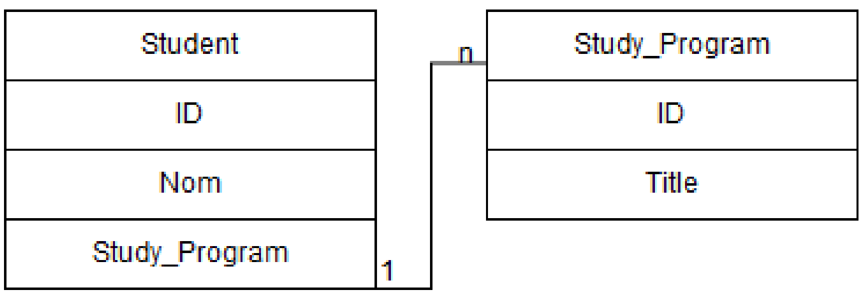

Figure 6.3: Example: Student database

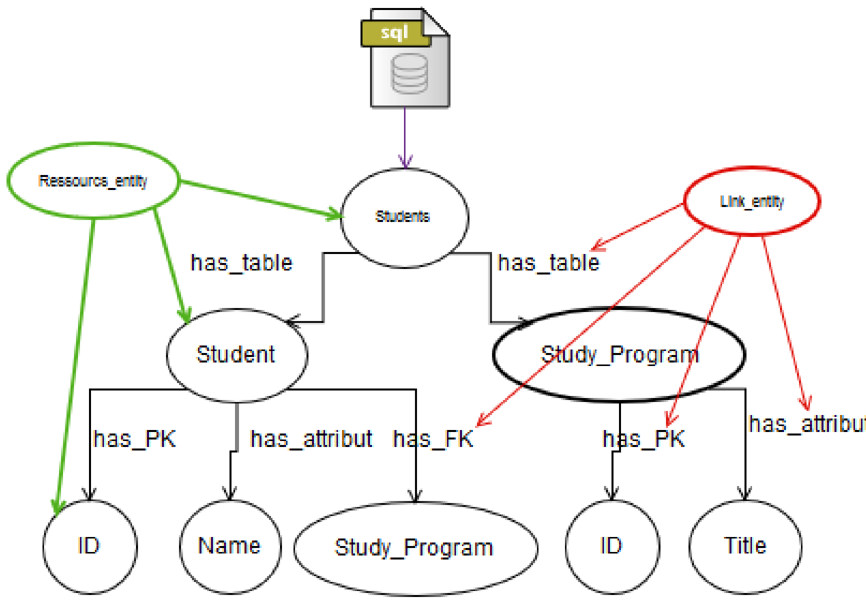

Figure 6.4: Database ontology constructed using the Database representation pattern

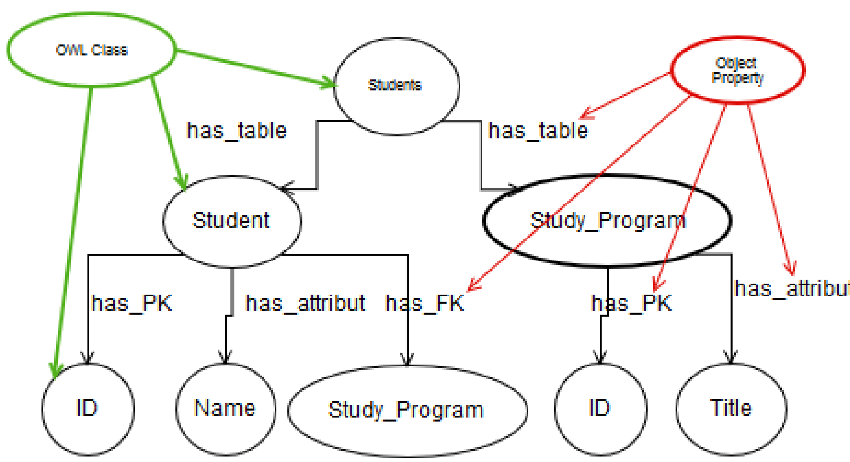

Figure 6.5: Transformation from the database representation to OWL ontology

The use of the database pattern to build the ontology representing a specific database ensure to find the role of a particular concept within the original resource. For example, we can easily find out that the concept (Study_Program) represents an attribute in the two tables of the database “Student.acces.”

6.2.2 Spreadsheets knowledge extraction

To represent a spreadsheet using our resource representation model, we investigate if any approach is capable of expressing a spreadsheet as per our pattern. For spear sheet knowledge resources, we instantiate the resource_entity as a class and the link _entity as an Object property.

The Object-properties we use for this format: HasColums, HasRows, hasCells. Testing this pattern, we figure-out that Spreadsheets do not contain significant knowledge if it is treated separately. Thus, for this resource format, we decided to transform a corpus of the spreadsheet document. Therefore, we used the Pattern2.1, and we add the objectProperties: Contains with a domain: Class: Corpus_name, and a range: Class Document_Name.

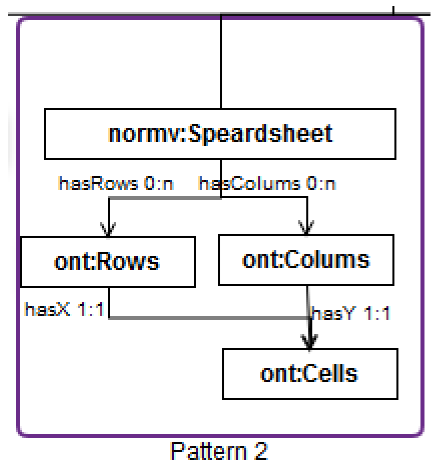

Figure 6.6: Spreadsheet transformation pattern

In the next chapter, we give a concrete example of the use of our implementation strategy, but to explain it, we present a simple algorithm 6.2 to transform a spreadsheet representation to an OWL.

```
Input: A knowledge resource D that represents a spreadsheet corpus
Output: An OWL ontology that represents the same spreadsheet
Create a class D (if it does not exist yet)
for each Resource_Entity S that is linked to D through a Corpus_To_Sheet link
     Create a class S (if it does not exist yet)
     for each Resource_Entity  R that is linked to S through a Sheet_To_Row link
         Create a class R (if it does not exist yet)
         Create a property has_attribute_S_R
         Create an axiom  has_attribute_S_R domain S
         Create an axiom  has_attribute_S_R range R
         for each Resource_Entity  C that is linked to S through a Sheet_To_Colum link
```

```
Create a class C (if it does not exist yet)
Create a property has_attribute_S_C
Create an axiom  has_attribute_S_C domain S
Create an axiom  has_attribute_S_C range
```

Algorithm 6.2: From a spreadsheet representation to OWL

6.2.3 Web services knowledge extraction

As discussed in section 3.2.2.2, the OWL-S[72]formerly DAML-S) is an ontology of services that allows user and software agent to discover, invoke, compose and monitor web resources offering particular services and having particular properties with a high degree of automation (Wali & Gibaud, 2012). Thus, we took this ontology as a pattern to represent web services resources.

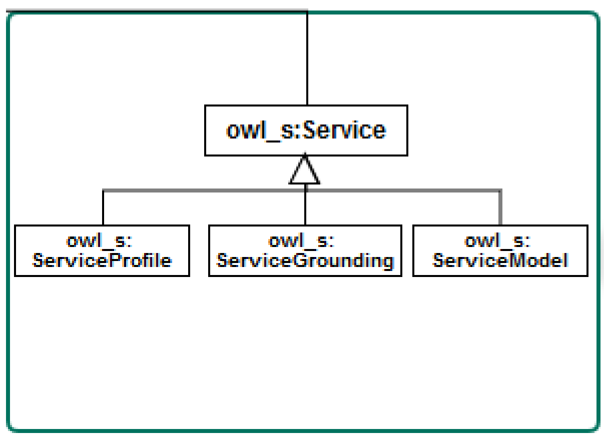

Figure 6.7: Webservice transformation pattern

As an example of the use of this ontology to represent web service flowing our resource representation model:

1. Link_entities will be instantiated as: ObjectProperty, DataTypeProperty and SubClass relations
2. Resource_entities will be instantiated as OWL Class such as Parameter name, Condition, Result.

[72] https://www.w3.org/Submission/OWL-S/

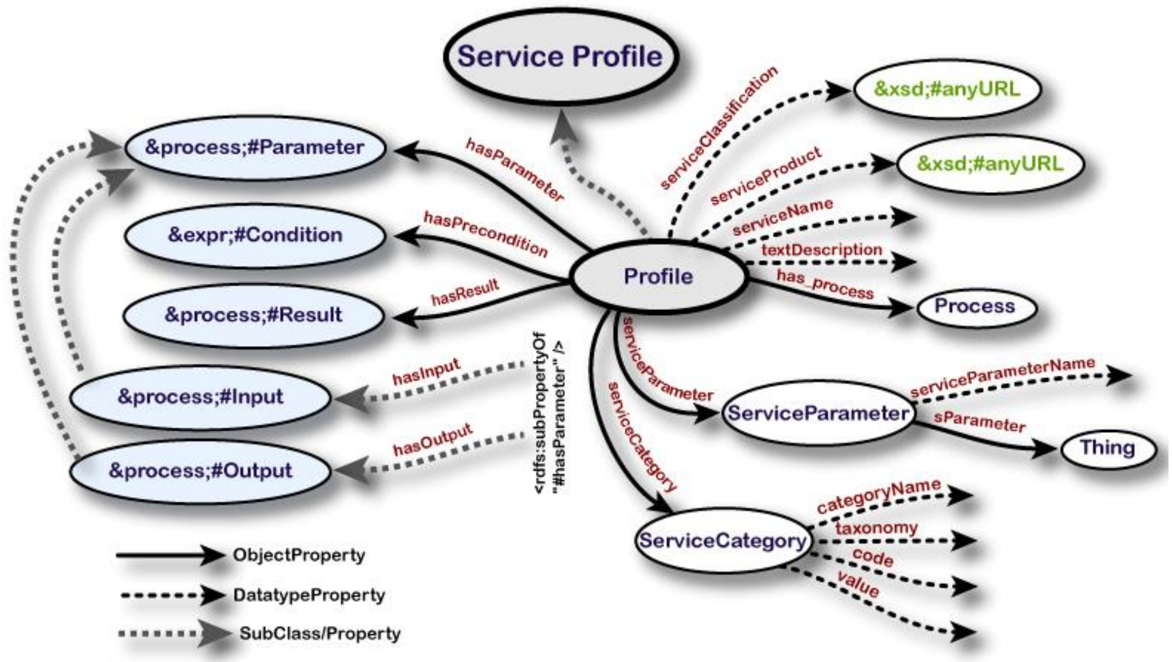

Figure 6.8: Selected classes and properties of a web service profile

6.2.4 HTML knowledge extraction

To represent a website composed of a set of HTML pages, we use the structure described in Figure 6.9. This structure allows us to have a view of the website structure. Using our model, we use as resource_entity: Class and as Linked entity Objectpropoerties. For this format, the object properties are: "is_LinkedTo" created to link the webpages and "has_Title" to specify the different titles in a webpage.

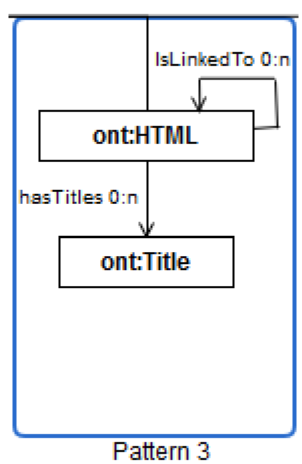

Figure 6.9: HTML transformation pattern

As an example of an algorithm that can transform a website into a resource representation, we present the algorithm 5.3. A Class is created containing the website name, and we create two objectProperties 1. Is_LinkedTo to connect the website pages and 2.hasTitle to inform about the different titles contained in a specific page. For each page, we create a specific class, and we linked the class to the main website via the ObjectProperty "prpo1" and then for each title of the page, we create a class that we linked to the page via the object property "prpo2".

```
Input: A knowledge resource W that represents a Website

Output: An OWL ontology that represents the same Website

Create a class W (if it does not exist yet)

for each Resource_Entity P that is linked to  through a Website_To_Page link

     Create a class P (if it does not exist yet)

         Create a property has_attribute_W_P

         Create an axiom  has_attribute_W_P domain W

         Create an axiom  has_attribute_W_P range P

         for each Resource_Entity  P' that is linked to P through a   Page_To_Page link

           Create a class P' (if it does not exist yet)

           Create a property has_attribute_P_P'

           Create an axiom  has_attribute_P_P' domain P

           Create an axiom  has_attribute_P_P' range P'
```

Algorithm 6.3: From HTML representation to OWL

6.2.5 Conclusion

In this section, we presented how using our model; we can transform knowledge resource into resources representations. As knowledge extraction is not the goal of this thesis, and to demonstrate the applicability of our model, we described four different formats transformation using simple algorithms. In the next chapter, we present real results of the algorithms used. From this section, we can conclude two major points: 1. that our approach can support a different pattern of resource transformation 2. Our model applies to different resources formats.

6.3 Network construction's operations

In the previous sections of this chapter, we described how we could use our model to transform knowledge resources into resources representation for different resource formats. In this section, we describe the processes behind the construct of the network of ontologies based on our Model. Our system can accept a new knowledge resource to add it to the network of ontologies and allows to access to the knowledge via a user interface designed for this purpose.

We demonstrate how our approach relates between 1. the resource representation and the knowledge resource, two resource representation's entities and 3.inter resources represent ions using the different compounds of our model.

Thus, we specified the different modules to describe the different operations involved in the life cycle of network construction: 1."Knowledge adding module" describes the process to add a new knowledge resource into the network.2."Network building Module". The algorithm specifies how we are linking the resources with each other and with the Profile Ontology.

6.3.1 Profile ontology- resources ontologies connection

Considering the formal definition, we define an algorithm to specify our construction strategy. We are using the Profile Ontology as another specific ontology that describes the Organization vocabulary. After being parsed, we align the new resource representation with the Profile Ontology. If the alignment process results in a non-empty alignment, then we save the concepts aligned using the Resource_Alignment_Manager, and we end up the process. This point of our approach means that the resource added contains entities expressed with the vocabulary of the organization or that the knowledge expressed is from the general concepts expressed within the Profile Ontology. If the alignment is empty then we try to align it with every ontology in the networks

```
Align the new resource ontology (O^new) with the Profile Ontology (O_p) to produce
an alignment A_1(O^new,O_p)
   if A_1(O^new,O_p) is not empty
      ResourceAlignmentManager.Save(A_1(O^new,O_p))
   else
    // Try to align O^r with the rest of the network ontologies: {O_1^r, ... O_n^r}
     for j = 1 to n
         A_j(O^new , O_j^r) = result of an alignment process between O^new and O_j^r
         if A_j(O^new, O_j^r) is not empty
             ResourceAlignmentManager.Save(A_j(O^new, O_j^r))
```

Algorithm 6.4: Network construction algorithm

6.3.2 Inter- ontologies connections

As described by the algorithm 6.4, in case of no correspondence between the concepts of the resource ontology and the profile Ontology, the system aligns the resource ontology with the rest of the resources 'ontologies already existed within the network. The concepts aligned will be saved using a

Resource_Alignment_Manager instance and the corresponding resources 'ontologies will be saved within the Network_Manager.

We then create two instances of Resource_Alignment_Manager class as per the Figure 5.9 where $\mathbf{A_1}$is the set of Resource_Alignment_Manager between the Profile Ontology and resource ontologies and $\mathbf{A_2}$is the set of Resource_Alignment_Manager between the resources ontologies.

6.3.3 Search Module

We described our approach for ontology network querying in the section (5.3). In this section, we present a search implementation using a keyword approach. Here, we reuse the$\mathbf{A_1}$ and $\mathbf{A_2}$described in the previous section. To find if the query terms are aligned with the profile or not form $\mathbf{A_1}$. In case that we find correspondence, we search in $\mathbf{A_2}$if any other ontologies correspond to our query terms.

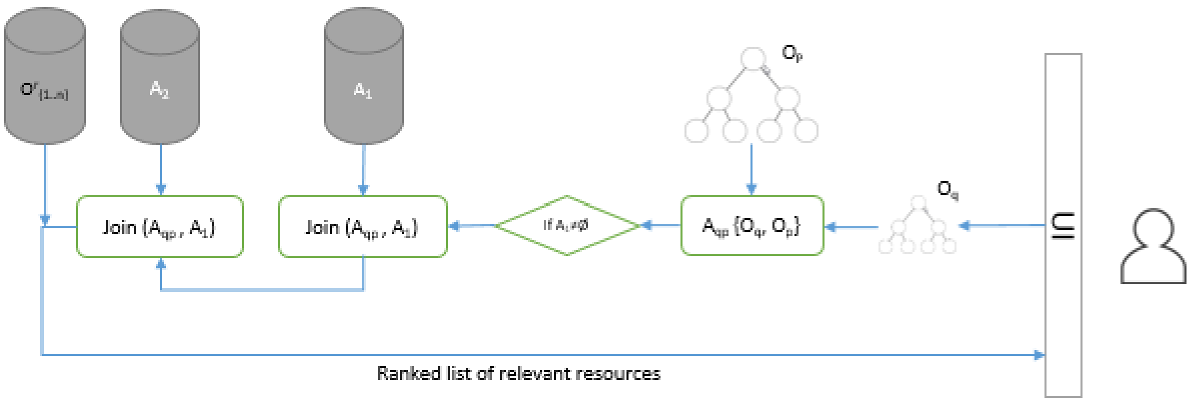

Figure 6.10: Accessing Network of ontologies workflow

Formally, the algorithm 6.2 of this module of our system is described as below:

- Parse (Q): is a procedure that takes the set of terms entered by a user and transform it into an ontology of concepts. Every term will be transformed into an OWL Class.
- AML-automatcher (O_q, O_p): is an alignment function that follows a pipeline of sub-function that we define in the next chapter.
- Join: is a function that output a list of the ontologies containing correspondence with the A_{qp}.

```
Parse(Q)
Aqp = AML − automatcher(Oq, Op)
Display(Join(Aqp, A1))
Display(Join(Aqp, A2))
```

Algorithm 6.5: Search algorithm

Our search approach can be easily extended to a hybrid approach using the keyword search approach, and navigation approach winch allows navigation inside a resource representation or between resources representations. As an example, the process can start with w key search operation that select different pertinent resource representation and then, a navigation process can allow users to have more idea about the resource structure (inside a resource representation) or the different representations connected with the selected resource.

6.4 Conclusion

In this chapter, we describe the implementation of our prototype system capable of adding a new resource into the network of ontologies by parsing the content of the resource into an OWL ontology schema following our model described in the previous chapter.

We described the system compounds, and we explained how the Profile Ontology can be a solution to homogenize the vocabulary of the network and how resource ontologies resulting from the parsing Module are related to it. We presented our approach for linking the nods of our network by defining our algorithm. A second algorithm has been described which can reuse the alignment done when adding new resources into the network to a semantically access to the knowledge represented within our conceptual schema.

Next, we evaluate how this software prototype supports the general framework by applying design science evaluation methods.

Chapitre 7. Use case and evaluation

The platform, as an instance of our general framework, demonstrates the feasibility of our design process and our designed artifacts. This software platform is an innovation in the way that it defines ideas, practices, and technical architecture through which we can analyze, design, and effectively implement conceptual knowledge schema. During the design process and the implementation, we aimed to demonstrate that our approach is realizable, implementable and usable. Consequently, the evaluation methods of our design artifact are chosen accordingly.

According to (Hevner & Chatterjee, 2010) “Information Technology (IT) artifacts can be evaluated regarding functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization, and other relevant quality attributes.” They suggest a knowledge base of five evaluation methods that should be matched appropriately with the designed IT artifact:

- Observational
- Analytical
- Experimental
- Testing
- Descriptive

To evaluate our approach, we use different and complementary patterns from the seven evaluation patterns suggested by (Vaishnavi, 2007). We use the experimentation pattern to assess and validate our solution in a real-life setting to prove the artifacts using the hermetic/inductive approach. We use metrics approach value the search approach.

Do so; we chose the empirical refinement alternative to develop our system. The selected pattern intent to develop a solution to a research problem through iterations of system development, empirical observation, and refinement. We assume that this pattern is applicable for our research problem as there is no specification on how to develop a system based on a network of ontologies model, and how we can query this type of knowledge model.

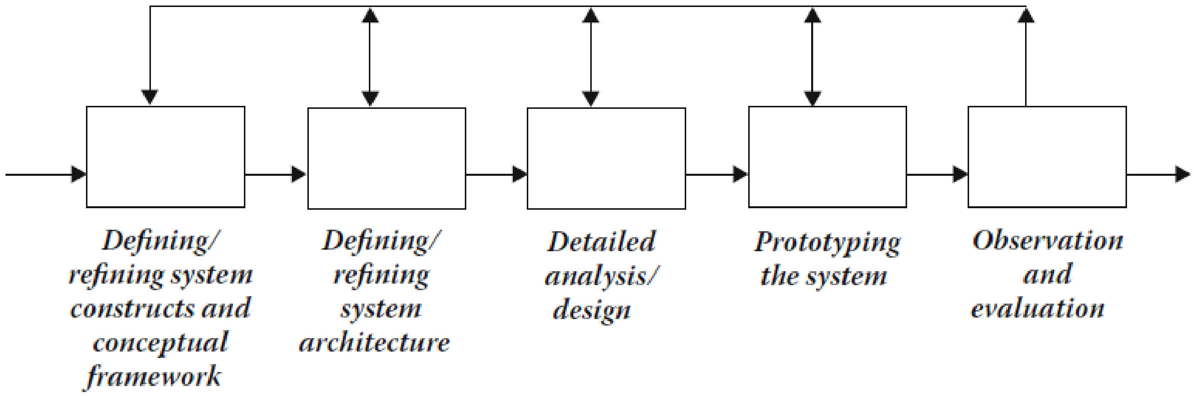

Figure 10.3 Empirical refinement. (Adapted from Nunamaker et al., *Journal of Management Information Systems* 7(3): 89–106, 1991.)

Figure 7.1: Empirical refinement pattern (Vaishnavi, 2007)

We create a prototype that implements the algorithms described in Chapters 6 and 5, and we applied it to an existing organization, which is the Geneva school of economics and management[73] (GSEM).

For the first evaluation approach, we compare the results with the expectations of the end-users and deduct its performance measurement. The second test is to use the performance measurements of the first test to compare the prototype with an indexation tool. To parameter the prototype for our specific organization, we developed a specific ontology describing its organization and its process. We then asked some of the GSEM employees to select some of the documents they produce to be integrated into our framework. The final step of this process is to query the system and to evaluate its answers. For the comparative with the indexing tool, we took the same documents selected by the organization employees, and we indexed them.

7.1 Organizational Knowledge Manager Framework

To support experimentation, we proceeded to the simulation approach (Navidi, Arnheim, & Waterman, 1992) which is a way of imitating the “inner” and the “outer” environments in the small, implementing the design using the imitated inner and external environments, and observing the behavior of the imitated system to understand and predict the behavior of the actual system (Vaishnavi, 2007). The aspects not supported by our simulation prototype is (1) that in real-world configuration, the system should be able to accept simultaneous access to the platform. Which was not implemented (2) not all the specificity of a resource format structure have been implemented. As knowledge extraction is not the goal of this work, we only implement simple algorithms to cover multiple resource formats. To process the simulation process, we

[73] https://www.unige.ch/gsem/

identify the compounds that we performed (Chapter 5) that imitate the compounds used in the real-life system.

The knowledge finder framework was programmed in Java and developed in Eclipse neon[74]. The core framework included main tree packages: The RessourceConvertModule, the network module, and the Search module. We use the AML_Matcher package to proceed with the alignment operations for the network building process and the search process.

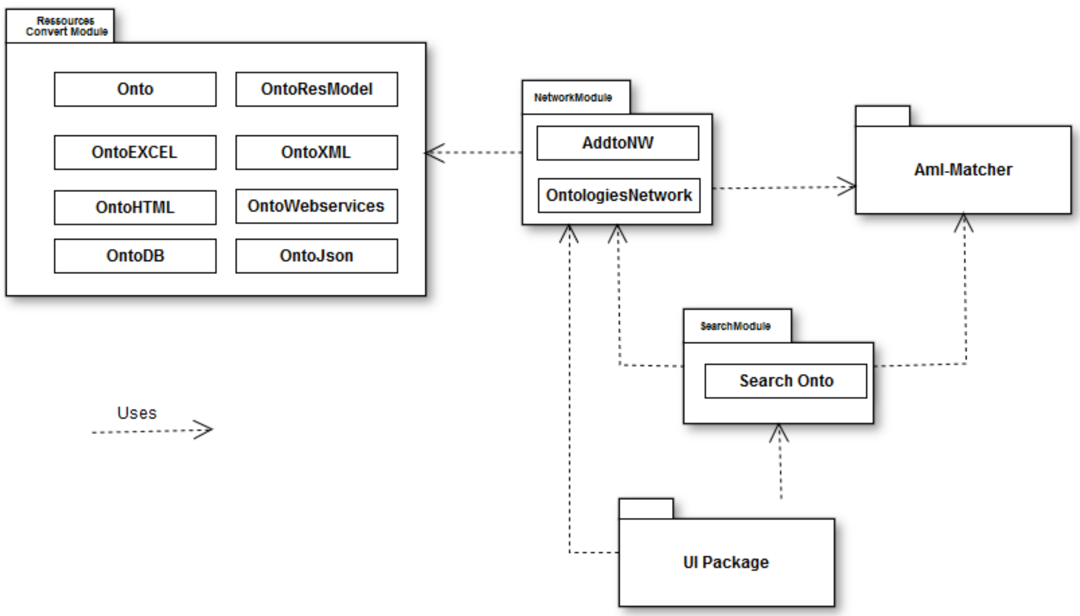

Figure 7.2: OKM Framework Packages

7.1.1 Resource Convert Module

To extract knowledge for different resource formats, we implement the resource convert module. This module is implanted based on our Model, The OntoResModel is a class representing the abstract form of a resource and for each type of resource, and we create a specific class to deal with the particular format of a resource. The Onto class was implemented to deal with the ontologies creation module (Class creation, ObjectProperty creation, DatapropertyType) which is composed of Onto class which implements ontological compounds as (ontologies, classes, datatype properties, object properties). The ResOntomodel is a class containing the metadata elements, which can be inherited by. For each resource format, we created a class to allow the knowledge extraction concerning the specificity of the structure.

To avoid non well-formed error, we added pre-treatment procedure that we applied the same for any of the resources that the result will be the same example (we extract Université de Genève) we replace “é” with

[74] https://github.com/RRNM/RRNM_FRAMWORK

“e” so even that this same string is obtained from the other resource the result will be the same, and we do not ..)

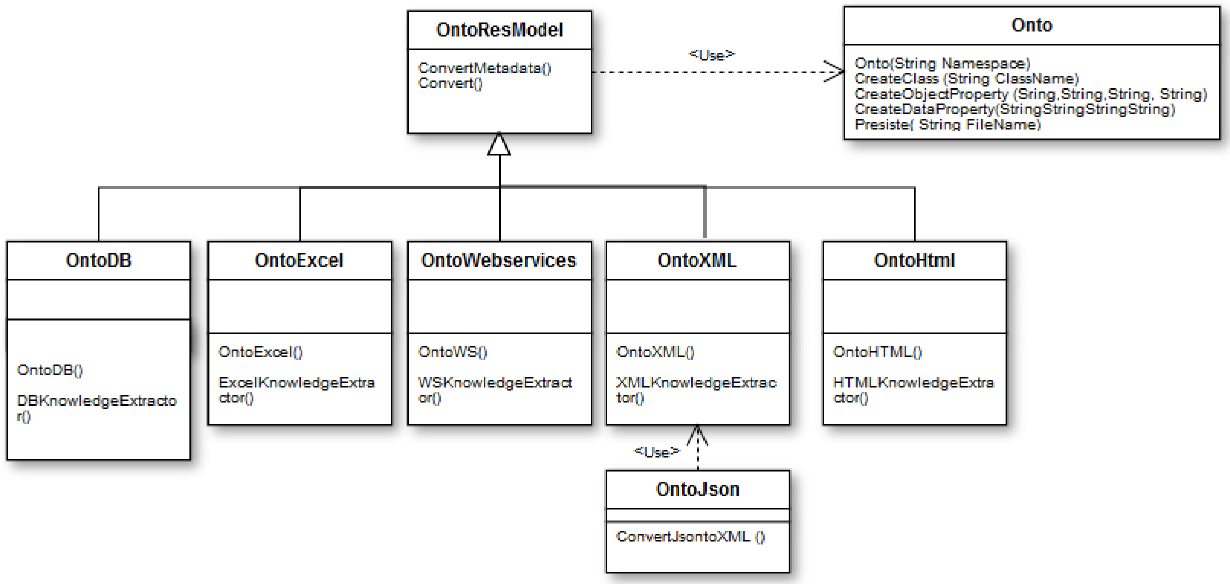

Figure 7.3: Resources converter package classes

7.1.2 Network package

To build and manage the network, we implemented the network package, which contains two classes: 1. the AddtoNetwork Class to add a new resource to the network via the algorithm 6.4. The ontology network that is implemented based on the Class OntologyNetwork of our model.

This package contains all the operation needed to 1. add a new resource to the network 2. save all the alignments between the new resource and the Profile Ontology and the other resources on the network.

7.1.3 Search package

The search package contains a class implementing the algorithm described in section 6.3.3. The search function takes as input a set of keywords and outputs a list of the paths of relevant resources and the specific relevant compound of the resource structure.

7.2 Experimentation setup

We set up our prototype with a real-world use case, which is GSEM University. The configuration is composed of two steps:

1. GSEM ontology: This ontology is built manually to describe the Faculty Organization. As ontology building is a time-consuming task, we used the Organizational ontology as a start point to identify the fundamental concepts, and we added a second layer more specific to match the specificity of the particular knowledge resource.

2. We asked the different department of the GSEM to give us access to a selection of their working documents.

7.2.1 GSEM ontology

As described, the GSEM ontology is the representation of the tacit knowledge that can hardly be extracted from explicit knowledge resource and which is essential to represent the knowledge using the organization logic and architecture.

Following the implementation methodology described in the section (5.2.2.1)

1. we identified the purpose of the ontology:
    1. To determine the structure of the GSEM, we asked for the Academic Organism that represents for the Organizational ontology the "Formal organization" and the Administrative Organism which represent them". For confidential, Figure 7.4 shows only the Academic Organism which is published publicly on the website of the GSEM.

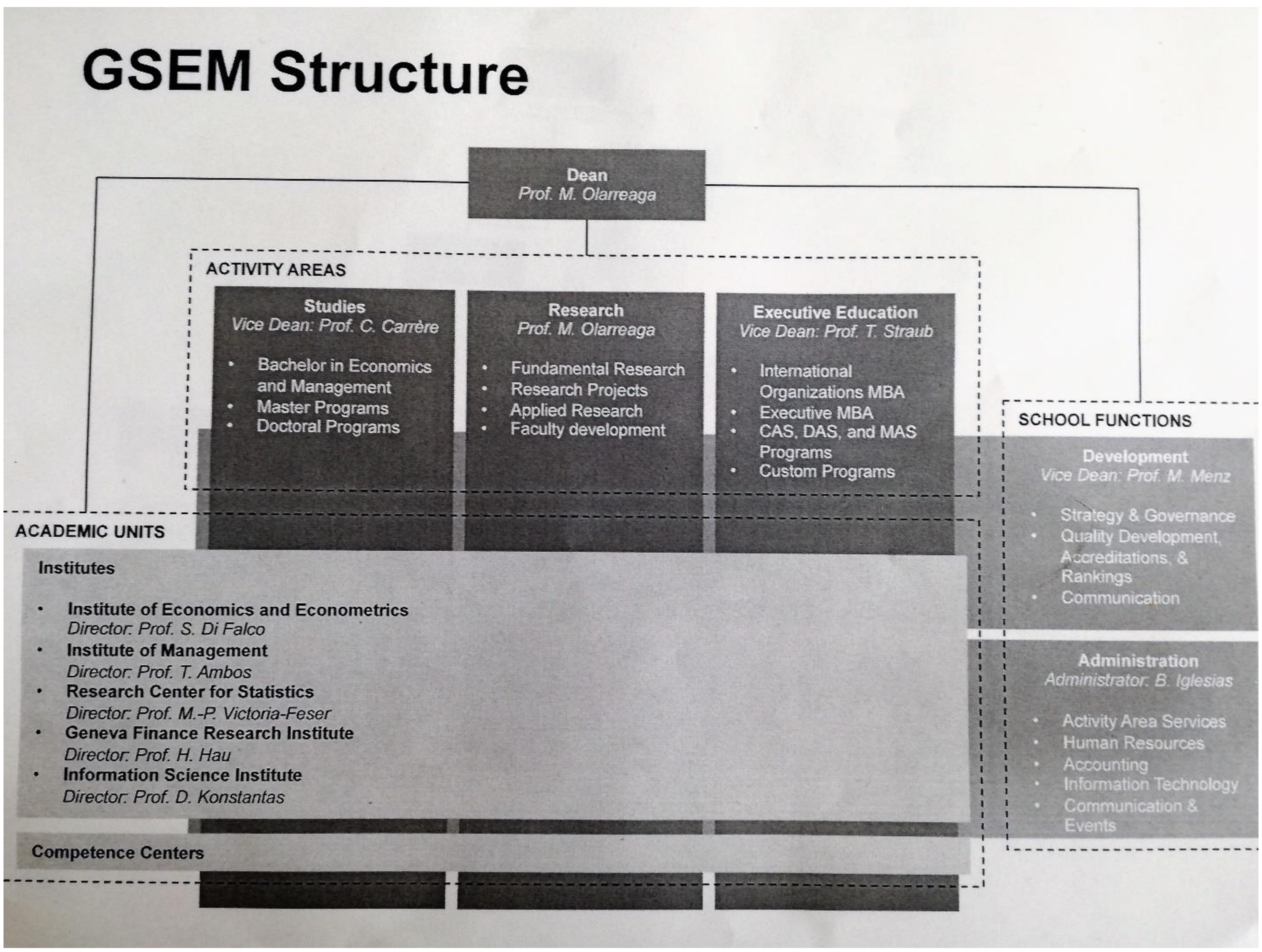

Figure 7.4: Academic Organism

2. we identified documents from where we can extract the exact definitions and vocabulary used within the company
    1. For this step, we based our investigation on the official website of the GSEM and internal confidential documentation. Examples of vocabulary that we took in consideration:
        1. Faculty = Professors
        2. Study Plan is not equal of Study Program
3. We interviewed the different services of the company to specify the process.

We selected relevant documents describing the different process, and then we enriched their metadata before adding them to the network.

4. We identified extra ontological resources that can enhance the enterprise ontology.
5. We defined the rules (axioms) relating the different concepts of the ontology.
6. We implement the ontology using Protégé
7. We then evaluate the ontology based on our final objective, which can connect the knowledge resources and present a full vocabulary to describe the organization.

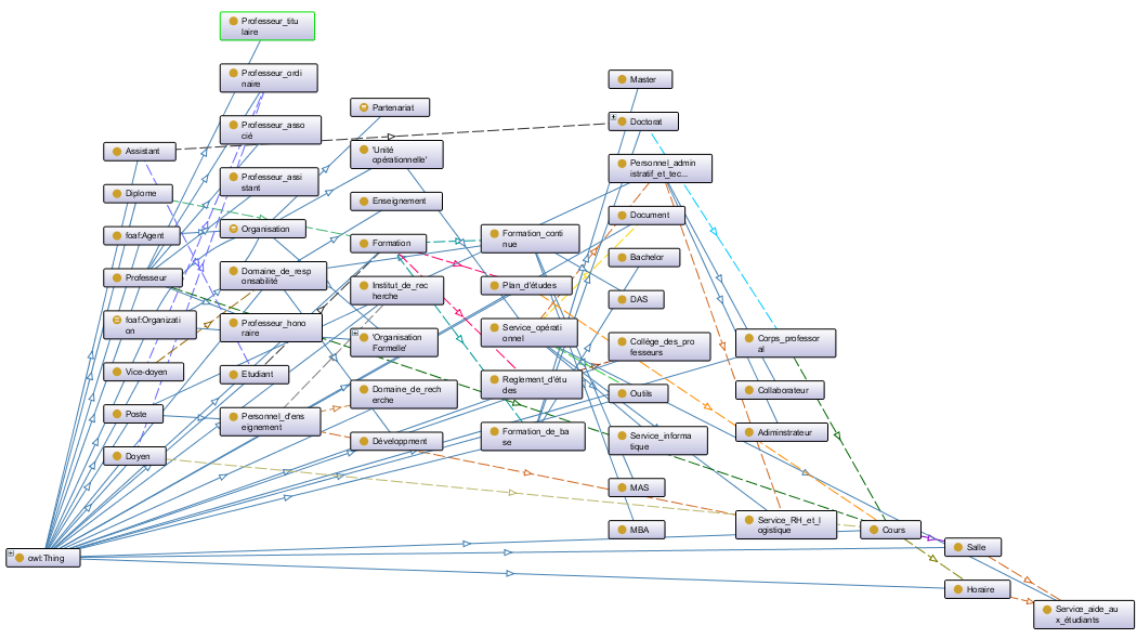

Figure 7.5: GSEM Profile Ontology

7.2.2 GSEM explicit resources

We explored a set of explicit resources. As we implement simple algorithms that do not cover all the specificity of the resource type. We described these algorithms in the first section of the previous chapter. Due to confidentiality reasons, we had access to a restricted set of resources. The Table 7.1 described the resources to which we have access to process the test.

Resource formats	Number	resources
Json	1	Widens.php
Data base	1	Phd.DB (database schema)
Website	1	GSEM website
Excel documents	9	Divers

Table 7.1: The test corpus

7.2.3 AML Alignment framework

For the implementation of our prototype, we choose AML-matching as alignment tool as it is complete, accessible and even if there is no specific documentation, the developer of the tool is available to answer any particular question. As it is an open source code, we modified multiple classes to AML tool is coded to align two specific ontologies. It proposes many alignment methodologies as structured alignment, property alignment, background alignment, and different matching techniques as offISub, Levenstein, Jaro-Winkle, and Q-gram.

We set up a pipeline for the alignment operation, which contains, Word matching, Structural matching, and property matching. Then, we test the four available techniques to choose the one that suits best our strategy. The AML alignment framework can detect if the two ontologies are in different languages, the system recognizes this difference and then translate the ontology to a specific language. This function uses Microsoft translation API.

For the background ontology, we choose WordNet as background ontology to detect synonyms.

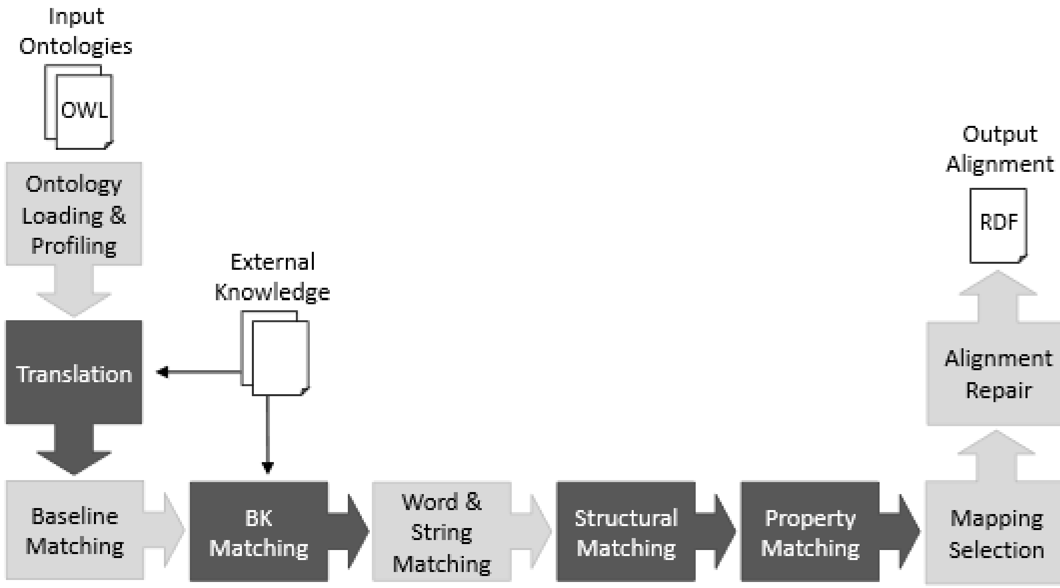

Figure 7.6: AML matching workflow (Faria, et al., 2013)

7.3 Experimentation process

To process the experimentation, we evaluate each part of our prototype separately, and then we assess the whole approach with a different system based on a different approach, which also aims to find knowledge from heterogeneous knowledge sources.

7.3.1 Knowledge Extraction process

To be able to evaluate the knowledge extraction process, we use mainly the OntoConvertModule described in section 7.1.1. The first step of this process is to extract the metadata and the structure of the resource and create an OWL resource representation. The OWL representations are generated using the algorithms described in the section (6.2). The table 7.2 details the result of the extraction of the different resource with a different format.

Resource	Type	Class count	Object Property
PHD DB	Database	94	89
http://www.unige.ch/gsem/fr/recherche/corps-professoral/tous/	Webpage	407	11876
Faculties	json	22	22
Units	Excel	38	68

Table 7.2: Knowledge extraction evaluation results

As expected, the structured resource formats gave reasonable conclusion regarding some classes per resource. The unstructured and semi-structured ones gave a significant number of classes, as we cannot detect which are the central concepts in the resource or the useless information as the concepts “O2” in the Units document shown in Figure 6.7.

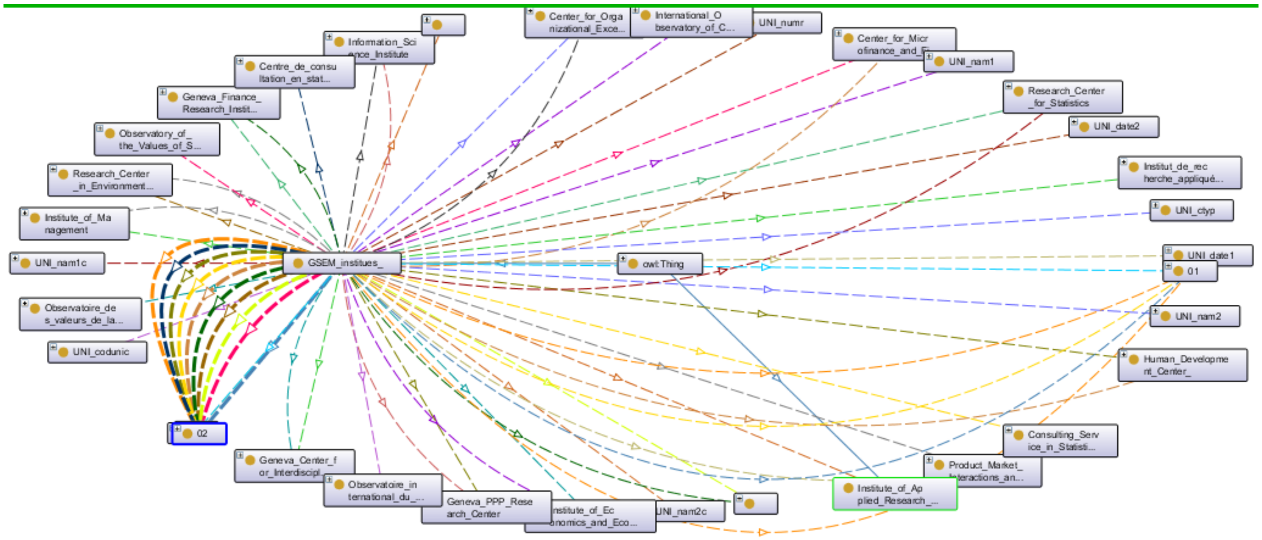

Figure 7.7: Excel extraction result

Nonetheless, this extraction allows us detecting if a specific term exists and which is its role in the resource. As per the example of Figure 7.6, that shows that “Institute of Applied Research in Economics and management” exist in the Row 16_Colum 6 of the GSEM_Institute document. Has a link to the page of “publication sur les filières” which is “https:\\www.unige.ch\\etudiants\ba

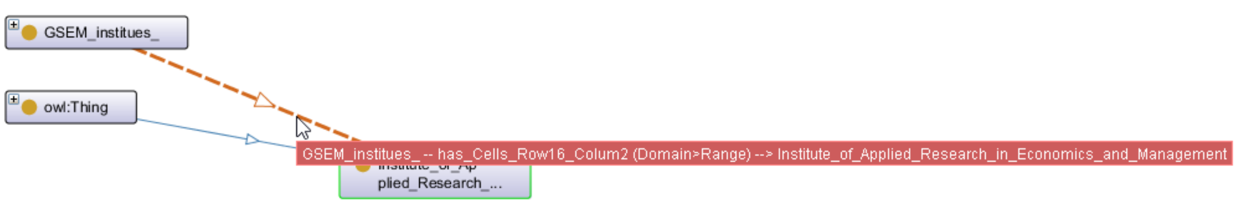

Figure 7.8: Structure detection result

Form this section we can conclude that the extraction step of our approach is not only feasible and straightforward but also extendable. We can quickly go further on one of the extraction processes to add new extraction rule or even use additional approaches as NPL or data mining for an advanced knowledge extraction module.

7.3.2 Network building process

To evaluate the Network building process, we test different alignment techniques available on AML-matcher package to check which one can connect the maximum of resources and concepts.

The Alignment techniques available on AML-matcher are: Jaro-Winkler (Jaro, 1995) similarity, which counts the common characters between two strings even if they are misplaced by a "short" distance ;the Q-Gram (Navarro, Sutinen, & Tarhio, 2005), which counts the number of tri-grams shared between the two strings, the sub-string distance (Arslan, 2006), which searches for the largest common substring and Levenstein distance[75], which counts the insertions an deletions needed to match two strings. To test these similarity measurements, we fixed the same criteria for the test:

- Translation: use_translator=True
- Background Knowledge: bk_sources=Wordnet
- Word Matcher: word_matcher=average
- String Matcher: string_matcher=global
- Structural Matcher: struct_matcher=descendants
- Property Matcher: match_properties=true

We applied the test on a corpus of 12 ontologies resulting from the Extraction process. For this corpus, the maximum possible Resource_level alignment is $\sum i_{i=1}^{12}$=66.

Similarity measurements	Entity_Level alignments	Resource_Level alignments
ISub	173	44
Levenstein	140	44
Jaro-Winkler	234	52
Q-gram	141	44
Jaro-Winkler without BK strategy	232	52

Table 7.3: Alignment techniques evaluation

As shown in table 7.3, The Jaro-Winkler similarity measurement aligned 234 entities from 2909 and 52 resources from 66 possible alignments. Using this similarity measurement, we can find more concepts matched which can better connect our network and then find relevant resources.

To test the contribution of the Background ontology, we examine the Jaro-Winkler similarity measurement while taking of the BK strategy parameter. As shown in the table, the background ontology contributes to finding more alignment between the resources.

[75] http://www.let.rug.nl/kleiweg/lev/levenshtein.html

From a quality point of view, we note that under a threshold= 0.65 the results are not relevant as shown in Figure 7.9.


```xml
<map>
    <Cell>
        <entity1 rdf:resource="http://www.ontologie.fr/prof#Programmes_d__tudes"/>
        <entity2 rdf:resource="http://www.ontologie.fr/monexcelplan#Plan_d__tudes"/>
        <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.7722</measure>
        <relation>=</relation>
    </Cell>
</map>

 <map>
    <Cell>
        <entity1 rdf:resource="http://www.ontologie.fr/monexcelLib#R_ussite_au_certificat_compl_mentaire_en_statistique_appliqu_e"/>
        <entity2 rdf:resource="http://www.ontologie.fr/prof#Certificat_compl_mentaire_en_statistique_appliqu_e"/>
        <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.7544</measure>
        <relation>=</relation>
    </Cell>
 </map>

<map>
    <Cell>
        <entity1 rdf:resource="http://www.ontologie.fr/monexcel#Geneva_Finance_Research_Institute"/>
        <entity2 rdf:resource="http://www.ontologie.fr/monexcel#Generalized_Linear_and_Additive_Models"/>
        <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.6269</measure>
        <relation>=</relation>
    </Cell>
</map>

<map>
    <Cell>
        <entity1 rdf:resource="http://www.ontologie.fr/monxmlfaculties#professor"/>
        <entity2 rdf:resource="http://www.semanticweb.org/ajmi/ontologies/2017/4/GSEMOntology#Professeur_associe"/>
        <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.7171</measure>
        <relation>=</relation>
    </Cell>
</map>
```


Figure 7.9: Alignment techniques evaluation

As described in section (6.3), our network building strategy is described by the algorithm (6.4). To compare our strategy to another strategy, we implemented an all to all strategy used in the system (Zhong, Zhu, Li, & Yu, 2002) and all to Op strategy described in the section (3.3.1.1). The table 7.4 details the result of our experimentation. Comparing to the first strategy, we can show using our strategy we obtained the sum of the resources aligned and the sum of the concepts aligned by the two first strategies by doing fewer alignment operations than the first strategy, which reduces the complexity of the first strategy and expands the results of the second one.

Alignment strategy	Alignment done	Resource_level alignments	Entity_level alignments
Strategy 1: all to all	66	52	234
Strategy 2: all to OP	12	12	68
Strategy 3: our Strategy	64	64	302

Table 7.4: Network building strategy evaluation

7.3.3 Knowledge search process

For this task, we test the knowledge search process. This evaluation aims to check the performance of our approach reusing alignment between knowledge resources ontologies. For that, we query the system using different keywords on a network of 12 resources ontologies resulting from the section (7.3.1). From the table 6.5 that shows examples of the results of the search process, we can conclude that:

1. Our knowledge search engine is precise as only relevant items have been selected (Precision =1).

2. The recall is zero when the keyword does not change much the Profile Ontology or if the keyword does not much any of the resource_representation ontologies. The average rate of the recall is 0.47%, which is very close to the results obtained by AML in OAEI 2017 [23].

3. F-score rate = 0.63

The issue of the point 2 comes from the translation process. That is because the Profile Ontology was created in English, and the keyword items use the French language.

Keyword	Keyword matching with Op	Items selected	Relevant items	Irrelevant items selected	Recall	Precision
"Bachelor"	yes	1	2	0	0.5	1
"Institute"	yes	2	2	0	1	1
"Etudiant"	no	0	7	0	0	1
"Student"	yes	0	7	0	0	1
"Faculties"	yes	3	3	0	1	1
"Cours"	yes	1	3	0	0.33	1

Table 7.5: Search process evaluation

AML has been the best performing system in Multifarm[76], but it does not even reach 50% F-measure. Using WordNet with the translation would expect to reverse the trend in precision vs. recall (in Multifarm ~70% precision / ~35% recall) but not improve F-measure. Thus, as a solution to this problem, we can use a different translation method or add concepts translations into the Op.

The answer of a keyword search is the list of the selected items and the exact structure item in the original resource. The Figure 7.10 shows an example of the "Institutes" keyword search answer. The Original

[76] http://oaei.ontologymatching.org/2014/multifarm/results/index.html

Keyword has a matching with the profile class “institute of economics and econometrics,” and then the process found the resources aligned with the Ontology Profile Class and gave the exact places where the found result occur.

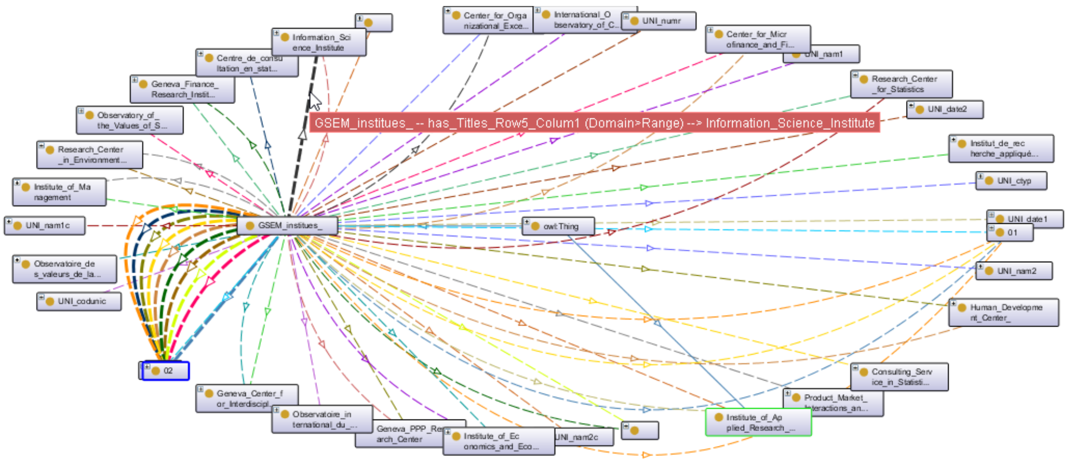

Figure 7.10: Example of the search process results

7.4 System evolution

Our prototype is constructed differently from compounds related following their role defined by our model. Thus, the development of the prototype can be done differently from the point of view depending on which part of the system we need to improve.

Form a technical perspective we can improve the knowledge extraction techniques or add new extraction techniques for additional resource formats. This improvement can be naturally inserted into the framework by updating or adding a new pattern to the ResourceConvert Package.

The alignment technique can be updated or replaced by more performer one. We can use a different alignment package, which offers new alignment techniques.

From the approach point of view, a significant evolution point is the evolution of the profile ontology. As discussed in the previous section, the ontology profile needs to be as complete as possible, which means to describe sufficiently the vocabulary of the organization that we can be able to find relevant resources. Thus possible improvement can be made by detecting concepts frequently aligned between resources and not existing or aligned in the Profile Ontology and then an expert can decide to enrich the Profile Ontology with them.

7.5 Conclusion

In this chapter, we evaluated our artifacts, which are the model we developed, the approach we took, and the framework. We did this by testing the different processes implemented to build an ontologies network based on heterogeneous resources and to have access to the knowledge contained in it. The results show that our model allows for a resource representation network by representing various resources in a unified way, maintaining their connections from entity level to resource level. The evaluation of our approach shows that we could add links between the different resources by applying fewer alignment operations than other methods. Our framework, using a pattern design approach, can easily be improved by using different alignment techniques or adding new extraction approaches.

By experimenting with the different processes, we show that creating an organizational memory by re-using the organization's daily basis resources is feasible. Any organization can re-use our work by merely creating the specific part of the Profile Ontology and providing access to the resources for re-use. Our work can also be used across domains in cases different to our real use one. Thus, one would need to create or use a specific Profile Ontology for each particular domain.

Chapitre 8. General conclusion

Creating an organizational memory is an amendatory process for any organization to preserve and re-use the organization's knowledge. Thus, organizations use different tools and systems to allow its employees to share and manage information. Nonetheless, the tools currently in use cannot collect all the knowledge that may be important to preserve. To create an organization memory is complicated, as it needs employees' commitment considering that in almost all cases, they need to re-enter all the information they possess manually using different and multiple tools and systems. These tools and systems generate heterogeneous resources that are difficult to manage.

Our work aims to create an organization memory automatically for a specific organization by reusing its employee's daily basis resources.

This thesis investigates how we can create an organizational knowledge memory based on a network of ontologies, and how we can access the corporate knowledge within the network. The knowledge graph needs to be constructed using the explicit resources of the organization. Thus, we investigate how we can represent structural and syntactically heterogeneous resources within the same network. After studying the existing approaches and methods, we created a MOF4 Model that enables us to represent heterogeneous resources from entity level and resource level. This model is able to express a network of resource representation by describing the connections between their entities and between the resources.

We then defined a semantic approach based on a different process that manages the creation of resource representation, its integration into the network, and its detection.

To evaluate our model and our approach, we implemented a framework composed of three modules identified as knowledge extraction, network building, and knowledge search. We set up our framework for a specific use case, and we tested the different processes.

Our work allows us to answer the different questions that we asked:

- How can one represent the syntactic and structural heterogeneity of organizational resources?

Using our model, we am capable of representing resources of different formats concerning their original structure and avoiding information loss. That is because our model is capable of describing the structure of any resource format using the resource entity and the linked entity, which together represent the link between any two entities of a resource. The alignment manager class represents the alignment between two entities of two resource representations, and then the network manager represents the alignment result of all the matching entities between two resource representations. An instance of the ontology network class is a representation of all the connections within the network. Thus, we can determine whether two resources are aligned or not, and if yes, we can see the specific entities aligned.

- Which tools or approaches can be used to extract knowledge from resources with different formats?

We investigate existing tools capable of obtaining information from different resource formats. For each format, we established a comparative analysis that concludes that different approaches can be taken to execute the process. We then demonstrate for four resource formats, how we can implement the transformation from an explicit resource to an OWL resource representation by extracting metadata, content, and structure.

- How do we align resource ontologies to construct a network of ontology?

Our approach explains the best way to create and access a network of ontologies. Thus, we defined an algorithm based on the hybrid approach that uses a Profile Ontology for describing the vocabulary of the network, and resource representation ontologies for representing each of the resources that we integrate into the network. When a new resource representation ontology is added into the network, it will be aligned first with the profile ontology, and then with the resource representation ontologies aligned with the profile of the same entity. A modification of the resource will not affect the profile ontology, as they are very independent. However, the Profile Ontology can be optimized using a list of recurrent concepts of the resource representation ontologies not aligned with the Profile Ontology.

- How can we access knowledge within a network of ontologies?

Using our model, we can re-use the alignment results from a resource level to detect whether two resource representations are connected. If they are, we can find out whether the entity aligned to correspond to a user keyword or not. The result of the search process is a list of relevant resources and the specific structural role of the entity in the original resource.

To evaluate our model and our approach, we implemented a framework to experiment with a real use case. Thus, we created a specific Profile Ontology for the GSEM faculty, and we imported a selection of resources to build a network of ontology for this particular organization.

The import processes are designed following the design pattern approach to facilitate adding other extraction methods. The alignment operations are done using the AML-matcher package, which is an open-source implementation of the best-ranked alignment method in the OAEI evaluation conference. Nonetheless, another one can easily replace this package. The search process was implemented using the Keyword approach, but as our methodology produced the relevant resources list and the exact roles of the entity searched in the original resources, we could also perform another search approach, in particular, the navigation approach.

Our framework can be the foundation for a more sophisticated knowledge network construction framework if we were to use advanced extraction methods based on a data-mining approach or a deep-learning approach, which could significantly improve the content of resource representation ontologies, and then the search performance of the framework as well.

Bibliography

Achichi, M., Cheatham, M., Dragisic, Z., Euzenat, J., Faria, D., Ferrara, A., . . . Zamazal, O. (2017, 10). Results of the Ontology Alignment Evaluation Initiative 2017. *OM: Ontology Matching*, (pp. 61-113). Wien. Retrieved from https://hal.archives-ouvertes.fr/hal-01670238

Adjiman, P., Goasdoué, F., & Rousset, M.-C. (2007). SomeRDFS in the Semantic Web. In S. Spaccapietra, P. Atzeni, F. Fages, M.-S. Hacid, M. Kifer, J. Mylopoulos, . . . I. Zaihrayeu (Ed.), *Journal on Data Semantics VIII* (pp. 158-181). Springer Berlin Heidelberg.

Ahlberg, C., & Shneiderman, B. (1994). Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. *Proceedings of the SIGCHwe Conference on Human Factors in Computing Systems* (pp. 313-317). New York, NY, USA: ACM. doi:10.1145/191666.191775

Ahmed, W., Aslam, M. A., Shen, J., & Yong, J. (2011, 6). A light weight approach for ontology generation and change synchronization between ontologies and source relational databases. *Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, (pp. 208-214). doi:10.1109/CSCWD.2011.5960076

Ajami, S., & Bagheri-Tadi, T. (2013). Barriers for Adopting Electronic Health Records (EHRs) by Physicians. *Acta Informatica Medica, 21*, 129-134. doi:10.5455/aim.2013.21.129-134

Ajmi, O., Ghoula, N., & Falquet, G. (2012). Méthodologie de construction d'outil d'aide à la décision pour le choix de techniques d'alignement à travers les évaluations. 1-3. Retrieved from https://archive-ouverte.unige.ch/unige:46137

Alaoui, L., Hajjamy, O. E., & Bahaj, M. (2014, 4). Automatic Mapping of Relational Databases to OWL Antology. *International Engineering Research Journal. Vol. 3 - Issue 4 (April- 2014).* IJERT. Retrieved from http://www.ijert.org/view-pdf/9397/automatic-mapping-of-relational-databases-to-owl-antology

Aleksovski, Z., Klein, M., Kate, W., & Harmelen, F. (2006). Matching Unstructured Vocabularies Using a Background Ontology. In S. Staab, & V. Svátek (Eds.), *Managing Knowledge in a World of Networks* (Vol. 4248, pp. 182-197). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/11891451_18

Amudaria, S., & Sasirekha, S. (2011, 7). Improving the precision ratio using semantic based search. *2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies*, (pp. 465-470). doi:10.1109/ICSCCN.2011.6024595

Arduin, P.-E., Grundstein, M., & Rosenthal-Sabroux, C. (2015). *Information and Knowledge System; Volume 2 – Advances in Information Systems* (Vol. 2). Paris: WIley.

Arduin, P.-E., Grundstein, M., & Rosenthal-Sabroux, C. (2015). *Système d'information et de connaissance.*

Arens, Y. I., Chin Y, C. H., Chun-Nsn, H. S., & Knoblock, C. R. (1993). Retrieving and Integrating Data from Multiple Information Sources. *International Journal on Intelligent and Cooperative Information Systems*, 34.

Arslan, A. (2006). An algorithm for string edit distance allowing substring reversals. *Sixth IEEE Symposium on BioInformatics and BioEngineering (BIBE'06)* (pp. 220-226). Arlington: IEEE. doi:10.1109/BIBE.2006.253338

Astrova, I., Korda, N., & Kalja, A. (2007). Rule-Based Transformation of SQL Relational Databases to OWL Ontologies. *In Proceedings of the 2nd International Conference on Metadata & Semantics Research.*

Averweg, U. R. (2009). Historical Overview of Decision Support Systems (DSS). *Encyclopedia of Information Science and Technology, Second Edition*, 1753-1758. doi:10.4018/978-1-60566-026-4.ch275

Badr, Y., Chbeir, R., Abraham, A., & Hassanien, A.-E. (2010, 6). *Emergent Web Intelligence: Advanced Semantic Technologies.* Springer Science & Business Media.

Bafoutsou, G., & Mentzas, G. (2002, 8). Review and functional classification of collaborative systems. *International Journal of Information Management, 22*, 281-305. doi:10.1016/S0268-4012(02)00013-0

Bailey, J., Bry, F., Furche, T., & Schaffert, S. (2009). Semantic Web Query Languages. *Encyclopedia of Database Systems*, 4. doi:https://doi.org/10.1007/978-0-387-39940-9_1317

Barros, V. F., Ramos, I., & Perez, G. (2015, 4). Information systems and organizational memory: A literature review. *JISTEM - Journal of Information Systems and Technology Management, 12*, 45-63. doi:10.4301/S1807-17752015000100003

Battle, S. (1999). *Gloze: XML to RDF and back again.* Tech. rep., HP Labs, Bristol, UK.

Bechhofer, S. (2008, 5). The Semantic Web: Research and Applications. Springer Science & Business Media.

Becker, C., & Duretec, K. (2013). Free Benchmark Corpora for Preservation Experiments: Using Model-driven Engineering to Generate Data Sets. *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 349-358). New York, NY, USA: ACM. doi:10.1145/2467696.2467719

Bohring, H., & Auer, S. (2005). Mapping XML to OWL Ontologies. *Leipziger Informatik-Tage, volume 72 of LNwe* (pp. 147-156). GI.

Bondiombouy, C., & Valduriez, P. (2016). Query processing in multistore systems: an overview. *IJCC, 5*, 309-346. doi:10.1504/IJCC.2016.10001884

Borlund, P. (2003, 8). The concept of relevance in IR. *Journal of the American Society for Information Science and Technology, 54*, 913-925. doi:10.1002/asi.10286

Bouziane, A., Bouchiha, D., Doumi, N., & Malki, M. (2015, 1). Question Answering Systems: Survey and Trends. *Procedia Computer Science, 73*, 366-375. doi:10.1016/j.procs.2015.12.005

Bruijn, J., Lara, R., Polleres, A., & Fensel, D. (2005). OWL DL vs. OWL flight: conceptual modeling and reasoning for the semantic Web. *05 Proceedings of the 14th international conference on World Wide Web*, 623. doi:10.1145/1060745.1060836

Buccella, A., Cechich R., A., & Brisaboa, N. (2005). Ontology-Based Data Integration Methods: A Framework for Comparison. *Revista Colombiana de Computacion*, 24.

Calvanese, D., Giacomo, G. D., & Lenzerini, M. (2002). Description Logics for Information Integration. In *Computational Logic: Logic Programming and Beyond* (pp. 41-60). Springer, Berlin, Heidelberg. doi:10.1007/3-540-45632-5_2

Caracciolo, C., Heguiabehere, J., Sini, M., & Keizer, J. (2009, 10). Networked Ontologies from the Fisheries Domain. *Metadata and Semantic Research* (pp. 306-311). Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-04590-5_29

Cerbah, F. (2008, 12). Mining the Content of Relational Databases to Learn Ontologies with Deeper Taxonomies. *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (pp. 553-557). IEEE. doi:10.1109/WIIAT.2008.382

Chardin, B., Coquery, E., Pailloux, M., & Petit, J. (2014, 12). RQL: A SQL-Like Query Language for Discovering Meaningful Rules. *2014 IEEE International Conference on Data Mining Workshop*, (pp. 1203-1206). doi:10.1109/ICDMW.2014.50

Chatterjee, I. (2010, 2). *MANAGEMENT INFORMATION SYSTEMS.* (P. H. Ltd., Ed.) PHwe Learning Pvt. Ltd.

Chbihwe Louhdi, M. R., Behja, H., & El Alaoui, S. O. (2013, 11). Transformation Rules for Building Owl Ontologies from Relational Databases. (pp. 271-283). Acadeour & Industry Research Collaboration Center (AIRCC). doi:10.5121/csit.2013.3822

Choudhary, L., & Burdak, B. S. (2012, 7). Role of Ranking Algorithms for Information Retrieval. *International Journal of Artificial Intelligence & Applications, 3*, 203-220. doi:10.5121/ijaia.2012.3415

Cimiano, P., Haase, P., Heizmann, J., Mantel, M., & Studer, R. (2008, 5). Towards portable natural language interfaces to knowledge bases – The case of the ORAKEL system. *Data & Knowledge Engineering, 65*, 325-354. doi:10.1016/j.datak.2007.10.007

Clough, P. (2015). Evaluation: Thinking Outside the (Search) Box. *Proceedings of the Forum for Information Retrieval Evaluation* (pp. 1-9). New York, NY, USA: ACM. doi:10.1145/2824864.2824890

Coyle, K. (2012, 5). Chapter 3: Metadata Elements. *Library Technology Reports, 48*, 15-26. Retrieved from https://journals.ala.org/index.php/ltr/article/view/4665

Crestani, F. (1995). Implementation and evaluation of a Relevance Feedback device based on Neural Networks. In J. Mira, & F. Sandoval (Ed.), *From Natural to Artificial Neural Computation* (pp. 597-604). Springer Berlin Heidelberg.

Cui, Z., & O'Brien, P. (2000). Domain Ontology Management Environment. *Proceedings of the 33rd Annual Hawaiwe International Conference on System Sciences. vol.1*, p. 9. Maui: IEEE Comput. Soc. doi:10.1109/HICSS.2000.926977

Cullot, N., & Yétongnon, K. (2007). DB2OWL : A Tool for Automatic Database-to-Ontology Mapping. *Proceedings of the Fifteenth Italian Symposium on Advanced Database Systems,.* Retrieved from /paper/DB2OWL-%3A-A-Tool-for-Automatic-Database-to-Ontology-Cullot-Ghawi/4af3614008a9339c8452248d88d8671c1db3af3b

Cur, O., & Blin, G. (2014). *RDF Database Systems: Triples Storage and SPARQL Query Processing* (1st ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Damljanovic, D., Agatonovic, M., & Cunningham, H. (2010). Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction. In L. Aroyo, G. Antoniou, E. Hyvönen, A. Teije, H. Stuckenschmidt, L. Cabral, & T. Tudorache (Ed.), *The Semantic Web: Research and Applications* (pp. 106-120). Springer Berlin Heidelberg.

Daniel, F., Papadopoulos, G. A., & Thiran, P. (2013, 7). *Mobile Web Information Systems: 10th International Conference, MobiWIS 2013, Paphos, Cyprus, August 26-29, 2013, Proceedings.* Springer.

Dargam, F., Hernández, J. E., Zaraté, P., Liu, S., Ribeiro, R., Delibašić, B., & Papathanasiou, J. (2014, 8). *Decision Support Systems IIwe - Impact of Decision Support Systems for Global Environments: Euro Working Group Workshops, EWG-DSS 2013, Thessaloniki, Greece, May 29-31, 2013, and Rome, Italy, July 1-4, 2013, Revised Selected and Extended Papers.* Springer.

Dartigues, C., Ghodous, P., Gruninger, M., Pallez, D., & Sriram, R. (2007, 6). CAD/CAPP Integration using Feature Ontology. *Concurrent Engineering, 15*, 237-249. doi:10.1177/1063293X07079312

Dau, F. (2011). Semantic Technologies for Enterprises. *Conceptual structures for discovering knowledge. 19th international conference on conceptual structures*, 18. doi:10.1007/978-3-642-22688-5_1

De Bruijn, J., Ehrig, M., Feier, C., Martín-Recuerda, F., Scharffe, F., & Weiten, M. (2006). Ontology mediation, merging and aligning. *Semantic web technologies*, 95-113. Retrieved from http://disi.unitn.it/ p2p/RelatedWork/Matching/mediation-chapter.pdf

Dehghani, R., & Ramsin, R. (2015, 6). Methodologies for developing knowledge management systems: an evaluation framework. *J of Knowledge Management, 19*, 682-710. doi:10.1108/JKM-10-2014-0438

Dieng, R., Corby, O., Giboin, A., & Ribière, M. (1999, 9). Methods and tools for corporate knowledge management. *International Journal of Human-Computer Studies, 51*, 567-598. doi:10.1006/ijhc.1999.0281

Dietz, J. L. (2006). *Enterprise Ontology: Theory and Methodology.* (S.-V. B. Heidelberg, Ed.) Berlin Heidelberg: Springer-Verlag. Retrieved from //www.springer.com/cn/book/9783540291695

Doing-Harris, K., Livnat, Y., & Meystre, S. (2015, 4). Automated concept and relationship extraction for the semi-automated ontology management (SEAM) system. *J Biomed Semantics, 6*. doi:10.1186/s13326-015-0011-7

Dulcic, Z., Pavlic, D., & Silic, I. (2012, 10). Evaluating the Intended Use of Decision Support System (DSS) by Applying Technology Acceptance Model (TAM) in Business Organizations in Croatia. *Procedia - Social and Behavioral Sciences, 58*, 1565-1575. doi:10.1016/j.sbspro.2012.09.1143

Edwards, J. S. (2016, 4). *The Essentials of Knowledge Management.* Springer.

Ehrig, M., & Sure, Y. (2004, 5). Ontology Mapping – An Integrated Approach. *The Semantic Web: Research and Applications* (pp. 76-91). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-25956-5_6

Ensan, F., & Bagheri, E. (2010). A Framework for Handling Revisions in Distributed Ontologies. *Proceedings of the 2010 ACM Symposium on Applied Computing* (pp. 1417-1422). New York, NY, USA: ACM. doi:10.1145/1774088.1774388

Euzenat, J. (2008). Algebras of Ontology Alignment Relations. In A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, & K. Thirunarayan (Eds.), *The Semantic Web - ISWC 2008* (Vol. 5318, pp. 387-402). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-88564-1_25

Euzenat, J. (2014). *The category of networks of ontologies.* Tech. rep., INRIA.

Euzenat, J. (2015, 11). Revision in networks of ontologies. *Artificial Intelligence, 228*, 195-216. doi:10.1016/j.artint.2015.07.007

Fahad, M. (2008). ERONTO: a tool for extracting ontologies from extended E/R diagrams - Semantic Scholar. *Intelligent Information Processing IV.* Retrieved from /paper/ERONTO%3A-a-tool-for-extracting-ontologies-from-E%2FR-Upadhyaya-Kumar/7a03da25da1c91316e3c1856e1ec6f0dccf01a57

Faria, D., Pesquita, C., Santos, E., Cruz, I. F., & Couto, F. M. (2014, 11). Automatic Background Knowledge Selection for Matching Biomedical Ontologies. *PLOS ONE, 9*, e111226. doi:10.1371/journal.pone.0111226

Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I., & Couto, F. M. (2013, 11). The AgreementMakerLight ontology matching system. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, (pp. 527-541). doi:10.1007/978-3-642-41030-7_38

Farquhar, A. D., Fikes, R. I., & Rice, J. A. (1997, 6). The Ontolingua Server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies, 46*, 707-727. doi:10.1006/ijhc.1996.0121

Fensel, D. (2013, 3). *Ontologies:A Silver Bullet for Knowledge Management and Electronic Commerce.* Springer Science and Business Media.

Ferdinand, M., Zirpins, C., & Trastour, D. (2004). Lifting XML Schema to OWL. In N. Koch, P. Fraternali, & M. Wirsing (Eds.), *Web Engineering* (Vol. 3140, pp. 354-358). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-27834-4_44

Fikes, R., Hayes, P., & Horrocks, I. (2004, 12). OWL-QL—a language for deductive query answering on the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web, 2*, 19-29. doi:10.1016/j.websem.2004.07.002

Fisher, M., & Dean, M. (2008). Automapper : Relational Database Semantic Translation using OWL and SWRL.

Fox, S., Karnawat, K., Mydland, M., Dumais, S., & White, T. (2005, 4). Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems, 23*, 147-168. doi:10.1145/1059981.1059982

Gagnon, M. (2007, 7). Ontology-based integration of data sources. *2007 10th International Conference on Information Fusion*, (pp. 1-8). doi:10.1109/ICIF.2007.4408086

Garvin, D. A. (1998). The Processes of Organization and Management. *MITSloan Management Review*.

Ghoula, N., Falquet, G., & Guyot, J. (2010, 8). TOK: A Meta-model and Ontology for Heterogeneous Terminological, Linguistic and Ontological Knowledge Resources. *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (pp. 297-301). Toronto: IEEE. doi:10.1109/WI-IAT.2010.204

Goh, C. H., Madnick, S. E., & Siegel, M. D. (1994). Context Interchange: Overcoming the Challenges of Large-scale Interoperable Database Systems in a Dynamic Environment. *Proceedings of the Third International Conference on Information and Knowledge Management* (pp. 337-346). New York, NY, USA: ACM. doi:10.1145/191246.191306

Goker, A., & Davies, J. (Eds.). (2009). *Information retrieval: searching in the 21st century.* Chichester, U.K: Wiley.

Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2009). NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology. *Proceedings of International Conference on Software, Services & Semantic technologies (S3T 2009)*, 8.

Gonzalez, R. V., & Martins, M. F. (2017, 6). Knowledge Management Process: a theoretical-conceptual research. *Gestão & Produção, 24*, 248-265. doi:10.1590/0104-530x0893-15

Gray, P. M., Preece, A., Fiddian, N. J., Gray, W. A., Bench-Capon, T. J., Shave, M. J., . . . Visser, P. R. (1997). KRAFT: knowledge fusion from distributed databases and knowledge bases. *Database and Expert Systems Applications. 8th International Conference, DEXA '97. Proceedings* (pp. 682-691). IEEE Comput. Soc. doi:10.1109/DEXA.1997.617411

Green, B. F., Wolf, A. K., Chomsky, C., & Laughery, K. (1961). Baseball: An Automatic Question-answerer. *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference* (pp. 219-224). New York, NY, USA: ACM. doi:10.1145/1460690.1460714

Grimnes, G. A., Sauermann, L., & Bernardi, A. (2009). The Personal Knowledge Workbench of the NEPOMUK Semantic Desktop. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, . . . E. Simperl (Eds.), *The Semantic Web: Research and Applications* (Vol. 5554, pp. 836-840). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-02121-3_66

Gruber, T. R. (1993, 6). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*, 199-220. doi:10.1006/knac.1993.1008

Gu, J., Chen, H., Yang, L., & Zhang, L. (2004, 9). OBSA: Ontology-Based Semantic Information Processing Architecture. *IEEE/WIC/ACM International Conference on Web Intelligence, 2004. Wwe 2004. Proceedings*, (pp. 607-610). doi:10.1109/WI.2004.10141

Gupta, S., Szekely, P., Knoblock, C. A., Goel, A., Taheriyan, M., & Muslea, M. (2015). Karma: A System for Mapping Structured Sources into the Semantic Web. *In: Simperl E. et al. (eds) The Semantic Web: ESWC 2012 Satellite Events. ESWC 2012. Lecture Notes in Computer Science,. 7540*, pp. 430-434. Berlin, Heidelberg: Springer,. doi:10.1007/978-3-662-46641-4_40

Hacherouf, M., Bahloul, S. N., & Cruz, C. (2015, 4). Transforming XML documents to OWL ontologies: A survey. *Journal of Information Science, 41*, 242-259. doi:10.1177/0165551514565972

Haizhou, F., & Kemafor, A. (2011). Effectively Interpreting Keyword Queries on RDF Databases with a Rear View. *Proceedings of the 10th International Conference on The Semantic Web - Volume Part we* (pp. 193-208). Berlin: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=2063016.2063030

Halevy, A. Y., Ashish, N., Bitton, D., Carey, M., Draper, D., Pollock, J., . . . Sikka, V. (2005). Enterprise information integration: successes, challenges and controversies. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (pp. 778-787). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=1066246

Han, L., Finin, T., Parr, C., Sachs, J., & Joshi, A. (2008, 10). RDF123: From Spreadsheets to RDF. *The Semantic Web - ISWC 2008* (pp. 451-466). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-88564-1_29

Harman, D. (2011). *Information Retrieval Evaluation* (1st ed.). Morgan & Claypool Publishers.

Hartmann, J., Palma, R., Sure, Y., Suárez-Figueroa, M. C., Haase, P., Gómez-Pérez, A., & Studer, R. (2005). Ontology Metadata Vocabulary and Applications. In R. Meersman, Z. Tari, & P. Herrero (Ed.), *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops* (pp. 906-915). Springer Berlin Heidelberg.

Hazber, M., Yang, J., & Jin, Q. (2010, 10). Towards Integration Rules of Mapping from Relational Databases to Semantic Web Ontology. *2010 International Conference on Web Information Systems and Mining*, *1*, pp. 335-339. doi:10.1109/WISM.2010.21

Hearst, M. A., & Stoica, E. (2009). NLP Support for Faceted Navigation in Scholarly Collections. *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries* (pp.

62-70). Stroudsburg: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1699750.1699760

He-ping, C., Lu, H., & Bin, C. (2008, 12). Research and Implementation of Ontology Automatic Construction Based on Relational Database. *2008 International Conference on Computer Science and Software Engineering*, *5*, pp. 1078-1081. doi:10.1109/CSSE.2008.1427

Hevner, A., & Chatterjee, S. (2010). Design Science Research in Information Systems. In *Design Research in Information Systems* (Vol. 22, pp. 9-22). Boston, MA: Springer US. doi:10.1007/978-1-4419-5653-8_2

Hirschman, L., & Gaizauskas, R. (2001, 12). Natural language question answering: the view from here. *Natural Language Engineering, 7*. doi:10.1017/S1351324901002807

Horowitz, D., & Kamvar, S. D. (2010). The Anatoour of a Large-scale Social Search Engine. *Proceedings of the 19th International Conference on World Wide Web* (pp. 431-440). New York, NY, USA: ACM. doi:10.1145/1772690.1772735

Ingwersen, P. (1992). *Information Retrieval Interaction.* London, UK: Taylor Graham Publishing.

Ioannidis, D., Tropios, P., Krinidis, S., Stavropoulos, G., Tzovaras, D., & Likothanasis, S. (2016, 12). Occupancy driven building performance assessment. *Journal of Innovation in Digital Ecosystems, 3*, 57-69. doi:10.1016/j.jides.2016.10.008

Isaac, A., Wang, S., Zinn, C., Matthezing, H., Meij, L., & Schlobach, S. (2009, 3). Evaluating Thesaurus Alignments for Semantic Interoperability in the Library Domain. *IEEE Intelligent Systems, 24*, 76-86. doi:10.1109/MIS.2009.26

Jaro, M. A. (1995, 4). Probabilistic linkage of large public health data files. *Statistics in Medicine, 14*, 491-498.

Jarrar, Y. F. (2002, 8). Knowledge management: learning for organisational experience. *Managerial Auditing Journal, 17*, 322-328. doi:10.1108/02686900210434104

Järvelin, K. (2011). IR Research: Systems, Interaction, Evaluation and Theories. *ACM SIGIR Forum, 45*, 15.

Jennex, M. E., & Olfman, L. (2004). Organizational Memory. In P. C. Holsapple (Ed.), *Handbook on Knowledge Management 1* (pp. 207-234). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-540-24746-3_11

Jiang, X., & Tan, A.-H. (2009). CRCTOL: A semantic-based domain ontology learning system. *Journal of the American Society for Information Science and Technology, 61*, 150-168. doi:10.1002/asi.21231

Jiménez-Ruiz, E., & Grau, B. C. (2011, 10). LogMap: Logic-Based and Scalable Ontology Matching. *The Semantic Web – ISWC 2011* (pp. 273-288). Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-25073-6_18

Jung, J. J. (2009, 10). Semantic business process integration based on ontology alignment. *Expert Systems with Applications, 36*, 11013-11020. doi:10.1016/j.eswa.2009.02.086

Kefi, H., Safar, B., & Reynaud, C. (2006). Alignement de taxonomies pour l'interrogation de sources d'information hétérogènes. *Quinzième congrès francophone Reconnaissance des Formes et Intelligence Artificielle, 2*, 126. Retrieved from https://hal.inria.fr/inria-00001091/document

Keng, S. (2011, 12). *Cross-Disciplinary Models and Applications of Database Management: Advancing Approaches: Advancing Approaches.* IGwe Global.

Kim, K.-Y., Manley, D. G., & Yang, H. (2006, 12). Ontology-based Assembly Design and Information Sharing for Collaborative Product Development. *Comput. Aided Des., 38*, 1233-1250. doi:10.1016/j.cad.2006.08.004

Klein, M. (2002, 9). Interpreting XML documents via an RDF schema ontology. *Proceedings. 13th International Workshop on Database and Expert Systems Applications*, (pp. 889-893). doi:10.1109/DEXA.2002.1046008

Koenemann, J., & Belkin, N. J. (1996). A case for interaction: a study of interactive information retrieval behavior and effectiveness. *Proceedings of the SIGCHwe conference on Human factors in* *computing systems common ground - CHwe '96* (pp. 205-212). Vancouver, British Columbia, Canada: ACM Press. doi:10.1145/238386.238487

Kontokostas, D., Bratsas, C., Auer, S., Hellmann, S., Antoniou, I., & Metakides, G. (2012, 9). Internationalization of Linked Data: The case of the Greek DBpedia edition. *Web Semantics: Science, Services and Agents on the World Wide Web, 15*, 51-61. doi:10.1016/j.websem.2012.01.001

Krahmer Emiel, T. M. (2010). *Empirical Methods in Natural Language Generation: Data-oriented Methods and Empirical Evaluation.* (Springer-Verlag, Ed.) Berlin Heidelberg: Springer-Verlag. Retrieved from //www.springer.com/gp/book/9783642155727

Krivine, S., Nobécourt, J., Soualmia, L., Cerbah, F., & Duclos, C. (2009). Construction automatique d'ontologie à partir de bases de données relationnelles: application au médicament dans le domaine de la pharmacovigilance. *IC 2009*, (p. pp). Retrieved from https://hal.archives-ouvertes.fr/hal-00377455/

Kutz, O., Hois, J., Bao, J., & Grau, B. C. (2010). *Modular Ontologies: Proceedings of the Fourth International Workshop (WoMO 2010).* IOS Press.

Lee, J.-H., & Suh, H.-W. (2008, 12). Ontology-based Multi-layered Knowledge Framework for Product Lifecycle Management. *Concurrent Engineering, 16*, 301-311. doi:10.1177/1063293X08100030

Lehner, F., & Fteimi, N. (2016, 10). *Knowledge Science, Engineering and Management: 9th International Conference, KSEM 2016, Passau, Germany, October 5-7, 2016, Proceedings.* (Springer, Ed.) Springer.

Li, J., Zhang, L., & Yu, Y. (2001). Learning to Generate Semantic Annotation for Domain Specific Sentences. In the Workshop on Knowledge Markup and Semantic Annotation. *the First International Conference on Knowledge Capture (K-CAP 2001).* Canada: Victoria B.C.

Li, M., Du, X.-Y., & Wang, S. (2005, 8). Learning ontology from relational database. *2005 International Conference on Machine Learning and Cybernetics*, *6*, pp. 3410--3415 Vol. 6. doi:10.1109/ICMLC.2005.1527531

Liddle, S. W., Schewe, K.-D., Tjoa, A. M., & Zhou, X. (2012, 8). *Database and Expert Systems Applications: 23rd International Conference, DEXA 2012, Vienna, Austria, September 3-6, 2012, Proceedings.* Springer.

Linckels, S., & Meinel, C. (2011, 4). *E-Librarian Service: User-Friendly Semantic Search in Digital Libraries.* Springer Science & Business Media.

López, F. (2002). Overview Of Methodologies For Building Ontologies. *The Knowledge Engineering Review*, 13.

Lopez, V., Uren, V., Sabou, M., & Motta, E. (2011, 1). Is Question Answering fit for the Semantic Web?: A survey. *Semantic Web, 2*, 125-155. doi:10.3233/SW-2011-0041

Lopez, V., Uren, V., Sabou, M., & Motta, E. (2011, 1). Is Question Answering fit for the Semantic Web?: A survey. *Semantic Web, 2*, 125-155. doi:10.3233/SW-2011-0041

Lubit, R. (2001, 12). The keys to sustainable competitive advantage: Tacit knowledge and knowledge management:. *Organizational Dynamics, 29*, 164-178. doi:10.1016/S0090-2616(01)00026-2

Mahmood, A. R., Aly, A. M., & Aref, W. G. (2017, 9). FAST: Frequency-Aware Spatio-Textual Indexing for In-Memory Continuous Filter Query Processing. *arXiv:1709.02529 [cs]*. Retrieved from http://arxiv.org/abs/1709.02529

Matsokis, A. (2010). *An Ontology-Based Approach for Closed-Loop Product Lifecycle Management.* Lausanne: EPFL.

Meersman, R., Panetto, H., Dillon, T., Eder, J., Bellahsene, Z., Ritter, N., . . . Dou, D. (2013, 9). On the Move to Meaningful Internet Systems: . *OTM 2013 Conferences: Confederated International Conferences.* Springer.

Mena, E., Kashyap, V., Sheth, A., & Illarramendi, A. (1996, 6). Observer: an approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Proceedings*

First IFCIS International Conference on Cooperative Information Systems, (pp. 14-25). doi:10.1109/COOPIS.1996.554955

Minaee, S., & Liu, Z. (2017, 8). Automatic Question-Answering Using A Deep Similarity Neural Network. *arXiv:1708.01713 [cs]*. Retrieved from http://arxiv.org/abs/1708.01713

Morris, M. R., & Factors, H. (2007). *Collaborating Alone and Together: Investigating Persistent and Multi-User Web Search Activities. Microsoft Research.* Tech. rep.

Morsey, M., Lehmann, J., Auer, S., Stadler, C., & Hellmann, S. (2012). DBpedia and the Live Extraction of Structured Data from Wikipedia. *Program: Electronic Library and Information Systems, 46*, 157-181. doi:10.1108/00330331211221828

Munir, K., & Sheraz Anjum, M. (2017, 8). The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics*. doi:10.1016/j.aci.2017.07.003

Myroshnichenko, I., & Murphy, M. C. (2009, 9). Mapping ER Schemas to OWL Ontologies. *2009 IEEE International Conference on Semantic Computing*, (pp. 324-329). doi:10.1109/ICSC.2009.61

Navarro, G., Sutinen, E., & Tarhio, J. (2005, 6). Indexing text with approximate q-grams. *Journal of Discrete Algorithms, 3*, 157-175. doi:10.1016/j.jda.2004.08.003

Navidi, W., Arnheim, N., & Waterman, M. S. (1992, 2). A multiple-tubes approach for accurate genotyping of very small DNA samples by using PCR: statistical considerations. *American Journal of Human Genetics, 50*, 347-359.

Nejdl, W. (2002). Semantic Web and Peer-to-Peer Technologies for Distributed Learning Repositories. In M. A. Musen, B. Neumann, & R. Studer (Eds.), *Intelligent Information Processing* (Vol. 93, pp. 33-50). Boston, MA: Springer US. doi:10.1007/978-0-387-35602-0_5

Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R. W., & Musen, M. A. (2001, 3). Creating Semantic Web contents with Protege-2000. *IEEE Intelligent Systems, 16*, 60-71. doi:10.1109/5254.920601

Nyulas, C., O’Connor, M., & Tu, S. (2007). *DataMaster – a Plug-in for Importing Schemas and Data from Relational Databases into Protégé.* Tech. rep., Stanford Medical Informatics, Stanford University School of Medicine, Stanford.

O’Connor, M. J., Halaschek-Wiener, C., & Musen, M. A. (2010). Mapping Master: A Flexible Approach for Mapping Spreadsheets to OWL. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, . . . B. Glimm (Eds.), *The Semantic Web – ISWC 2010* (Vol. 6497, pp. 194-208). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-17749-1_13

O'Brien, H. L., & Toms, E. G. (2010, 1). The development and evaluation of a survey to measure user engagement. *Journal of the American Society for Information Science and Technology, 61*, 50-69. doi:10.1002/asi.21229

Oren, E., Moller, K. H., Scerri, S., Handschuh, S., & Sintek, M. (2006). What are Semantic Annotations? 14.

Päivärinta, T., & Munkvold, B. E. (2005). Enterprise Content Management: An Integrated Perspective on Information Management. *th Hawaiwe International Conference on System Sciences*, 10.

Pang, B., Knight, K., & Marcu, D. (2003). Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences. *In Proceedings of HLT/NAACL.*

Paredes-Moreno, A., Martínez-López, J. F., & Schwartz, D. G. (2010, 11). A methodology for the semi-automatic creation of data-driven detailed business ontologies. *Information Systems, 35*, 758-773. doi:10.1016/j.is.2010.03.002

Parent, C., & Spaccapietra, S. (2000). Database Integration: the Key to Data Interoperability. *Advances in Object-Oriented Data Modeling*. Retrieved from https://infoscience.epfl.ch/record/99123

Pattuelli, C., & Rubinow, S. (2013, 10). The knowledge organization of DBpedia: a case study. *Journal of Documentation, 69*, 762-772. doi:10.1108/JD-07-2012-0084

Peltonen, J., & Kaski, S. (2011, 6). Generative Modeling for Maximizing Precision and Recall in Information Visualization. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, (pp. 579-587). Retrieved from http://proceedings.mlr.press/v15/peltonen11a.html

Perez, G., & Ramos, I. (2013, 12). Understanding Organizational Memory from the Integrated Management Systems (ERP). *Journal of Information Systems and Technology Management, 10*, 541-560. Retrieved from http://www.jistem.fea.usp.br/index.php/jistem/article/view/10.4301%252FS1807-17752013000300005

Pérez-Ramos, I. M., Rodríguez-Calcerrada, J., Ourcival, J. M., & Rambal, S. (2013, 4). Quercus ilex recruitment in a drier world: A multi-stage demographic approach. *Perspectives in Plant Ecology, Evolution and Systematics, 15*, 106-117. doi:10.1016/j.ppees.2012.12.005

Pillania, R. K. (2005, 8). Leveraging Knowledge: Indian Industry's Expectations and Shortcomings. *Global Business Review, 6*, 231-249. doi:10.1177/097215090500600204

Pinkel, C. (2016). *i3MAGE: Incremental, Interactive, Inter-Model Mapping Generation.* Doctoral dissertation, Universität Mannheim. Retrieved from https://ub-madoc.bib.uni-mannheim.de/40946

Pirolli, P., & Rao, R. (1996). Table Lens as a Tool for Making Sense of Data. *AVwe '96 Proceedings of the workshop on Advanced visual interfaces.*

Poli, R., Healy, M., & Kameas, A. (2010, 9). *Theory and Applications of Ontology: Computer Applications.* Springer Science & Business Media.

Premkumar, V., Krishnamurty, S., Wileden, J. C., & Grosse, I. R. (2014, 1). A semantic knowledge management system for laminated composites. *Advanced Engineering Informatics, 28*, 91-101. doi:10.1016/j.aei.2013.12.004

Rajpathak, D., Chougule, R., & Bandyopadhyay, P. (2012, 6). A domain-specific decision support system for knowledge discovery using association and text mining. *Knowledge and Information Systems, 31*, 405-432. doi:10.1007/s10115-011-0409-1

Ramkumar, A. S., & Poorna, B. (2014, 3). Ontology Based Semantic Search: An Introduction and a Survey of Current Approaches. *2014 International Conference on Intelligent Computing Applications*, (pp. 372-376). doi:10.1109/ICICA.2014.82

Rao, R., & Card, S. K. (1994). The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. *Proceedings of the SIGCHwe Conference on Human Factors in Computing Systems* (pp. 318-322). New York, NY, USA: ACM. doi:10.1145/191666.191776

Reif, G., & Gall, H. (2009). Using weESA to Semantically Annotate Cocoon Web Applications. *Proceedings of the 14th international conference on World Wide Web*, (p. 7).

Remenyi, D. (2007). *ECKM2007-Proceedings of the 8th European Conference on Knowledge Management: ECKM.* Academic Conferences Limited.

Rexhepi, G., Ibraimi, S., & Veseli, N. (2013, 4). Role of Intellectual Capital in Creating Enterprise Strategy. *Procedia - Social and Behavioral Sciences, 75*, 44-51. doi:10.1016/j.sbspro.2013.04.006

Robertson, S. (2008, 8). On the history of evaluation in IR. *Journal of Information Science, 34*, 439-456. doi:10.1177/0165551507086989

Rosé, C. P., & Hall, B. S. (2004). A Little Goes a Long Way: Quick Authoring of Semantic Knowledge. *Proceedings of the 2nd International Workshop on Scalable Natural Language Understanding (ScaNaLU 2004) at HLT-NAACL 2004* (pp. 17–24). Boston, Massachusetts, USA: Association for Computational Linguistics.

Rowley, J., & Hartley, R. (2017, 5). *Organizing Knowledge: An Introduction to Managing Access to Information.* Routledge.

Sabou, M., D'Aquin, M., & Motta, E. (2006). Using the Semantic Web As Background Knowledge for Ontology Mapping. *Proceedings of the 1st International Conference on Ontology Matching - Volume 225* (pp. 1-12). Aachen: CEUR-WS.org. Retrieved from http://dl.acm.org/citation.cfm?id=2889633.2889634

Safar, B., Reynaud, C., & Calvier, F. (2007). Techniques d'alignement d'ontologies basées sur la structure d'une ressource complémentaire. *1ères Journées Francophones sur les Ontologies (JFO 2007)*, 21-35. Retrieved from http://abiteboul.com/gemoReports/GemoReport-482.pdf

Sanderson, M. (2010, 6). Test Collection Based Evaluation of Information Retrieval Systems. *INR, 4*, 247-375. doi:10.1561/1500000009

Santoso, H. A., Haw, S.-C., & Abdul-Mehdi, Z. T. (2011, 4). Ontology extraction from relational database: Concept hierarchy as background knowledge. *Knowledge-Based Systems, 24*, 457-464. doi:10.1016/j.knosys.2010.11.003

Satapathy, S. C., Avadhani, P. S., & Abraham, A. (2011, 12). *Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (India 2012) held in Visakhapatnam, India, January 2012.* Springer Science & Business Media.

Sedbrook, T. A. (2012, 12). Modeling the REA Enterprise Ontology with a Domain Specific Language. *Journal of Emerging Technologies in Accounting, 9*, 47-70. doi:10.2308/jeta-50406

Serafini, L., & Tamilin, A. (2005). DRAGO: Distributed Reasoning Architecture for the Semantic Web. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, . . . J. Euzenat (Eds.), *The Semantic Web: Research and Applications* (Vol. 3532, pp. 361-376). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/11431053_25

Shah, C. (2010, 1). Collaborative Information Seeking: A Literature Review. In A. Woodsworth (Ed.), *Advances in Librarianship* (Vol. 32, pp. 3-33). Emerald Group Publishing Limited. doi:10.1108/S0065-2830(2010)0000032004

Shankar, R., & Gupta, A. (2005, 10). Towards framework for knowledge management implementation. *Knowledge and Process Management, 12*, 259-277. doi:10.1002/kpm.234

Shneiderman, B. (1992, 1). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics, 11*, 92-99. doi:10.1145/102377.115768

Shvaiko, P., & Euzenat, J. (2013, 1). Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering, 25*, 158-176. doi:10.1109/TKDE.2011.253

Sicilia, M.-A., & Mora, M. (2010, 6). On Using the REA Enterprise Ontology as a Foundation for Service System Representations. *Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science* (pp. 135-147). Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-16496-5_10

Sieg, A., Mobasher, B., & Burke, R. (2007). Web Search Personalization with Ontological User Profiles. *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management* (pp. 525-534). New York, NY, USA: ACM. doi:10.1145/1321440.1321515

Smallwood, R. F. (2013, 4). *Managing Electronic Records: Methods, Best Practices, and Technologies.* John Wiley & Sons.

Smucker, M. D., & Clarke, C. L. (2012). Time-based Calibration of Effectiveness Measures. *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 95-104). New York, NY, USA: ACM. doi:10.1145/2348283.2348300

Solimando, A., Jiménez-Ruiz, E., & Pinkel, C. (2014). Evaluating Ontology Alignment Systems in Query Answering Tasks. *Proceedings of the 2014 International Conference on Posters & Demonstrations Track - Volume 1272* (pp. 301-304). Aachen: CEUR-WS.org. Retrieved from http://dl.acm.org/citation.cfm?id=2878453.2878529

Sposito, F. (2007). CSCW , Groupware and Social Issues Applications, analytical perspectives and possible solutions in an ever-changing and critical study field.

Sposito, F. (n.d.). CSCW, Groupware and Social Issues. 10.

Staab, S. (2006, 9). Managing Knowledge in a World of Networks. (P. C.-1. International Conference, Ed.) *15th International Conference, EKAW 2006, Podebrady, Czech Republic, October 6-10, 2006, Proceedings*.

Strydom, J. (2005, 9). *Introduction to Marketing.* Juta and Company Ltd.

Stuckenschmidt, H., & Harmelen, F. v. (2006, 3). *Information Sharing on the Semantic Web.* (Springer, Ed.) Springer Science & Business Media.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. *Semantic Web*, 10.

Supnithi, T., Yamaguchi, T., Pan, J. Z., Wuwongse, V., & Buranarach, M. (Eds.). (2015). *Semantic Technology* (Vol. 8943). Cham: Springer International Publishing. doi:10.1007/978-3-319-15615-6

Svátek, V., Sváb-Zamazal, O., & Vacura, M. (2010). Adapting Ontologies to Content Patterns using Transformation Patterns. *WOP.*

Tello-Leal, E., Rios-Alvarado, A. B., & Diaz-Manriquez, A. (2015, 9). A Semantic Knowledge Management System for Government Repositories. *2015 26th International Workshop on Database and Expert Systems Applications (DEXA)*, (pp. 168-172). doi:10.1109/DEXA.2015.48

Telnarova, Z. (2010, 10). Relational database as a source of ontology creation. *Proceedings of the International Multiconference on Computer Science and Information Technology* (pp. 135-139). IEEE. doi:10.1109/IMCSIT.2010.5679648

Terzieva, M. (2014). Project Knowledge Management: How Organizations Learn from Experience. *Procedia Technology, 16*, 1086-1095. doi:10.1016/j.protcy.2014.10.123

Thiéblin, É., Haemmerlé, O., Hernandez, N., & Trojahn, C. (2018). Task-Oriented Complex Ontology Alignment: Two Alignment Evaluation Sets. In A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, . . . M. Alam (Ed.), *The Semantic Web* (pp. 655-670). Springer International Publishing.

Thuy, P. T., Lee, Y.-K., & Lee, S. (2009). DTD2OWL: Automatic Transforming XML Documents into OWL Ontology. *Proceedings of the 2Nd International Conference on Interaction Sciences: Information Technology, Culture and Human* (pp. 125-131). New York, NY, USA: ACM. doi:10.1145/1655925.1655949

Tran, T., Cimiano, P., Rudolph, S., & Studer, R. (2007). Ontology-Based Interpretation of Keywords for Semantic Search. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, . . . P. Cudré-Mauroux (Eds.), *The Semantic Web* (Vol. 4825, pp. 523-536). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-76298-0_38

Trinkunas, J., & Vasilecas, O. (2007). International Conference on Computer Systems and Technologies-CompSysTech'07 Building Ontologies from Relational Databases Using Reverse Engineering Methods. *International Conference on Computer Systems and Technologies .*

Tsoukas, H. (2005). *Complex Knowledge: Studies in Organizational Epistemology.* (O. U. Press, Ed.) Oxford University Press.

Tsoukas, H., & Vladimirou, E. (2001, 11). What is Organizational Knowledge? *Journal of Management Studies, 38*, 973-993. doi:10.1111/1467-6486.00268

Tzitzikas, Y., Spyratos, N., & Constantopoulos, P. (2005, 3). Mediators over taxonomy-based information sources. *The VLDB Journal, 14*, 112-136. doi:10.1007/s00778-003-0119-8

Uschold, M. (2003). Where Are the Semantics in the Semantic Web? *Awe Magazine, 24*, 25-36.

Usman, Z., Young, R. I., Chungoora, N., Palmer, C., Case, K., & Harding, J. A. (2013, 11). Towards a formal manufacturing reference ontology. *International Journal of Production Research, 51*, 6553-6572. doi:10.1080/00207543.2013.801570

Vaishnavi, V. K. (2007, 10). *Design Science Research Methods and Patterns: Innovating Information and Communication Technology.* CRC Press.

Villazón-Terrazas, B. M. (2012, 4). *A Method for Reusing and Re-engineering Non-ontological Resources for Building Ontologies.* IOS Press.

Wali, B., & Gibaud, B. (2012, 10). Extending OWL-S for the Composition of Web Services Generated With a Legacy Application Wrapper. *arXiv:1210.1179 [cs]*. Retrieved from http://arxiv.org/abs/1210.1179

Wang, H., Zhang, K., Liu, Q., Tran, T., & Yu, Y. (2008). Q2Semantic: A Lightweight Keyword Interface to Semantic Search. In S. Bechhofer, M. Hauswirth, J. Hoffmann, & M. Koubarakis (Eds.), *The Semantic Web: Research and Applications* (Vol. 5021, pp. 584-598). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-68234-9_43

Wang, Y., Wan, C., Deng, L., Liu, X., Cao, X., Gilbert, S. R., . . . Clemens, T. (2007, 6). The hypoxia-inducible factor α pathway couples angiogenesis to osteogenesis during skeletal development. *Journal of Clinical Investigation, 117*, 1616-1626. doi:10.1172/JCI31581

Weigand, H. (2016, 2). The e3value Ontology for Value Networks: Current State and Future Directions. *Journal of Information Systems, 30*, 113-133. doi:10.2308/isys-51409

Wiig, K. M., Hoog, R., & Spek, R. (1997, 7). Supporting knowledge management: A selection of methods and techniques. *Expert Systems with Applications, 13*, 15-27. doi:10.1016/S0957-4174(97)00019-5

Xu, Z., Cao, X., Dong, Y., & Su, W. (2004, 5). Formal Approach and Automated Tool for Translating ER Schemata into OWL Ontologies. *Advances in Knowledge Discovery and Data Mining* (pp. 464-475). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-24775-3_57

Yahia, N., Mokhtar, S. A., & Ahmed, A. (2012, 6). Automatic Generation of OWL Ontology from XML Data Source. *arXiv:1206.0570 [cs]*. Retrieved from http://arxiv.org/abs/1206.0570

Zarri, G. P. (2015, 4). Semantic/Conceptual Annotation Techniques Making Use of the Narrative Knowledge Representation Language (NKRL). *The Twenty-Eighth International Flairs Conference.* Retrieved from https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS15/paper/view/10350

Zhang, X., Di, R., & Feng, X. (2012, 9). Ontology Based Data Conversion from Spreadsheet to OWL. *2012 Seventh ChinaGrid Annual Conference*, (pp. 76-79). doi:10.1109/ChinaGrid.2012.17

Zhao, S., & Qian, Q. (2017, 10). Ontology based heterogeneous materials database integration and semantic query. *AIP Advances, 7*, 105325. doi:10.1063/1.4999209

Zhong, J., Zhu, H., Li, J., & Yu, Y. (2002). Conceptual Graph Matching for Semantic Search. *Proceedings of the 10th International Conference on Conceptual Structures: Integration and Interfaces* (pp. 92-196). London: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=645495.657719

Zhou, Q., Wang, C., Xiong, M., Wang, H., & Yu, Y. (2007). SPARK: Adapting Keyword Query to Semantic Search. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, . . . P. Cudré-Mauroux (Eds.), *The Semantic Web* (Vol. 4825, pp. 694-707). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-76298-0_50