FACULTY OF ENGINNERING OF UNIVERSITY OF PORTO

PORTUGAL

INFORMATION MANAGEMENT IN WORK ORGANIZATION DOMAIN IN NETWORK ORGANIZATIONS

By
Cristóvão Dinis Polido Sousa
Graduated in Informatics Engineering

Dissertation submitted for partial satisfaction
of master degree requirements in
Electrical and Computers Engineering

Dissertation developed under the supervision of
Professor Dr. António Lucas Soares,
Departamento de Engenharia Electrotécnica e Computadores da Universidade do Porto

Porto, December 2007

Abstract

The main goal was to develop an ontology and its practical application scenario in work organization domain, supported by a closer interaction between work design processes which calls for multi-disciplinary developing teams sharing knowledge and competencies.

For achieving this purpose it was used a new category of tools and methodologies supported by new knowledge management technologies. Consequently, and after an analysis of the different knowledge representation mechanisms, it was adopted a graphical-based knowledge representation formalism using concept maps for the domain conceptualization in a distributed environment, provided by CmapTool software from IHMC – Institute for Human and Machine Cognition. The choice of a less formal knowledge representation emerged with the purpose of enabling, in an initial phase, the domain experts to express and share, in an intuitive way, their knowledge about the work organization domain.

Subsequently, the concept maps were converted to their OWL specification using CmapTools COE software.

The final part of the work addressed the practical application of the ontology. Therefore, it was found a technological architecture allowing "transferring" the developed ontology to the Plone Content Management System. Trough that architecture it was generated a set of Plone objects according to a specific domain of interest specified by the ontology, in order to become easier the organizational content management.

Keywords: Information Management; Ontology; Concept Maps; Knowledge Representation; Work Organization Design; Job Design; Collaboration.

Resumo

O presente trabalho apresenta como objectivo fucral o desenvolvimento e aplicação prática de uma ontologia de alto nível no domínio da organização do trabalho, apoiada numa abordagem aos processos de redesenho do trabalho o que exige equipas multidisciplinares de desenvolvimento partilhado conhecimento e competências.

Para a concretização deste propósito foi utilizada uma nova categoria de ferramentas e metodologias apoiadas por novas tecnologias de gestão de conhecimento. Consequentemente, e após uma análise aos diferentes mecanismos de representação de conhecimento, adopou-se uma abordagem de representação gráfica baseada em mapas de conceitos para a conceptualização do domínio, num ambiente distribuido e colaborativo proporcionado pela ferramenta CmapTool do IHMC – Institute for Human and Machine Cognition. Esta opção de representação menos formal do conhecimento surge com o propósito de habilitar, numa fase inicial, os especialistas do domínio de expressarem e partilharem de forma simples e intuitiva o seu conhecimento acerca do domínio da organização do trabalho.

Posteriormente, os mapas de conceitos foram actualizados através da sua especificação em OWL, utilizando a ferramenta CmapTool COE.

Com a ontologia desenvolvida e estável, foi possível colmatar o último objectivo, relaccionado com a aplicação prática da ontologia. Por conseguinte, foi encontrada uma arquitrectura tecnologica que permitiu "transferir" a ontologia desenvolvida para o sistema de gestão de conteúdos Plone. Através dessa arqwuitectura foi gerando um conjunto de objectos do Plone, personalizados de acordo com o domínio de interesse especificados pela ontologia, com o intuito de facilitar a gestão do conteúdo organizacional e sem a necessidade de desenvolvimentos adicionais de software.

Palavras-Chave: Gestão da Informação; Ontologia; Mapas de Conceitos; Representação de Conhecimento; Organização do Trabalho; Desenho do Trabalho; Colaboração.

Acknowledgments

Close to the end of this work, come into my mind all this long journey of anxiety and hard labour. During this period several were the feelings that get hold of my soul; some ones were nice, other ones were less enthusiastic.

I am able to say now that I am really glad to see this work complete, and I am happy no matter what, because I learned so many things beyond what is described in this thesis.

Thus, it is time to mention a special word of gratitude to those that were always present during the nice and the bad moments of this thesis development.

In first place I want to give special thanks to my supervisor, Professor Dr. António Lucas Soares, for the technical, scientific and psychological support during the work progress. His personal advices and his professional orientation was the driving engine of this travel. Once again thank you very much.

To the president of the Association for Integrated Development of Ermesinde City (ADICE), Dr. Maria da Trindade Vale, I would like to express my pleasure for her understanding and facilitation, which became crucial for me to combine my professional activity with the research activity inherent to this work.

I want to thanks to Professor Dr. José Oliveira Restivo for his understanding about bureaucratic issues related to the elapse of my thesis work.

But for all this to be possible it is necessary a solid social background, which in my case was made up of my family. They were really super in all the support, enthusiasm and motivation transmitted to me. To my father Joaquim, my mother Maria de Fátima and my lovely brothers, António, Manuel and Esmael thanks very much for being always there all the time.

At last but not the least, I would like to return the tenderness, patience and the stimulation that I have been receiving from my girlfriend Liliana. Thank you Liliana, for your concern dedication and motivation.

I reserved an exceptional gratitude to my uncle, Carlos António Lebreiro Polido. Nothing would be possible if you didn’t intercede for me.

Thank you, God.

Index

List of Figures

List of Tables

Chapter 1

Introduction

It's a fact that enterprise competitiveness depends on its capability of creating value in a global and instable environment in constant change. However this level of economic competitiveness has created an unprecedented set of new opportunities and at the same time uncertainties [27].

Flexibility, efficiency, innovation in enterprises are key factors for productivity improvement achieved trough new effective business process management and new forms of work organization in a tri-dimensional plan where it's fundamental the understanding, leading and managing of the markets, human resources and technology.

Change is not reached only with the implementation of new electronic work tools or with the suppression or substitution of organizational resources or with the increase of new channels for doing business. Organizational performance depends on process efficacy and quality of life at work.

According to the green book "Partnership for new organization of work" from the European commission, the improvement of competitiveness and employment is achieved trough a better organization of work, individuals and groups that intervene in the organizational processes based in an effective management of high skills, high trust and high quality. The implementation of new forms of work organization which are more flexible is a complex development process that implies an interaction between work design, individuals and groups.

Technical and organizational innovation processes e.g., business process reengineering and work redesign are normally loosely coupled leading to a sub-optimal socio-technical system. Closer interaction between these design processes is needed which calls for multi-disciplinary development teams sharing knowledge and competencies. Multidisciplinary cooperation is a new challenge and a new category of methodological tools supported by modern knowledge management technologies is needed in order to make feasible such complex development processes. However in a knowledge management initiative technology alone is not sufficient: knowledge management is a socio-technical discipline.

1.1 Objectives

According to the previous contextualization done either internally and externally at the level of an organization, the main objectives of this work are:

1. Characterization of work within and between organizations regarding the organizational concepts, types of relationships and the new trends of work organization.
2. Structuring and modelling the domain of organization of work using ontologies' engineering. An ontology can present in a simple way the action fields in the work organization domain and group the main formal concepts and their correlation without the constraints of pre-defined reorganization models.
3. With the purpose of achieving the previous goal, an approach to collaborative work design of individuals and groups will be adopted. In this context is identified the following sub-objectives:
    a. Find a way to formalize and represent knowledge in a distributed environment.
    b. Find a technological architecture that allows sharing knowledge collaboratively in a multidisciplinary context, in order to find a model of organization of work that can be maintained in a practical and useful way, without any special technological skills needed. The tools that support this architecture rely on situational agreements regarding the terminology and semantics of the involved scientific and practice fields. These agreements need to be formalized in "ontologies" that serve both the purposes of human and machine understanding the domain of discourse.
4. Implementation of a “prototype” for information management of work organization using a content management system.

This work tries to demonstrate how useful the ontology engineering is in the process of information management and knowledge sharing in order to perform collaborative work among units within or between organizations.

1.2 Thesis Structure

This dissertation work is organized in 7 chapters as shown in the next figure.

Figure 1.1 – Thesis Structure

This first chapter is initiated with a brief introduction to the developed work, contextualizing it and presenting its general goals. The structure of the thesis is presented in a map (previous figure) where can be seen the main topics discussed in the current document.

Chapter 2 surveys the representation of conceptual systems. In this context we'll find a short description of the main knowledge representation (KR) formalisms and later a deeper analysis in collaborative and graphical knowledge representation field. Still in this chapter, some graphical collaborative tools and visualization tools will be briefly presented. Along this chapter an ontological approach for KR is also discussed. Ontologies play an important role in this context, providing semantics to the KR formalisms in order to obtain a conceptual system capable of being encoded and distributed among different computational systems and at the same time understandable for human being.

Chapter 3 begins with a more specific approach to the knowledge representation through the creation of ontologies with conceptual maps. Regarding ontologies, a synthesis of the most significant development methodologies was performed. However and despite of all interesting methodologies available for building an ontology, it none was used in as our main goal was to develop a tool for specifying an ontology that could hopefully be used in any development methodology.

The activities and practical experiences that were developed in the CODEwork@VO project (**CO**llaboration and **DE**mocratic **work** design in **V**irtual **O**rganizations) are here included as a demonstration of Concept Maps usage in the process of ontology construction.

Chapter 4 presents the work organization ontology developed with concept maps in the CmapTools COE authoring environment. A succinct explanation of each map is given when all the concept's definitions are catalogued in annex A.

Chapter 5 discloses a practical and possible application scenario of the developed ontology. All the steps needed to build a hierarchically set of folders and files into a **C**ontent **M**anagement **S**ystem (CMS) according to the ontology exported in OWL, are described in detail. This process has several stages and constraints, which are mentioned during this chapter. The process is, in fact, very interesting but has too many manual tasks yet. In this particular case I foresee some future work to be done.

Chapter 6 concludes the thesis and presents some directions for further work along the lines proposed in this one.

Chapter 2

Representation of Conceptual Systems

The world is a huge system of interdependent objects and events. The observation and experimentation of things delineate a mental map of ideas and concepts in a relational scheme which allows us to understand and interact with the environment. The sets of organized definitions, names, symbols and other instruments of thought and communication are called Conceptual System (CS).

The representation of those schemes in a structured language of interrelated concepts aims to symbolize the contents of thinking and it's fundamental for the understanding and communication between people about a specific domain. A set o musical notes, for example, is a CS which has a specific representation, in a way that every musician can interpret a certain sequence of musical notes. In the organizational domain we may have exactly the same approach in a way that all structure of an organization can be explained trough the design of its CS. However, nowadays organizations are in constant mutation due the global environment and market pressures. Companies, work more often spread geographically forming a network in a complex value chain where its elements may change regularly. In this context, new forms for representing CS supported by ICT in a collaborative environment are needed. This technological approach drives to the concept of knowledge representation [86].

According to Patrick Hayes [82], *"Knowledge Representation (KR) refers to general topic of how information appropriately encoded and utilized in computational models of cognition"*. KR has two main work areas which are Knowledge Representation Formalism and Knowledge Engineering which refers to the development and maintenance of Knowledge Based Systems.

The capability of knowledge representation and knowledge sharing is the main challenge in all knowledge management process which involves knowledge acquisition, knowledge representation, knowledge maintenance and knowledge use.

2.1 Knowledge representation Formalisms

There are several ways of representing knowledge and consequently we may have several representation formats in knowledge based systems depending on what formalisms were chosen by knowledge system developers. Hence, sharing and reusing knowledge expressed in different representations becomes more important as the use of knowledge based systems continues to expand [1]. Therefore, the secret is in how the knowledge is represented, which demands for new knowledge representation tools capable of easily share knowledge either between different knowledge based systems and people. From an organizational perspective is very important if organizational knowledge can be effectively used in automated systems and also understood and updated by humans [21].

There are several formalisms in order to develop representations about a conceptualization of a given reality. Such representation formalism, known as Knowledge Representation Formalisms (KRF), are divided in five main levels [24], namely *Implementation Level, Logical Level, Epistemological Level* , *Ontological Level* and *Conceptual Level.*

At the implementation level the main issue is how to build a computer program under some KR formalism. Concerning to the logical level, the major goal is on logical properties of KR formalism, mainly the meaning of expressions and if expressions are well formed. This is the level of formalization. Epistemological level concerns are about the knowledge structure primitives that are needed and the types of inference types. At this level the main issue is about the nature and sources of knowledge. The fourth level goal is characterizing the meaning of the basic ontological categories used to describe the domain: the ontological is therefore the level of meaning. Conceptual level is the top level of KR formalisms and it main concern is with the actual primitives that should be included in KR formalisms defining cognitive interpretation to language-independent concepts. [23], [24].

According with Guarino [24], KR formalisms can be classified according to the kinds of primitives.

Table 2-1 KR Formalism Classification

Level	Primitives	Interpretation	Main feature
Logical	Predicates, function	Arbitrary	Formalization
Epistemological	Structuring relations	Arbitrary	Structure
Ontological	Ontological relations	Constrained	Meaning
Conceptual	Conceptual relations	Subjective	Conceptualization
Linguistic	Linguistic terms	Subjective	Language Dependency

From an organizational management perspective two dimensions for knowledge creation can be distinguished [87], [88]: **Ontological dimension** in which an organization cannot create knowledge without individuals. The organization supports creative individuals or provides contexts for them to create knowledge in an interaction Individual ↔ Group ↔ Organization ↔ Inter Organization; **Epistemological dimension** which comprises tactic and explicit knowledge.

Tacit knowledge is not easy to visualize or express. It's highly personal and hard to formalize. Explicit knowledge (or in other words, information) can be expressed in words and numbers and can easily be communicated and shared. Tacit knowledge is created and shared through social interaction and explicit knowledge interacts and interchanges into each other in the creative activities of human beings [87].

John Sowa [2] asserts that the task of knowledge representation has two parts: a) to analyze some body of knowledge and identify the relevant concepts, relations and assumptions; b) to translate the result of the analysis into some notation that can be processed by computer. The first part is the focal point of this second chapter.

In a generic approach there are two main groups of knowledge representation design and reasoning, namely: Semantic networks which provide a visual representation for knowledge and Logic which provide family of formalisms that can be used to represent formal taxonomies of a specific domain. According to these two main groups, Frame-based formalisms, logic-based formalisms and graphical formalisms will be reviewed in the next sections.

2.2 Logic-based Knowledge Representation Formalisms

Logic representation schemes provide a set o textual formal notations to represent facts as logical formulas which represent the description of a specific domain [73]. It is the most used form of knowledge representation at a computational level because it has a good syntax for describing concepts and rules of inference for information retrieval. Logic-based formalisms focus in statements, their content (semantics), their form (syntax) and their relation. According to [73] their main advantages are:

- Availability of inference rules.
- Clean, well-understood and well-accepted formal semantics
- Simplicity of the notation employed which leads to knowledge base descriptions that are understandable.

As disadvantages, the same author mentioned:

- Lack of organizational principles
- Difficulty in representing procedural and heuristic knowledge.

There exist several standards for logic such as: propositional logic (PL); first-order logic (FOL); second-order logic; modal logic; temporal logic. Among all this languages, FOL can be viewed as a "common language" to talk about Knowledge Management formalisms because many of the existing KR formalisms can be understood by translating them into FOL or they are a variant of FOL or else they are closely related with FOL. For that reason, in the next section it will be briefly described FOL as the main logic knowledge representation formalism.

2.2.1 First Order Logic (FOL)

FOL actually extends propositional logic and it is composed of statements (propositions) that are assumed to be true. The syntax of those statements is composed of:

- Objects (e.g. *Portugal, JobDesign, Automobile*)
- Relations between objects (*connectivities - see table below*)
- Properties which distinguish objects (e.g. *little, fast, red, big*)
- Functions (e.g. *isPartOf*)

Table 2-2 FOL connectivities

$\wedge$	and	conjunction	&
$\vee$	or	disjunction	|
$\neg$	not	negation	~
$\Rightarrow$	if … then	implies	$\rightarrow$
$\Leftrightarrow$	if and only if	iff	$\leftrightarrow$

Indeed, FOL is well suited to the idea of Knowledge capture in a domain of discourse; it allows quantification over individuals of a given domain of discourse, it means that, for example, it can be stated in FOL that "**every** individual has the property **P**". Next table shows the "qualifies" that can be used in FOL.

Table 2-3 FOL Qualifiers

$\forall$	For all	indicate that the sentence should hold when anything is substituted for the variable
$\exists$	There exists	Indicate that there is something that can be substituted for the variable such that the sentence holds (but doesn't necessarily tell us what it is).

According to the explanations presented knowledge can be represented in FOL as shown in the next samples:

a) 5 is greater than 3.

- **greater(5,3)**

b) Grass color is green

- **color(Grass, Green)**

c) Cristóvão is not tall

- **~tall(Cristóvão)**

d) There are exactly two work design scenarios

- **(Ex)(Ey) DesignScenario(x) ^ Work (x) ^ DesignScenario (y) ^ Work(y) ^ ~(x=y) ^ (Az) (DesignScenario (z) ^ Work (z)) => ((x=z) v (y=z))**

e) Job Design is part of Job analysis

- **JobDesign (is_part_of(JobAnalysis))**

2.3 Frame-Based Representation Formalisms

Frames representation scheme became popular in the 70s when Minsky introduced a completely different representation of knowledge for that time. In general, "frames are structures representing classes of objects in terms of properties that their instances must satisfy. Such properties are defined by the frame slots, which constitute the items of a frame definition. [76]".

Frames are something close to the database records but with more capacity to express data. It is a complex structure composed by:

- An identification, which is basically a name.
- Slots, which correspond to the object attributes. Each slot has its identification; its default value and its current value.

We can say that frames are "object-oriented" representation formalisms and each frame has at least another frame hierarchical superior. Hence, frames allow the implementation of the inheritance mechanism, and thus, frames can establish a set of relations between each other according to following list:

- "is-a" relation, which is a relation of hierarchical dependency.
- Inherit relation, which is a relation that allows inheritance of frame slots that are not in the same hierarchical line.
- Do not inherit, where there are no inheritance of frame slots in the same hierarchical line
- A *slot*, which can be also a *frame*

Frames had a huge impact as KR formalisms and quite a few languages and tools implementing frames emerged subsequently. Other ones, like Ontology Web Language (OWL), emerged from the new approach to Frames based ontology languages. Regarding frame-based implementation tools, Protégé, which is the most used software tool for knowledge acquisition and representations, uses frames in its core specification.

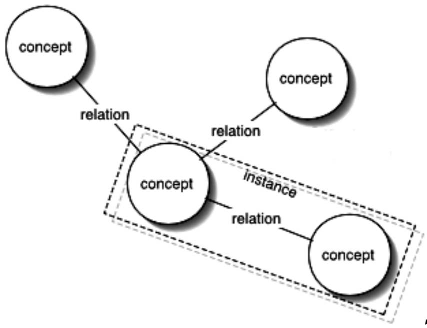
2.4 Graphical-based Knowledge Representation Formalisms

2.4.1 Semantic Networks

Semantic Networks were developed by M.R.Quilian in 1968 for representation of the contents of dictionaries for artificial intelligence and machine translation [22]. It's an old way of knowledge representation formalism where the concepts and its relations of the problem domain are expressed in a declarative graphical notation. There are 3 basic elements in semantic networks:

- ***Concepts***
- ***Relationships***
- ***Instances***

Figure 2-2 Elements of Semantic Networks



Source: Semantic Research Inc.

In a semantic network, each node represents a concept, object or situation connected by labeled arcs or edges which represent a semantic relation between concepts. In such representation, meaning is implied by the way a concept is connected to other concepts [18].

The most important semantic relations in a semantic network are:

- *A is part of B*
- *B has A as a part of itself*
- *A is a kind of B; A is a subordinated of B*
- *A is superordinated of B*
- *A denotes de same as B*

Therefore, semantic network may use special labeled arcs giving a partial order on concepts. Such arcs can be from different types:

- Arcs that assign properties to concepts or objects.
- *Is-a* arcs that introduce hierarchical relationships. *Is-a* arcs indicate that a concept is a subclass of another
- *Instance-of* arcs indicates that a concept is an example of another

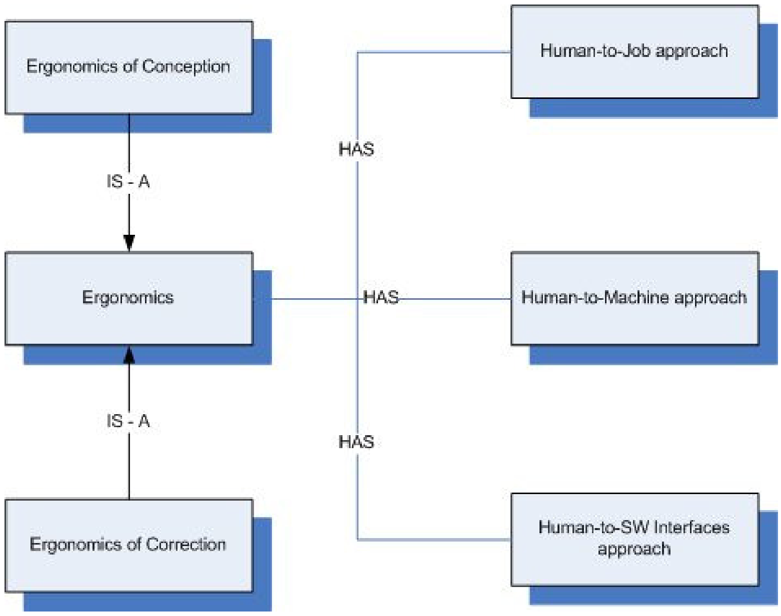
Figure 2.2 illustrates a simple example of a semantic network. In this simple network, nodes are objects and the links (arcs) represent i*s-a* and *has* the relationships between items.

As shown in the figure a car *is-a* vehicle; a vehicle *has* wheels. This kind of *is-a* relationship represent:

- ***Hierarchical*** organization of concepts, e.g. “Ergonomics of Conception” is a sub-object or sub-concept of Ergonomics.
- ***Instantiation*** of concepts, e.g. “Ergonomics of Correction” is an instance of “Ergonomics”.

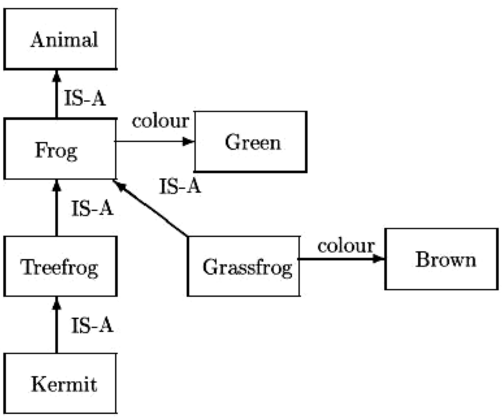
- ***Inheritance*** hierarchy in the network, with objects lower down in the network inheriting properties from objects higher up. Ergonomics has "Human-to-Job" approach, so every "Ergonomics of Correction" has the same approach.

Figure 2-3 Semantic Network example



According to [21], semantic networks are a powerful knowledge representation system because they are easy to understand by humans and can be used in automated processing systems which means that they can also became a vehicle to archive organization knowledge. However, due to the lack of formal semantics, the meaning of a given network is not clear [22]. The meaning of the nodes, property edges and "*is-a"* edges, may not be very clear in a semantic network.

Figure 2-4 Semantic network example with conflict



Source: (Baader, 1999)

Figure 2.4 illustrate an example of a semantic network with conflict.

As for the meaning of nodes, there is no clear distinction between concepts (classes of objects) and objects [22]. *Is-a* edges describe concept hierarchy and instantiation, however grass frogs do not inherit the property green, since this would be in contradiction with the explicit property edge saying that grass frogs are brown [22]. In the same way, the property edge "colour" form "Frog" to "Green" means [22]:

- that a frog has only the color green *(value restrictions)*
- a frog has a least the color green, but may have other colors *(existential restrictions)*
- in general a frog has the color green, but there may be exceptions *(default property)*

Due its informal characteristic, there are many variances of semantic networks, some versions are highly informal, but other versions are formally defined systems of logic and can be categorized into six categories [2]:

- *Definition Networks*
- *Assertion Networks*
- *Implication Networks*
- *Executable Networks*
- *Learning Networks*
- *Hybrid Networks*

All these categories inherit from declarative graphic notation to be used to support knowledge representation or supported automated systems reasoning knowledge. Each semantic network category uses its mechanism to represent knowledge in a specific problem domain which allows choosing the semantic network formalism that best serves the representation of our problem domain.

2.4.2 Conceptual Graphs

Conceptual Graphs (CGs) is a formal logic-based knowledge representation developed by John Sowa, based on the existential graphs work of Charles Pierce and the semantic networks. Conceptual Graphs are particularly suitable for formal representation of meaning. They can be used in making explicit the semantics of concepts in an expressive way of knowledge representation, offering a mathematic formalism. CGs are easy to use and most of all, they are human readable.

In the basis of Conceptual Graphs model they are constituted by:

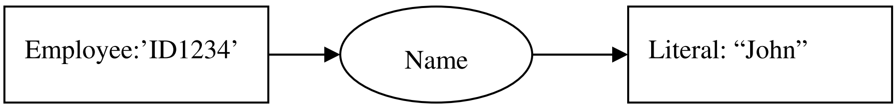
- Two kinds of nodes:
    - Concepts – represented by boxes or [square brackets].
    - Conceptual Relations – represented by circles or (parentheses).
- Arcs between concepts.

CGs formalism can be represented in several notations: *Display Form* (DF) which represents a CG as a truly graph in a bipartite model. This means that there are no arcs between a concept and another concept, and no arcs between a relation and another relation.

A second notation is called *Linear Form* (LF) and represents a CG in a compact from more human-readable.

Conceptual Graph Interchange Form (CGIF) is another CG notation to make CGs machine-process-able. CGIF is employed as the representation for encoding knowledge represented by bipartite graphs into machine-readable character strings [1].

Figure 2-5 Conceptual Graph represented in Display Form



Source: (Yao, 2006)

The figure above represents a simple CG in DF form. The same example in LF looks like this:

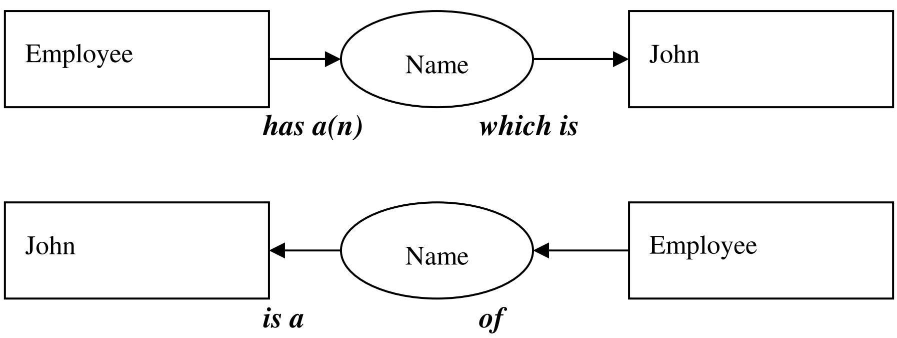
- [Employee:'ID1234'] → (Name) → [Literal:"John"]

In CGIF form:

- (Name [Employee:'ID1234'][Literal:"John"])

Before go further its important to well read a Conceptual Graph. There are two gold rules that we must not forget for an adequate reading of a CG. Let's look at the next example.

Figure 2-6 Reading a conceptual graph



As figure 2.6 illustrates, CGs can be draw in two directions which may represent a different denotation. The words under the arrows are only mnemonics for help reading the graph. Thus the two CGs should be read:

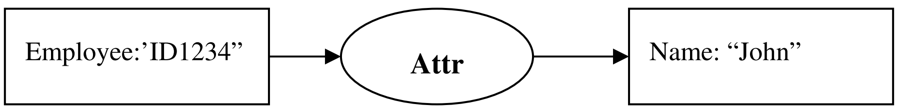
Employee has a Name which is John.

And,

John is a Name of Employee.

However, the Display Form of representing conceptual graphs has two ways for representing conceptual relations. The first part of the last figure represents one way of Display Form for CGs while the second part represents another view for a CG with a different choice for conceptual relations.

Figure 2-7 CG with predefined relation type



The Conceptual Graph from figure 2.7 can be read as the following:

"ID1234 Employee has an attribute which is the Name"

"ID1234 Employee is John"

Every relation has a relation type allowing us to better perceive the kind of relation between concepts. Relation types are simple names that are given to the relation. [8].

In the Online Conceptual Graphs course [26] at Aalborg University - Department of Communication (http://www.huminf.aau.dk/cg), we can find an overview of the relation types and its meaning. Next are presented a few examples:

On		(on)
In		(in)
Dest		(destination)

Agnt		(agent)
Thme		(theme)
Ptnt		(patient)
Rcpt		(recipient)

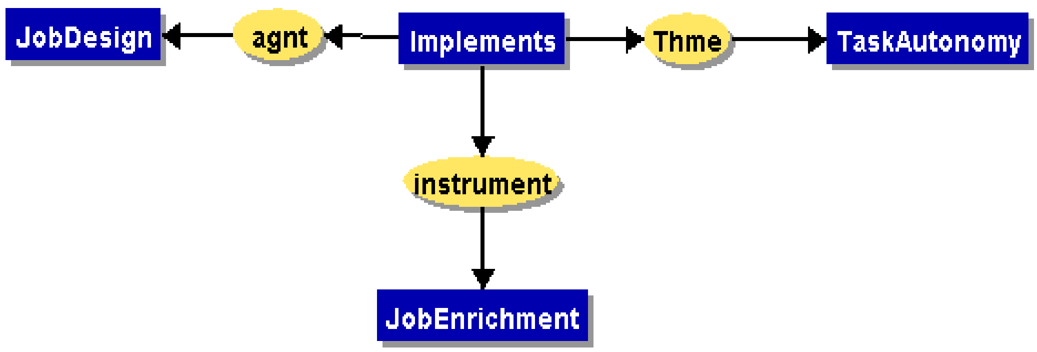
A very important relation is "*Agnt*" or "agent". It relates an act and an animated being which perform the act [17], such as the relation "*Attr*" or "attribute" in figure 2.7 relates an object to an entity where an entity is a property of some object.

Representing relations with formal relation types are convenient for translations to and from natural language [17].

Concepts can also be created with one or two entities: the *concept type* and its *referent*. The concept type is a name which defines a certain group of entities in a categorized way e.g., Person, Employee, etc. The referent of a concept is an individual from the group specified by the concept type. Figure 2.7 hows two concepts, each one with a concept type (Employee and Name) and a referent (ID1234 and John) respectively.

Let's analyze the next CG.

Figure 2-8 CG Display Form Example



The previous figure represents an example of the work organization domain where we can read the following:

- "Job Design", is the agent of "Implements", and the theme of "Implements" is "Task Autonomy", and the "instrument" of "implements" is "Job Enrichment".

This means:

- Job Design implements Task Autonomy through Job Enrichment.

CGs are a very powerful and versatile tool for knowledge representation, human readable and machine process able. Nevertheless it is not possible to draw a CG without having a basic knowledge of logic and CGs itself.

2.4.3 RDF

Resource Description Framework (RDF) is a frame work for representing information in the web. RDF has the capability to formal express the data meaning allowing interoperability and provides an integration environment between different patterns of metadata[1].

RDF emerged aiming to represent information about the web resources. RDF language was created to represent a simple data model based on XML and using vocabulary based on Uniform Resource Locator (URL). The data model is graphically represented through **triples** which consist in 3 types of objects that describe relationships between resources regarding properties and values.

Figure 2-9 RDF Triple

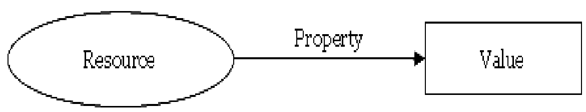
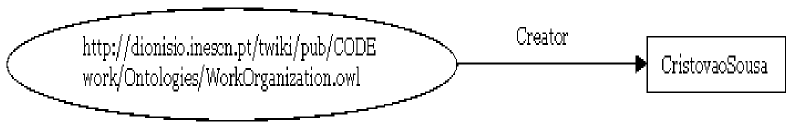


Figure 2-10 RDF Triple document representation



The RDF implementation of the previous triple:

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

<rdf xmlns=http://www.w3.org/1999/02/22-rdf-sysntax-ns#

Xmlns:s=http://description/org/schema/>

<Description
about=http://dionisio.inescn.pt/wiki/pub/CODEwork/Ontologies/WorkOrganization.owl>

<s:Creator>CristovaoSousa</s:Creator>

</Description>

</rdf>
```

[1] http://www.w3c.org/TR/rdf-concepts

Each RDF triple is composed by:

- **Resource**: Anything that can contain a Uniform Resource Identifier (URI), including web pages, as well as elements of a XML document. A resource can be also something that is not directly accessible in the web.
- **Property**: A resource that has a name and can be used as a property
- **Statement**: Is the combination of a resource, a property, and a value.

The elements of a triple are represented by nodes and arcs, where:

- A node (represented by an ellipse) identifies a resource;
- A node (represented by a rectangle) contain the value;
- Arc (represented by an arrow) identifies the property. The arrow direction is important.
- A node (represented by an empty ellipse) identifies an anonymous resource;

RDF expresses the meaning of data allowing interoperability in the web. It is therefore a knowledge representation formalism which provides the structure that is used to represent data models for objects and their relations. However it needs RDF Schema in order to provide mechanisms to declare properties and define relations between properties and resources. RDF Schema is used to describe, semantically, properties, classes of web resources and the type of data for the property values. It extends RDF with new vocabulary allowing the knowledge to be represented through ontologies.

2.4.4 Topic Map

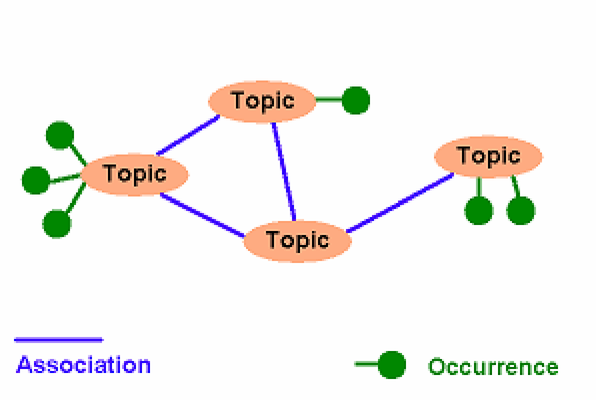
Topic Maps (TM) is a way to represent knowledge that came up close together to Cmaps but within information science area.

Topic Maps is a graphical KR formalism regarding the data structure of information resources, standardized by the International Standards Organization in 2000 (ISO/IEG 13250) as XML Topic Maps (XTM). The fact that TM is associated to an international standard is a relevant aspect to make use of it.

By analogy, TM concept is similar book indexes. Like book indexes, typically, point to all topics comprised in the work and all related sections and as well as how we may find it, also TM organize subjects by topics allowing a broad vision about the collection of resources an how they are connected between them. However, a book has only topics about itself; TM goes further and represents knowledge from several subjects.

Technically a TM is formed by three concepts: *Topic name*, *Association* and *occurrence*.

Figure 2-11 Topic Map elements



Source: Wikipedia.org

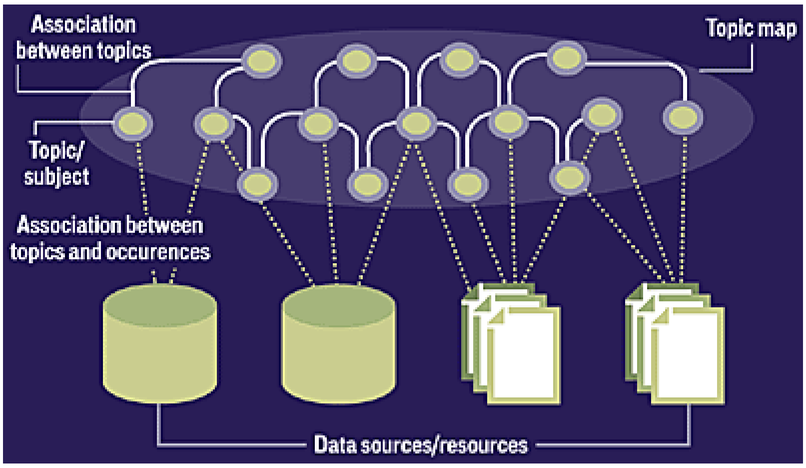
The topic can be anything – an object, a person, a concept, etc. – and usually has a name though is not always necessary but it's advisable and must be unique. The association indicates how a topic is related with other topics. In practical terms, an association is the link element between to or more topics. Each topic involved in an association, is said to play a rule [20]. For example: From a Organizational point of view, "Work organization defines work practices". In this case there are two topics – *WorkOrganization* and WorkPractice and there is an association between them, where *WorkOrganization* plays the rule of the one that definesand the *WorkSysten* the defined.

Occurrences are pointers for relevant information resources such as:

- Articles
- Books
- Images
- Videos
- Source code
- People, etc.

For a best understanding of the TM concepts, figure 2.12 shows the representation of knowledge structures in Topic Maps and their association with information sources.

Figure 2-12 Topic Map Information Navigation



Source: (Russell, 2004)

This last feature is similar to CMs and the resources are represented by a URI (Uniform Resource Identifier), e.g., a URL is a kind of URI. Besides, all the good practices described in the last section for Cmaps are valid for TM. Defining a scope, identifying a specific subject for a TM, collecting the topics and resources and then build the TM is the common thing to do.

Between all KR formalisms discussed, TM is the most recent one and the only one that is completely standardized as XTM (XML Topic Maps), which make TM very well accepted among the information science community.

TM becomes very useful essentially because of its ability for information navigation - an effect from the book-index approach on its development. However and despite TMs formal syntax being compliant with the XTM standard, there is no formal semantics associated with it (nor with XTM syntax). As well as for Cmaps, ontologies seem to be the solution for this problem, either for TM or Cmaps.

Topic Maps are also well suited to represent ontologies. Because of the key role of ontologies in many real-world applications, the ability of Topic Maps to link resources anywhere in the semantic web, and then organize these resources according to a single ontology, will make Topic Maps a key component of the new-generation of Web-aware knowledge management solutions [84].

Concept Maps, described next, is a graphical and flexible notation for knowledge representation. They are informal and useful as they are difficult to convert to a formal specification; however its broad applicability overcomes that constriction. "*With appropriate constraints and conventions, CMs have been used as a highly readable front end to formalize notations such as RDF and OWL*" [1].

2.4.5 Concept Maps

Concept maps (Cmaps) is a non-logic based tool for organizing and representing knowledge in an informal way [3].

Cmaps represent meaningful relationships between concepts linked by words to from a semantic unit. The concepts are included in circles or boxes while relations between concepts are represented by links connecting the boxes. The links are labeled, describing the relation between two concepts. Propositions result from the phrases composed by the concepts and the link label (concept - verbal phrase - concept).

According to Garcia *et al.* [14] Cmaps are very useful in facilitating the visualization and discussion, and in providing domain experts with a tool that could be used to declare the primary elements of their knowledge. Cmaps support the declaration of nodes and relationships; it was easy to assimilate these to classes and properties. [14]

According with Shneiderman, Cmaps have several advantages:

- Novices can learn basic functionality quickly, usually through a demonstration by a more experienced user.
- Experts can work extremely rapid to carry out a wide range of tasks, even defining new functions and features.
- Knowledgeable intermittent users can retain operational concepts.
- Error messages are rarely needed.
- Users can see immediately if their actions are furthering their goals; if not, they can simply change the direction of their activity.
- Users have reduced anxiety because the system is comprehensible and because actions are so easily reversible.

Novak [3], give some simple advices about how to construct Cmaps. His guidance steps can be enumerated as follows:

- Domain identification
- Context identification
- Focus questions identification
- Concepts identification
- Concepts listing and classification

In first place, we must know what we are talking about; well then, we must define a domain of interest. A domain has always a context and it must be defined as well e.g., creating a set Cmaps for work organization domain within an enterprise is a little different from creating Cmaps for work organization domain in a virtual enterprise environment.

A cmap results always from a specific question elaborated according with the domain of context. Each question may correspond to one cmap that answers to that question.

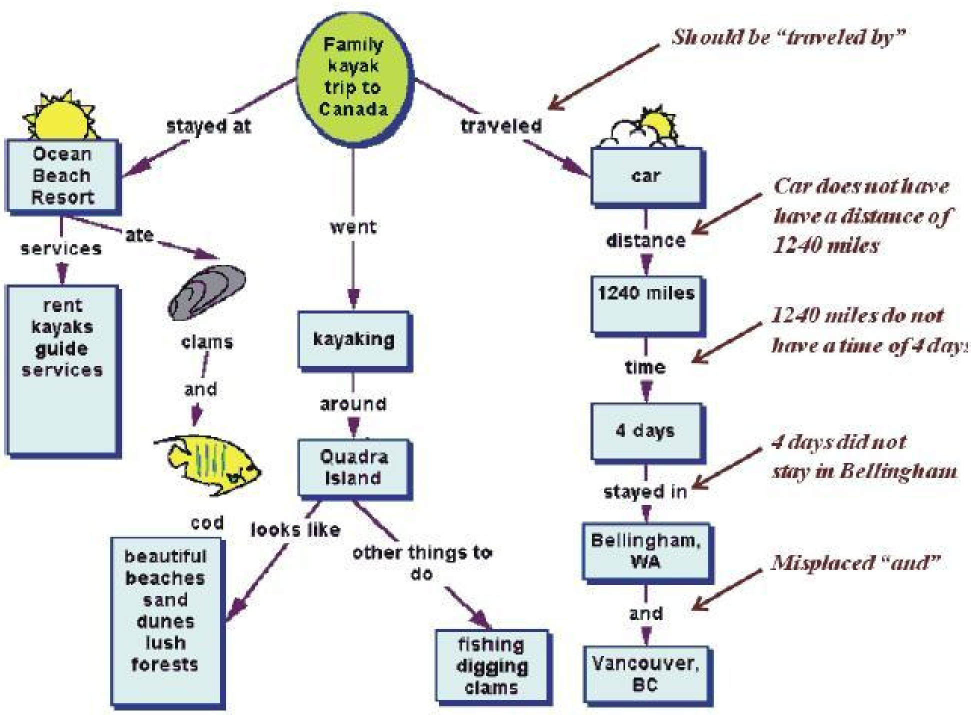
The fourth step is to start collecting the concepts and then make a list ordering the concepts from the most general to the less general. After that, we are able to build the concept map, which should be constructed in an iterative way, starting with a preliminary one and develop it. According to Novak [3], a cmap is never finished.

Despite of Cmaps simplicity and facility in its construction, several errors can occur, some more complex than others, making a cmap hard to read and to understand.

John Sowa [17] classified those errors into 3 categories:

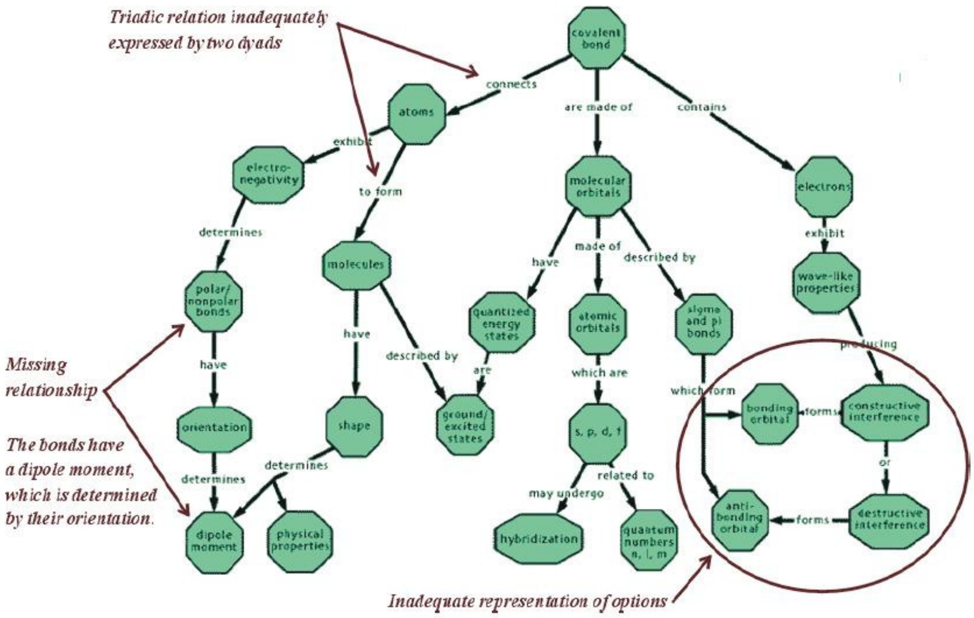
- Typical errors
- Worse errors
- Sophisticated errors

Figure 2-13 CMs Typical Errors



(Source: http://www.jfsowa.com/talks/cmapping.pdf)

Figure 2-14 CMs sophisticated errors



In our opinion and despite of the errors liable to happen when creating a cmap, they still are extremely useful among an organizational environment when we want to get a clear picture of the thoughts and ideas from people and share that knowledge within and between organizations. Cmaps can be used with ontologies to create more formal and regular knowledge that can be converted into formalized notations in order to enable Cmaps to become easier to exchange between Knowledge management systems.

Cmaps flexibility and relevance make this tool useful in a quantity of domains, namely:

1. Knowledge organization and creation
2. Collaborative learning
3. Domain summarization
4. Browsing tool

At the organizational level, Cmaps can be used in several tasks, such as:

- Training
- Consulting
- Strategic planning
- Business Process Management
- Product Development
- Marketing
- Performance Management
- Customer Service Support
- Workflow

- Human Resource Management

Well defined and with the right tools in a consistent Technology architecture, Cmaps are a powerful management tool.

2.5 Ontology as a Tool for Knowledge Representation

Knowledge Representation formalisms aim to make knowledge as explicit as possible in order to be feasible to share it and exchange it as reliably as possible. Ontological level of Knowledge representation is based in Conceptualization. Every Knowledge Based System is related with explicit or implicit conceptualization. Conceptualization comprises a simplified vision of the objects, concepts, and other entities that are assumed to exist in some area of interest and the relations among them [25].

John Sowa, mentioned in his book *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, that *"knowledge representation is a multidisciplinary subject that applies theories and techniques from three other fields:*

- ***Logic** provides the formal structure and rules of inference.*
- ***Ontology** defines the kinds of things that exist in the application domain.*
- ***Computation** supports the applications that distinguish knowledge representation from pure philosophy.*

Without logic, a knowledge representation is vague.. Without ontology, the terms and symbols are ill-defined, confused, and confusing. And without computable models, the logic and ontology cannot be implemented in computer programs. ***Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domain**."*

The use of ontologies come to confirm that all the efforts around knowledge representation formalisms (e.g. frames, semantic networks, logic…) are not much relevant without a good content organization about a specific domain of knowledge [89].

As mentioned earlier at the beginning of chapter 2, knowledge sharing and reuse is the main challenge of KR either from an organizational point of view (exchanging knowledge between humans) or in an information technology environment (exchanging knowledge between computerized systems). Ontologies made possible a shared understanding of a specific domain being able to be used in describing a business process or even an entire organization.

To understand ontologies goals and scope it is first needed to define ontology. There are several definitions in the literature, however, as a baseline, we can say that an ontology is a shared, formal conceptualization of a domain.

Ontology was simply defined by *Tom Gruber* as an explicit specification of a conceptualization *[25]*. This definition was upgraded in the computer science context in which an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members) *[74]*.

We can resume ontologies functionalities and scope in the following table [12]:

Ontology	
Functionality	**Application scope**
"Ontologies build upon a shared understanding within a community. This understanding represents an agreement over the concepts and their relationships that are present in a domain."	*"Ontologies use machine processable, logic-based representations that allow computers to manipulate ontologies. This includes transferring ontologies among computers, storing ontologies, checking the consistency of ontologies, reasoning about ontologies etc".*

The ontology distributed characteristic and its capability of agreements formalization between concepts, allowing a uniform vision about a specific domain within and between organizations, linked to the capability of being manipulated by computers in an automated way, is either from business point of view or Artificial intelligence point of view in step with the new challenges with the representation of conceptual systems of knowledge. Regarding a knowledge management systems, it "*must support the integration of information from different sources, wherein a decision maker manipulates information that someone else have conceptualized and represented, so the system must minimize ambiguity and imprecision in interpreting shared information. This can be achieved by representing the shared information using ontologies. [75]*". However, not every type of knowledge can be represented in an easy way through ontologies. Knowledge representation formalisms can be very simple, for instance defining a set of reserved natural language words with an agreed community meaning. This is an example of a textual formalism, but there are also graphical ones such as concept maps. These graphical representation formalisms are also called diagrammatic formalisms and they are quite simple and easier to interpret (for humans), but they are not so easy to be represented in an ontological level due their informal characteristic, nevertheless the frame-based formalisms are more likely to be represented at the ontology level.

2.6 *Graphical Tools for Knowledge Representation Formalisms*

The use of graphical knowledge representation formalisms with a representational vocabulary agreement of terms of conceptualization of the universe of discourse is a new high potential approach in the Knowledge Management context. There are some tools that combine these factors in an all-in-one environment. Some of those tools can, in fact, be used as visual knowledge representation tools for several formalisms. Nevertheless, some tools just permit a pure visualization of the knowledge structures while others allow to build and to edit knowledge structures trough direct handling of drawing objects.

The idea of having intuitively built and managed representations of conceptual systems that can be automatically translated to formalized knowledge, able to be processed and exchanged by a network of KB systems, is a fundamental to make possible the use of sophisticated KM systems in organizations by non KR specialists.

In the next point will be presented some tools that allow a graphical representation of knowledge without the need of any kind of logic-based programming language skills. The selection was made according to the distribution model and versatility of the tools as well as with its potential application in the knowledge representation domain.

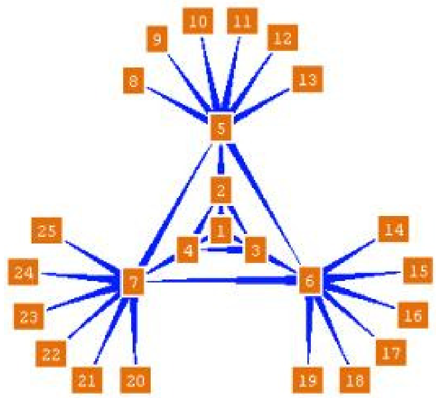
In this description, some of the ontological capabilities of the tools will mentioned.some of its ontological capabilities.

2.6.1 TouchGraph[2]

TouchGraph (TG) is a set of interfaces for **graph visualization** and **information categorization** using spring-layout and focus-context techniques, developed by Alex Shapiro. Focus-context refers to a visualization that presents the user with an information overview while being able to examine specific details [11].Current applications include a utility for organizing links, a visual Wiki Browser, and a Google Graph Browser which uses the Google API.

[2] www.touchgraph.com

Figure 2-15 - TouchGraph Spring-Layout



Source: http://www.touchgraph.com

TG has a "radius" feature used to denote the degree of similarity [11]. A radius of 1 means that the surrounding nodes will be directed similar to the selected node. A Radius of 10 means the items are less similar [11].

Figure 2-16 - TG Layout using Google Browser for fe.up.pt with radius 1

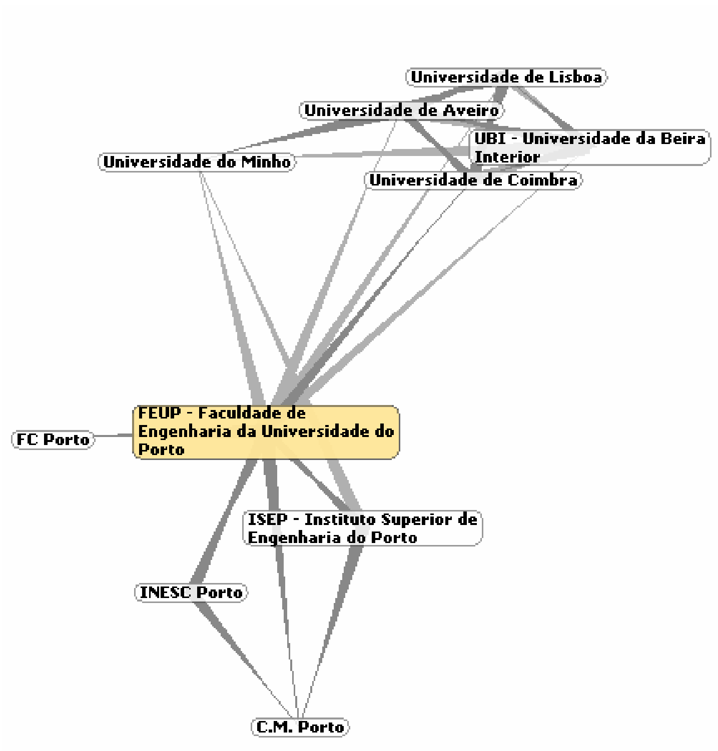


Figure 2-17 TG Layout using Google Browser for fe.up.pt with radius 2

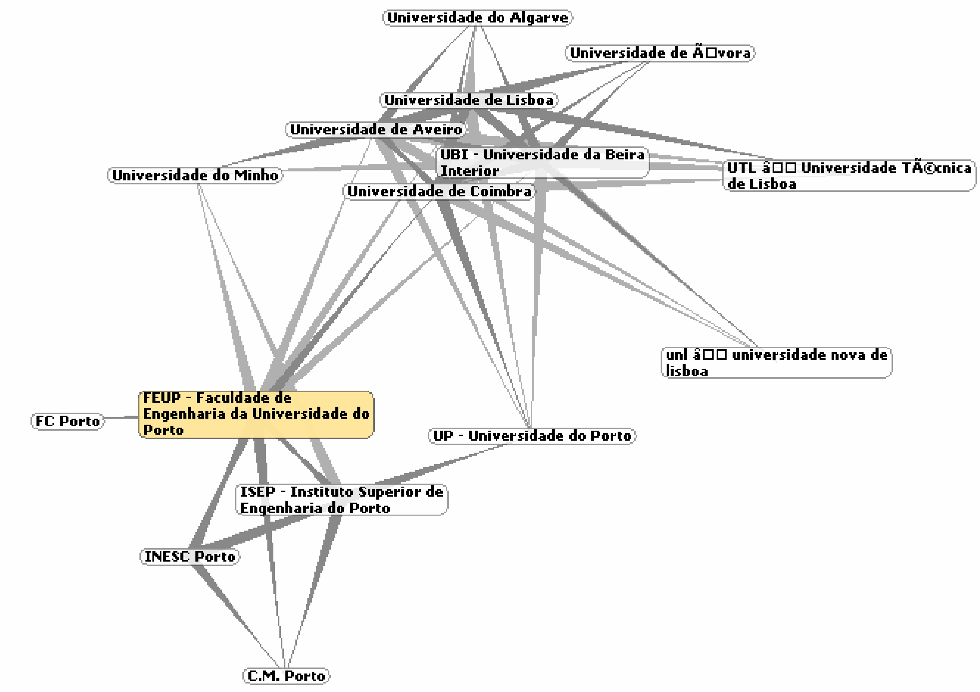
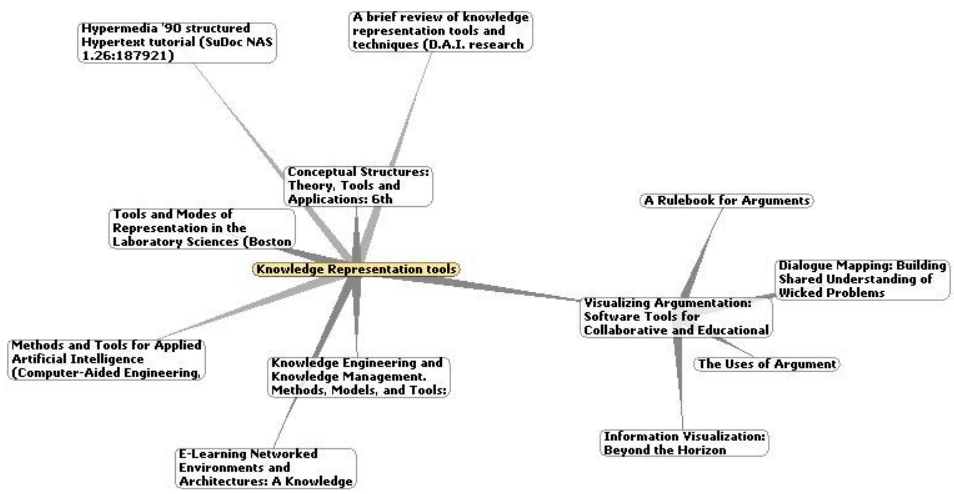


Figure 2-18 TG Layout using Amozon Browser for query "Knowledge Representation Tools"



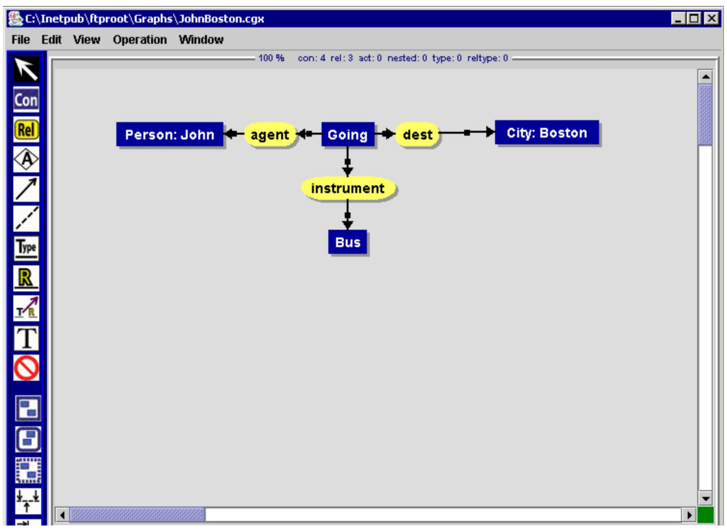
TG allows the user to zoom out or in to the graph. Nodes may be expanded to show additional linked nodes or they may be clicked on to view additional information in a pop-up window. TG renders networks of information concepts as interactive graphs that lend themselves to a variety of transformations. By engaging with the visual image, a user is able to navigate through large networks of information and to explore different ways of arranging the network's components on the screen because the view can be restricted to a local portion of the graph.

TG *LinkBrowser*, for example, can be used as a basic graph editor, but creating graphs is still quite hard work, especially if we want to include external links or other extra information for nodes. TG's major goal is to visualize huge amount of information and for information categorization. It is possible and simple to add more nodes to a graph but it is not acceptable to build a graph from the beginning. However TG skill being an interesting tool since it is an open source product and can be used in several projects. *Froudbuster* [3], for example, is an *Firefox* extension which provides detailed information of relations between buyers and sellers. The main goal is to deliver information about the trustfulness of a certain given eBay members. *Froudbuster* gets a visualization of the connections of that certain member to other members. You can choose different views of graphs. You can also click on every node of this graph to get detailed information of the buyers, sellers and the items.

2.6.2 CharGer

CharGer[4] is a Conceptual Graph (CG) editor intended to support research projects and education [70]. It is a graph authoring tool allowing the construction of conceptual graphs exactly as the Display Form of CG demands. (see section 2.1.2). *CharGer* permits all graphic elements manipulation as a reliable CG (See figure above).

Figure 2-19 CharGer GUI



[3] http://d-sided.de/fraudbuster/index.php

[4] http://charger.sourceforge.net

Source: http://charger.sourceforge.net/

CharGer can export a CG in vector graphics formats: EPS, PDF, SVG and EMF or Raster (bitmap) graphics formats: GIF and PNG. A CG can also be saved or retrieved from files in CGIF standard interchange format. These graphs have the extension”.*CGF*” and are portable across all platforms [70].

CharGer it is a very useful tool for concept mapping, in particularly for CG’s research teams. Nevertheless its exclusive Conceptual Graphs approach is a limitation for its use among the broad KR domain.

2.6.3 CmapTool

IHMC[5] Concept Map tool is a free concept mapping tool ideal for authoring, since it is easy to use and it handles external links, automatic layout as well as multiple output formats.

CmapTools is a software that offers a work environment that allows the construction, collaboration, sharing and constant improvement of knowledge Models such as concept maps. *CmapTools* became an easy-to-use navigation tool for “Hyper Media” environments. Due his graphical nature, the information is more accessible. However its HTML output has some limitations, though concept map is exported as a single image, which can be impractical when large graphs are considered. Concepts can be visually styled with colors, shadows, background images, etc. The auto-format feature can automatically position concepts relative to each other, to make a well-proportioned map. Concept maps can be exported in a wide variety of formats: as an image, a web page, an outline, propositions as text, an XML document.

IHMC's CMapTools is a good concept mapping tool for those interested in sharing maps across repositories and organizations, and delivers a number of visual styling options.

2.6.4 Protégé Plug-ins

Protégé OWL

Protégé is an open-source ontology development environment developed at Stanford Medical Informatics, originally for medical proposes in construction of complex bases of knowledge, but however is domain-independent. *Protégé* provides a reasonably intuitive editor for classes, properties, instances etc, and has extensions for ontology visualization, project management, software engineering, and many other modeling tasks.

[5] http://cmap.ihmc.us/index.html

Protégé is very flexible and extensible software, that was build into a system based in a plug-in architecture, that allows the users and developers to create new GUI widgets, application Logic an input/output extensions. The *OWL Plug-in* is a major extension of Protégé, allowing users to load, save, edit, visualize and classify ontologies in the Web Ontology Language, and to acquire Semantic Web contents. The *Protégé OWL* provides interfaces to Description Logics reasoners such as Racer and allows accessing other services provided by HP Lab's Jena library.

Protégé OWL Plug-in is one of the most popular editors for *OWL*.

Protégé OWL is not complex, however, it is necessary that users have some expertise about *OWL*, or about some object orientation, or about some graphical environment for data base development, because some concepts are analogous.

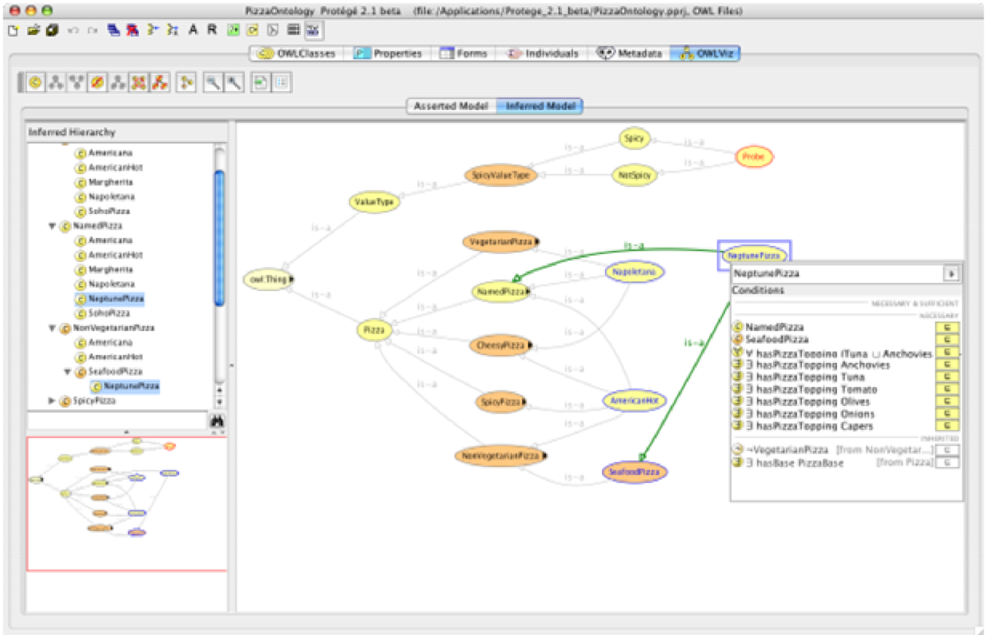
Features such as *WYSIWYG HTML* and forms editing, *XML* name space support, version control, import and merging of ontologies, data entry wizards, *JDBC* back end, and multi-user support among others, make the protégé owl a very interesting tool for ontology authoring. Notwithstanding its powerfulness and usefulness *protégé OWL* does not have visual authoring. *OWLViz* came to cover this gap.

OWLViz Plug-In

OWLViz is a visualization plugin for Protégé developed as part of the CO-ODE project (*www.co-ode.org). "It enables the class hierarchies in an OWL Ontology to be viewed and incrementally navigated, allowing comparison of the asserted class hierarchy and the inferred class hierarchy."*[6] *OWLViz* is integrated with *Protégé OWL*, but it not allows users to edit ontologies trough the graphical scheme of the ontology. It's a pure visualization feature. However this attribute revealed to be very interesting as complement to the OWL Plug-in. The color scheme used is the same between *OWLviz* and *OWL Plug-in* which allows the user to easily identify possible errors that may have been committed. Primitive and defined classes can be distinguished, computed changes to the class hierarchy may be clearly seen, and inconsistent concepts are highlighted in red (http://www.co-ode.org/downloads/owlviz/). Next picture shows the "Pizza Ontology" classical example with OWLviz.

[6] http://www.co-ode.org/downloads/owlviz/OWLVizGuide.pdf

Figure 2-20 OWLViz Plug-in used With Protégé Plug-In



Source: (http://www.co-ode.org/downloads/owlviz/)

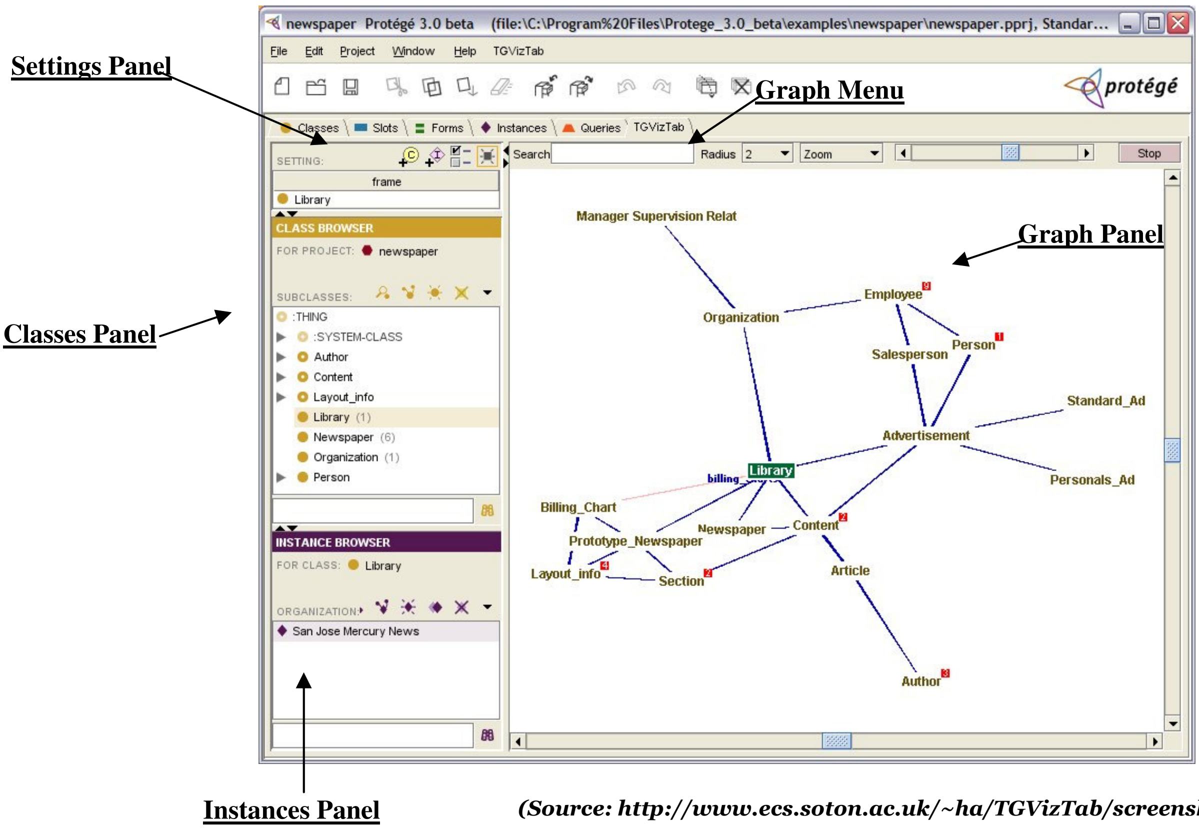
TGVizTab

TGVizTab (shown in figure 2.21) is an ontology visualization plug-in for Protégé based on TouchGraph[1]. This plug-in aims at helping Protégé users to visualize and browse their ontologies and KBs via interactive dynamic graphs [15].

In short TGVizTab plugin has the following features:

- Visualizing classed and instances
- Network depth control
- Change graph colors
- Different slots can be displayed in different colors
- Hide/show individual slots and nodes
- Geometric and hyperbolic zooms
- Graph rotation
- Node search
- Save/load graphs and settings

Figure 2-21 - TGVizTab ScreenShot



(Source: http://www.ecs.soton.ac.uk/~ha/TGVizTab/screenshot.htm)

TGVizTab users are given full control over the color and visibility of each ontology relation, which can be modified in the Settings Panel. Graph edges are labeled after the ontology relations they represent. To avoid clutter, edge labels are only displayed when having the mouse over the edges, or over their originating nodes . Right clicking a node brings up the Node Menu .This menu contains options for hiding, expanding, and collapsing individual nodes, and for viewing their Protégé description forms [15].

Protégé OntoSphere3D plug-in

"*OntoSphere3D* is a Protégé plug-in for ontologies navigation and inspection that uses a visually enriched 3-dimensional space [39]."

"*OntoSphere aims to graphically represent both the taxonomic and the not taxonomic links as well as selecting and presenting information on the screen at an appropriate detail level according to what is relevant to the user's interest* [37]."

The plug-in conceptual elements are represented as follows:

- *concepts* are *spheres*
- *instances* are *cubes*
- *literals* are *cylinders*

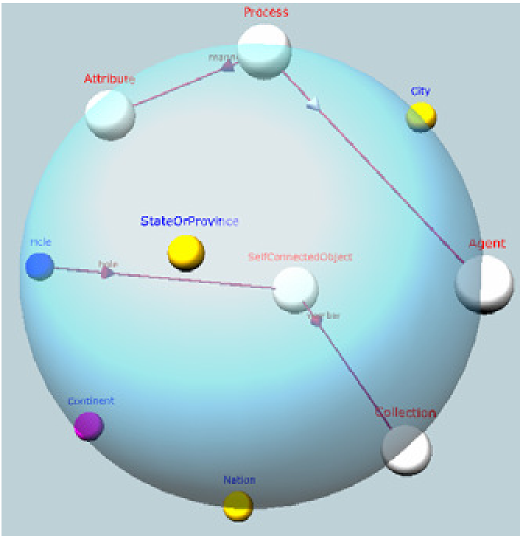
- *relationships* between entities are symbolized by *arrowed lines*

"OntoSphere3D allows direct manipulation operations such as zooming, rotating, and translating objects, and introduces more dimensions to convey information on the visualized knowledge model as the color or the size of visualized entities[39]."With this approach we have a better organization of the visual occupation of the ontology and a better navigation through the ontology.

OntoSphere3D has different scene managers - through rotation, panning and zoom - for visualizing de ontologies to the mouse click range [37], [38],[39]:

- **Root Focus** permit the visualization of the concepts within a 3D globe and the concepts are on its surface in which concepts are represented as spheres and the blue ones represent the terminal concepts. The size of each sphere depends of the number of sub elements.

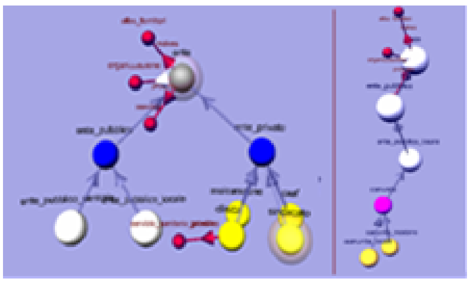
Figure 2-22 OntoSphere3D Plug-in Root Scene



Source: (http://ontosphere3d.sourceforge.net/userGuide.html)

- **Tree Focus** *"displays the hierarchical structure as well as semantic relations between classes* [37]."

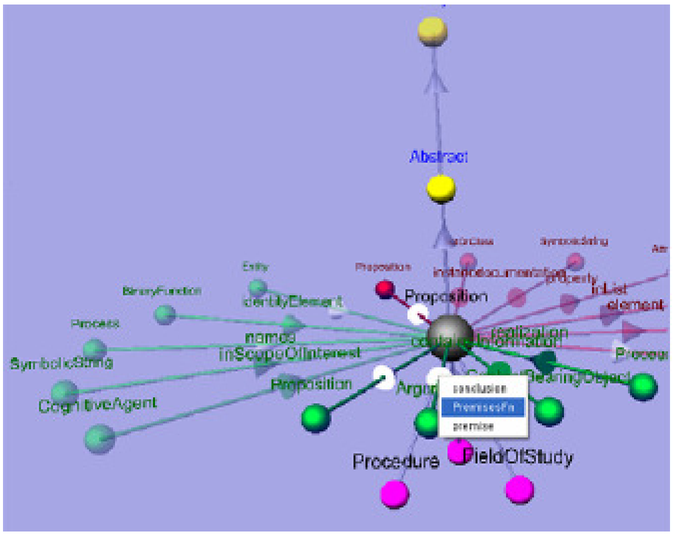
Figure 2-23 OntoSphere3D Plug-in Tree Focus Scene



Source: (http://ontosphere3d.sourceforge.net/userGuide.html)

- **Concept Focus** *"perspective depicts all the available information about a single concept, at the highest possible level of detail; it reports the concept's children and parent(s), its ancestor root(s) and its semantic relations, both the ones directly declared for the given concept and the ones inherited from its ancestors* [37]."

Figure 2-24 OntoSphere3D Plug-in Concept scene



Source: (http://ontosphere3d.sourceforge.net/userGuide.html)

Each scene organizes information on screen in a different detailed perspective and the concepts can be directly manipulated using the mouse.

In spite of all the advantages of OntoSphere3D, it doesn't allow directly manipulation of the ontology's, that is, add, edit, and delete concepts or relations through the graphical scheme.

This plug-in is very useful grouped with protégé OWL plug-in for huge concept systems visualization and for individual concept analysis in the entire system. All of this is accessible only trough the click of the mouse.

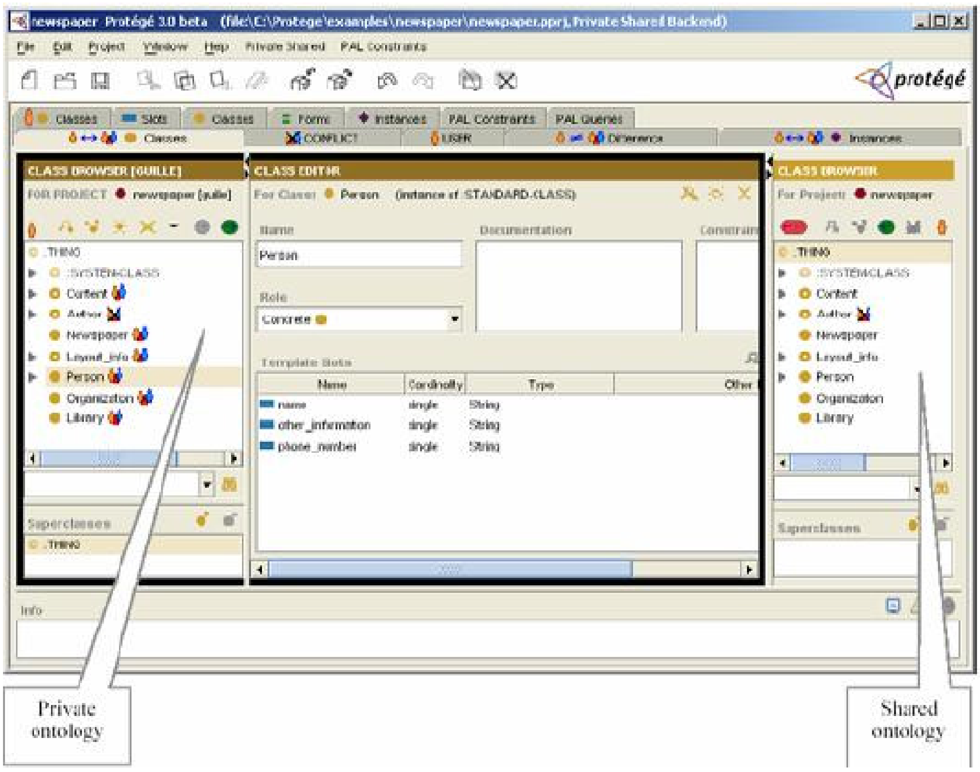
CO-Protégé

According to [10], CO-Protégé is a set of plug-ins that extends Protégé in order to support a collaborative developing of an ontology.

CO-Protégé can be seen as a tool for ontology development in a collaborative way, where users can perform theirs tasks through private and shared areas, discussion support and collaborative activities tracking.

In this approach there are two ontologies: a private one and a shared one. In practice we can have two workspaces, one for stand-alone edition and other for sharing ontologies. These workspaces are representing as Tabs in the software tool as show the figure above.

Figure 2-25 Co-Protégé Plug-in



Source: (Díaz, 2005)

We can build an ontology (privately), and then we can publish it into the shared ontology.

The shared and private tabs are overlapped which is easy to change between private or shared ontologies. It is important to know that only private ontology tab has the same features for ontology edition as Protégé core itself. Shared tan doesn't have that features and it is updated by the publishing functionality. The class, instance and slot tabs are present in both private and public tabs.

There is also a Conflict Tab. Conflict tab allow the discussion of a specific ontology. Finally there is a User Tab that allows the creation and edition a user profile and domain interests.

CO-Protégé Still a prototype, however it could be a trump card in the formal representation of conceptual systems giving to protégé an up-to-date feature.

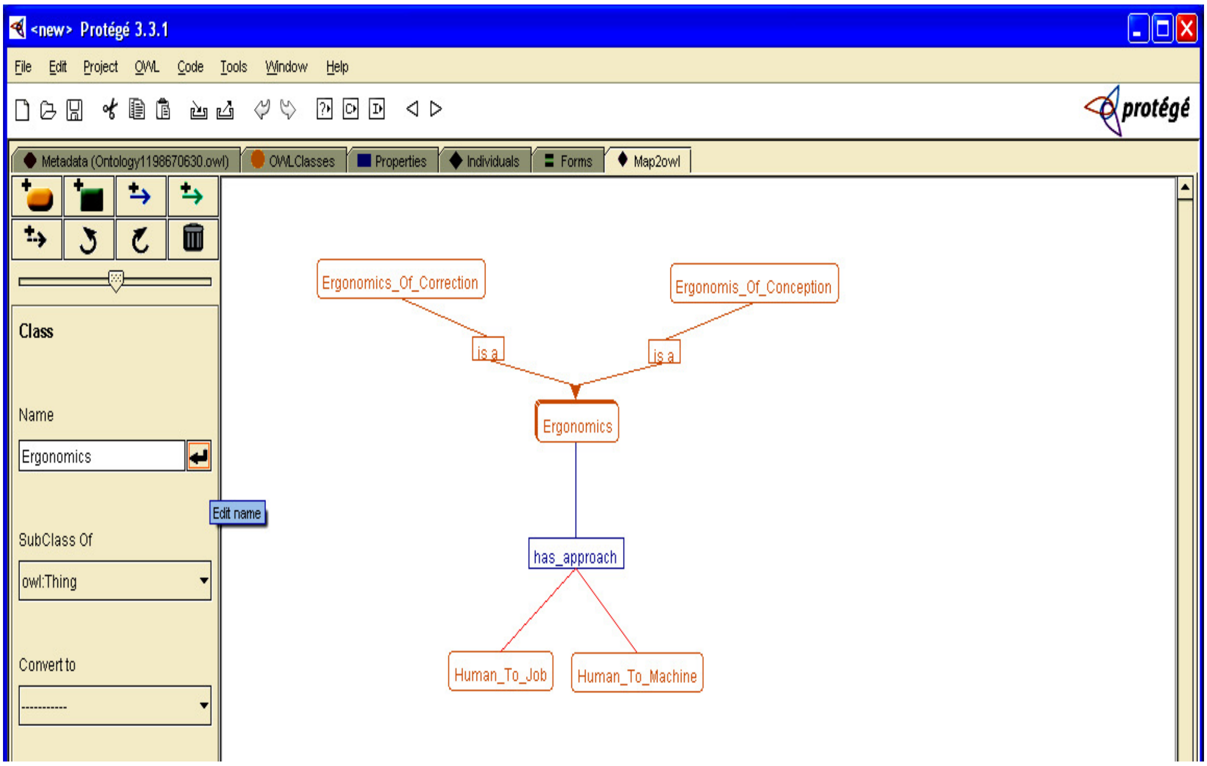
Map2OWL

Map2OWL[7] is a tab widget plug-in for Protégé that can be used to create and visualize ontologies, allowing direct graphical manipulation of OWL characteristics.

Deceloped by Alexander Garcia and his team, the first version of Map2OWL was released in the beginning of October 2007. The main goal of this plug-in is to facilitate the manipulation of OWL constructs in a graphical way and allow domain experts to easily build cmaps within Protégé without any concern about OWL technical aspects. Protégé will automatically construct the OWL classes and its constraints based on the concept map. Despite of its initial development state, Map2OWL is a very interesting and useful feature, simplifying the translation between concept maps and a more formal ontology specification.

In order to install the Map2OWL plug-in, we just have to download it and extract the files into the Protégé plug-in folder. Next, we must select the tab widget choosing configure option from the Protégé Project menu. The result is shown in the next figure.

Figure 2-26 Map2OWL plug-in

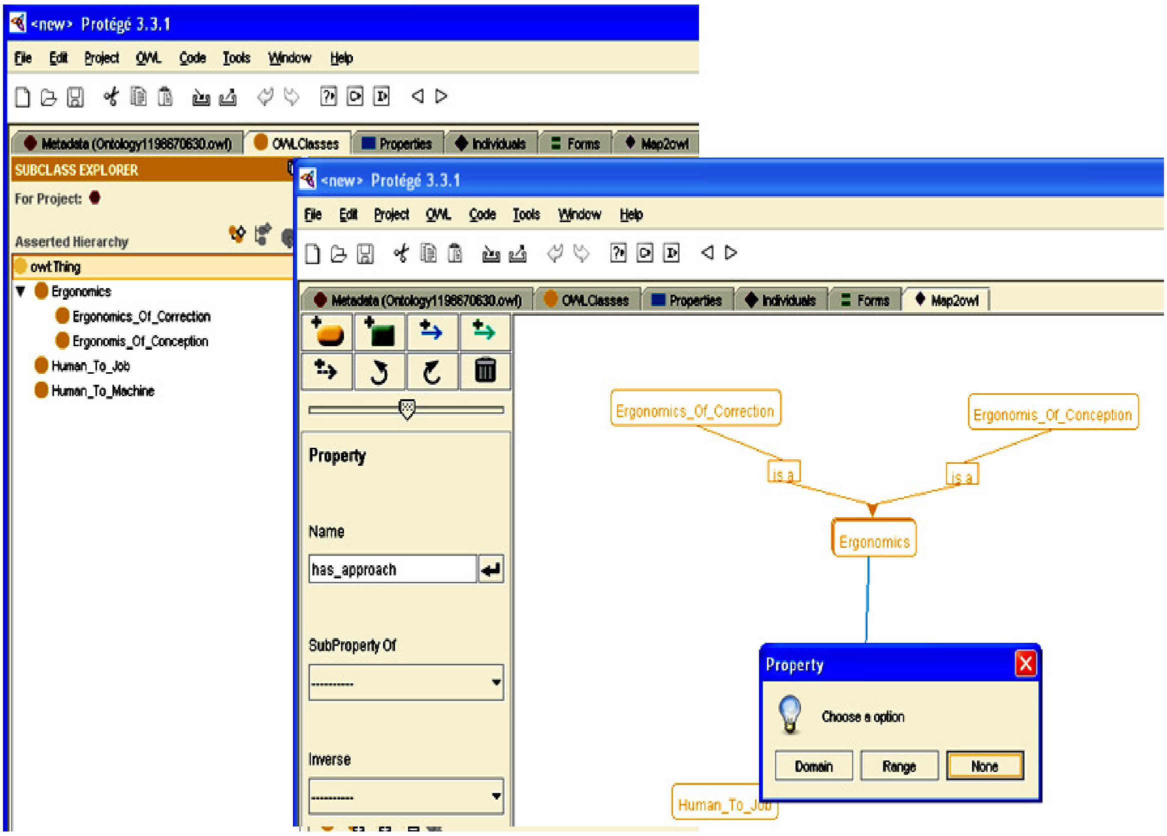


The environment is very simple and intuitive. Using “drag and drop” feature we can easily create classes, properties and relations while the OWL structures are immediately created (see next figure).

---

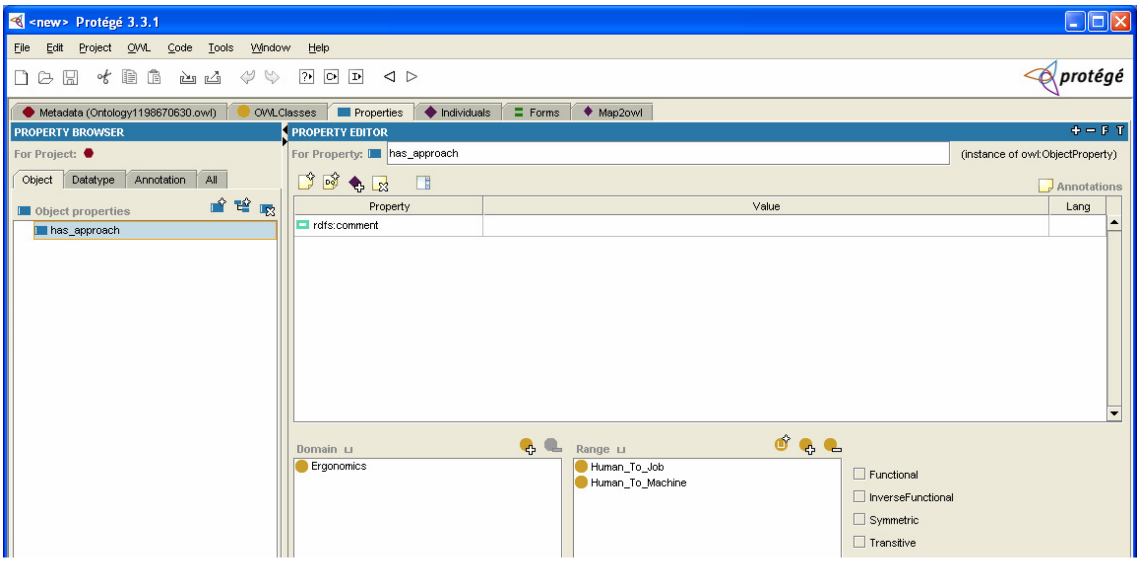
[7] http://map2owl.sourceforge.net/default.htm

Figure 2-27 Map2OWL – Direct translation form cmap to OWL



The OWL properties are also directly converted.

Figure 2-28 Map2OWL – Direct translation form cmap to OWL



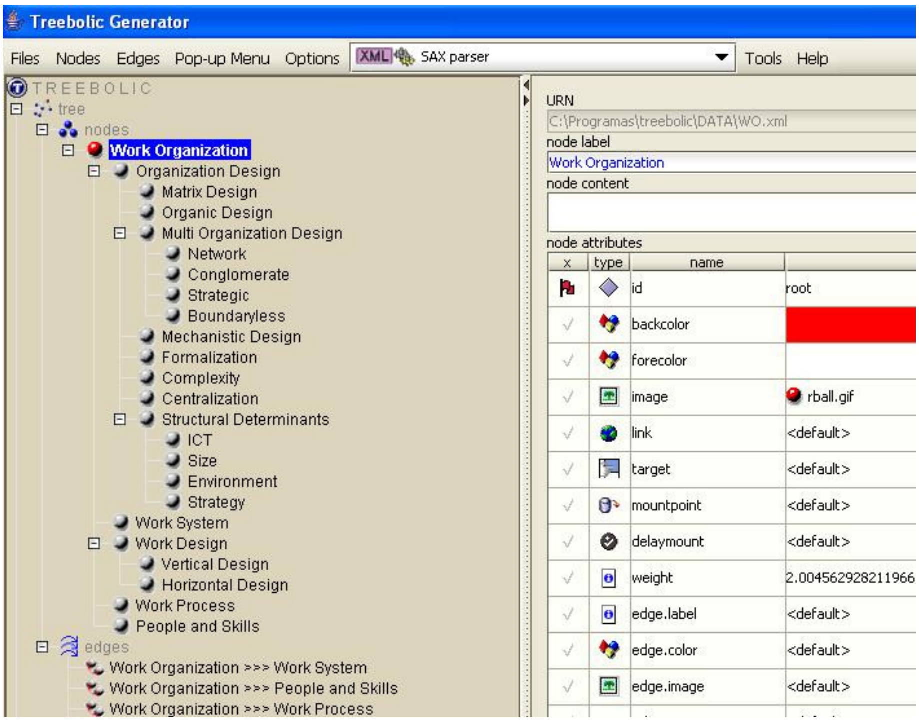
This plug-in come to take a step forward in the graphical knowledge representation at the ontological level.

2.6.5 Treebolic

Treebolic is set of tools for visualization of hierarchical data in a hyperbolic tree. *Treebolic* represent a hierarchy through a radial layout in a hyperbolic plan, with a simple navigation by choosing a specific node which is displayed in the center of the detailed representation. The rest of the nodes remain in the context with a smaller size [7].

In a hyperbolic approach, the user is able to navigate through the nodes and visualize the relations of the visible portion with the rest of the structure [40]; however it is not possible to edit relation labels between nodes for specifying it's kind of relations. For creating a map using treebolic, we must first generate de xml trough the TreeBolic XML Generator as the next figure shows.

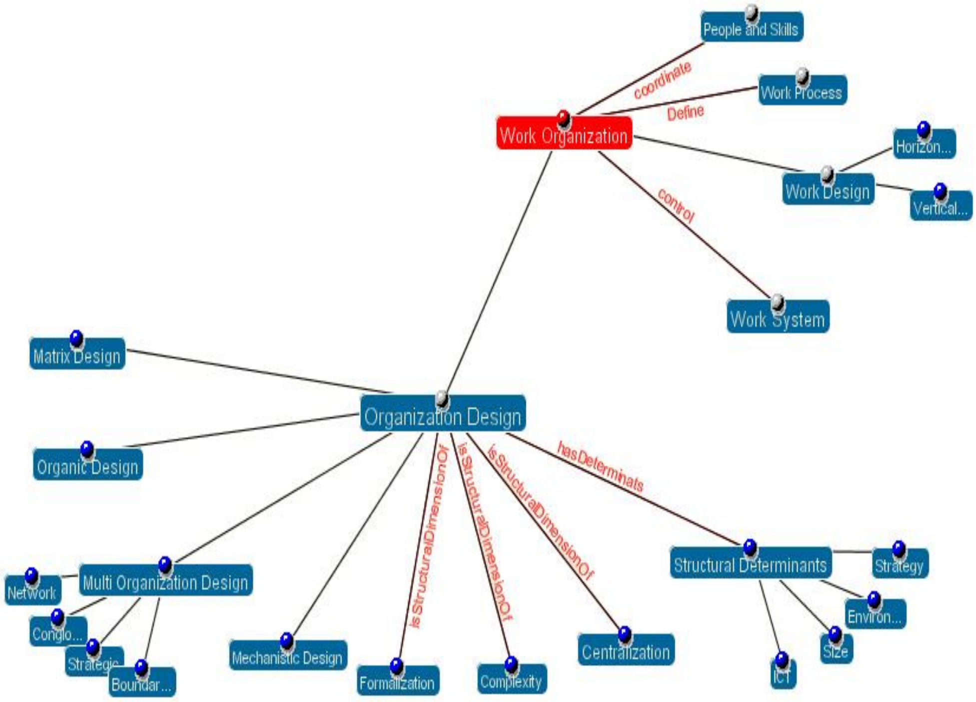
Figure 2-29 - Treebolic Generator Example



Treebolic uses a hierarchical taxonomy in a tree classification where we may assume that the top (parent) nodes are classes and the child nodes are sub-classes and the terminal nodes are instances. Edges – the links between nodes - allow the definition of relationships rather than only parent-child relations. If we aim to build Ontologies, *Treebolic* doesn't aid too much. Ontologies may need more than hierarchical relationship, such as network structures and *Treebolic* scope is essentially hierarchical classification and categorization.

Figure 2.29 shows a picture a simple example of an attempt to create a simple ontology trying to define other relationships between nodes beyond parent-child ones. However when trying to export it to XML, labeled relation are lost.

Figure 2-30 TreeBolic Map trying to define an ontology



Another important aspect is that *Treebolic* does not import/export owl files.

Treebolic is a very useful tool to visualize huge amount of information, thankfulness to its hyperbolic approach, but is far to be more than a pure visualization tool in terms of graphical KR, since it doesn't have direct graph editing features.

It ere presented some of the most interesting and useful tools within the compass of graphical knowledge representation. All the tools presented are open source or freeware, and for that reason they deserved evidence, although there is other commercial software's used for the same propose but with the same investigation basis.

There is not an ideal or perfect tool concerning to knowledge representation and sharing yet, and there is a long track to be on the way to it. However the notion that the old track to knowledge representation was usurped is clear. There is a new drive and a new sense of meaning in knowledge management and this new tools used with ontologies is an effective trial on this trail.

Chapter 3

Ontology Development with Concept Maps

This Chapter describes the state-of-the-art of the use of concept maps as tools for collaborative ontology development. The use of cmaps in this context represents a particular approach in ontology development, therefore, it is appropriate to present a set of the most important methodologies for ontology development.

KR is a multidisciplinary process and transversal to several areas, namely: logic, **ontology** and technology. These three areas complement each other. Logic provides a formal structure and a set o inference rules. **Ontology** defines the kind of things that exist in a specific domain, as well as the set o terms a vocabulary for its definition. Technology supports applications for knowledge representation. Without logic KR is unclear. Without ontology the terms are imprecise and confusing. Without technology, the logic and ontology can not be implemented in computer programs [90].

In the KR domain ontologies reveal a particular importance in a way that aid sharing and common understanding of domain concepts and their relations in a (more or less) formal way.

The question, however, is to what extend conceptual maps can answer to ontology building requirements, concerning to the art of acquiring and representing knowledge.

3.1 Ontology Development Methodologies

There are several technical and scientific approaches available in order to achieve an optimal drive to the development of ontologies. Some of them focus on ontology construction issues whereas others focus on ontology transforming and re-using aspects.

In spite of the current discussion in this area about which is the best way for ontology development, none standardized methodology emerged.

The different standardized methodologies may satisfy some practical needs on ontology creation but do not cover other ones. Nevertheless, there are some fundamental aspects that we should not ignore.

Among all the methodologies, we selected the following ones as the most relevant:

- TOVE
- METHONTOLOGY
- DILIGENT
- ON-TO-KNOWLEDGE
- KATUS
- ENTERPRISE
- SENSUS

3.1.1 TOVE (Toronto Virtual Enterprise)

Gruninger and Fox [63], proposed the TOVE methodology for the design and evaluation of ontologies based on their research in industrial environment. The methodology encompasses six steps which are:

- **Motivation Scenarios** - Motivation scenarios are the beginning of ontology development providing the description of the problems within the organizations. The capture of motivation scenarios gives us a first sight of the semantic of objects and its relations, at the same time, a sketch of a possible solution for the problem come up attached [51][63][64].
- **Informal Competency Questions** – According with an earlier specification of motivation scenarios, a set of questions will arise which describes them in an informal way. These are the questions that the ontology must answer [51][63][64].
- **Terminology** – At this stage a terminology of the ontology is specified using a formal language. All the domain of discourse objects and its attributes are formally identified [51][63][64].
- **Formal Competency Questions** – The next Step is to create formal competency questions which are obtained either from previous specifications of informal competency questions and terminology. Formal competency questions will define the drives for the specification of axioms [51][63][64].
- **Axioms** – Axioms are a set of prepositions describing the initial consensus about the action domain. At this stage the ontology is formal specified based on first-order sentences defining the terms and their interpretations. These axioms should provide answers for all competency questions [51][63][64].
- **Completeness Theorems** – a set of conditions are defined in order to provide evidence to the usefulness of the ontology [51][63][64].

3.1.2 Methontology

In spite of some techniques and activities need to be specified in more detail, Methontology is the most mature one and it's recommended by the FIPA (Foundation for Intelligent Physical Agents) [64]. According with Methontology framework the ontology development processes find out the main activities which are divided in three groups:

- Project Management Activities which includes the guideline for Planning, Control and Quality Assurance.
- Development Activities which includes Specification, Conceptualization, Formalization, Implementation and Maintenance.
- Support Activities which includes essential for building the ontology, such as, Knowledge Acquisition, Evaluation, Integration, Documentation and Configuration management.

The framework enables the construction of ontologies at the *knowledge level*, betting in a life cycle based on evolving prototypes with the methodology itself specifying the steps for performing the activities, the techniques used, the outcomes and their evaluation [*http://ontoworld.org/wiki/METHONTOLOGy].*

3.1.3 DILIGENT

DILIGENT methodology emerged in order to implement a suitable process to enable decentralized knowledge sharing such as in virtual organization environment.

DILIGENT methodology includes five main activities [65]:

- **Construction**, which aims to rapidly identify and formalize the concepts and its relations. A small team should be created, composed by domain experts, users, knowledge engineers and ontology engineers in order to develop the initial ontology [65].
- **Local Adaptation** is the next step where users should use and adapt the ontology to their local needs. According with the authors, each one has its local business requirements as its specific local needs. These local changes are submitted and may be or not, fully integrated in the centralized ontology [65].
- **Analysis** is the activity where the Control Board Editors analyze the local ontologies and the respective orders and decide which changes will be integrated in the next version of the shared ontology [65].

- **Revision** – The Control Board Editors should regularly analyze the shared ontology in order to avoid inconsistencies between local and shared ontologies [65].
- **Local Update** – Each time a new version of the shared ontology is distributed, it should be included locally in order to improve the use of knowledge [65].

3.1.4 ON-TO-KNOWLEDGE or CommonKADS

On-to-Knowledge (OTK) is an improvement of **CommonKADS** methodology by introducing, among others, specific guidelines for developing and maintaining the respective ontology. The feasibly study is based on [66] .OTK uses two-loop architecture which is composed of knowledge processes (KP) and knowledge meta-processes (KMP). Knowledge meta-processes (KMP) describe building ontologies in 5 steps, namely: Feasibly study, Kick-Off, Refinement, Evaluation, Application and evaluation [67]. Each step has numerous sub-steps, requires a main decision to be taken at the end and results in a special outcome.

Scoping and feasibility study – The purpose is to identify problem/opportunity areas and potential solutions, embedded into an organizational perspective. The result is the domain for the ontology based system to be developed.

Kick-Off – Like in software development we start with an ontology requirements specification document (ORSD). It describes what ontology will support and what the areas of the ontology application are. It also contains guidelines on designing target ontology. In detail, the ORSD contains the following information [68]:

- Domain and goal of the ontology
- Design guidelines
- Knowledge sources
- Potential users and usage scenarios
- Competency questions
- Applications supported by the ontology

Refinement phase. During this phase, a mature and application-oriented target ontology should be created according to the instructions contained in the ORSD [68] [69].

Evaluation phase aims checking the usefulness of the developed ontology and the software environment [68] [69].

Maintenance phase is a continuous process, during which all required changes of the target ontology are made [68] [69].

3.1.5 KACTUS

The **KACTUS** project focused on the issue of ontology development. An engineering approach is adopted, stressing modular design, redesign and reuse [70]. The adopted steps are:

Specification of the application. At this point we should achieve a vision of the application context and the components to model defining the list of terms and tasks [64].

Preliminary design based on relevant top-level ontological categories. This process will produce/design a global vision of the model. Other ontologies should be used and extended [64].

Ontology Refinement and structuring. This is the process where it should be defined the definitive ontology. All the modules should be combined in order to snap all the pieces consistently [64].

3.1.6 Enterprise Ontology [71]

[64] This methodology is also known as Methodology by Uschold And King due its authors and it lies in four main simple steps which are:

- Identify Purpose
- Building the ontology
- Evaluation
- Documentation

Identifying the purpose has to do with the exact understanding of the intended goal of the ontology that will be developed and the reason why it will be developed [64], [71].

Building the ontology is the process of ontology construction itself and encompasses ontology capture with the identification of the key concepts and relations, specifying the definitions and formalizing the terms associated with the concepts and respective relations. Coding is the step where the concepts and relations captured are explicit represented in a formal way. While the ontology is being built, it is important to be aware of the possibilities for integrating existing ontologies [64], [71].

Evaluation is the process of reviewing the ontology either considering technical issues, software environment and documentation [64], [71].

Finally the ontology must be **documented** [64], [71].

3.1.7 SENSUS

"This is an ontology for use in natural language processing and was developed at the ISI (Information Sciences Institute) natural language group to provide a broad-based conceptual structure" [64].

The developing process has the following steps [72]:

- Identify main terms
- Link main terms to SENSUS by hand
- Include nodes on the path to root
- Add entire sub trees using the heuristic:
    - If many nodes in a sub tree are relevant,
    - the other nodes in the sub tree are relevant
- Prune the skeletal ontology using heuristics

One of the main objectives in this work was the development of an ontology in work organization domain within a decentralized and multidisciplinary environment in which the development process could focus in a conceptual level and not so much ate the application level.

The intention would be to have, in a first stage, an high level ontology. The concepts and relations should be captured and represented in a simply way (preferably in a graphical way) without formal constraints.

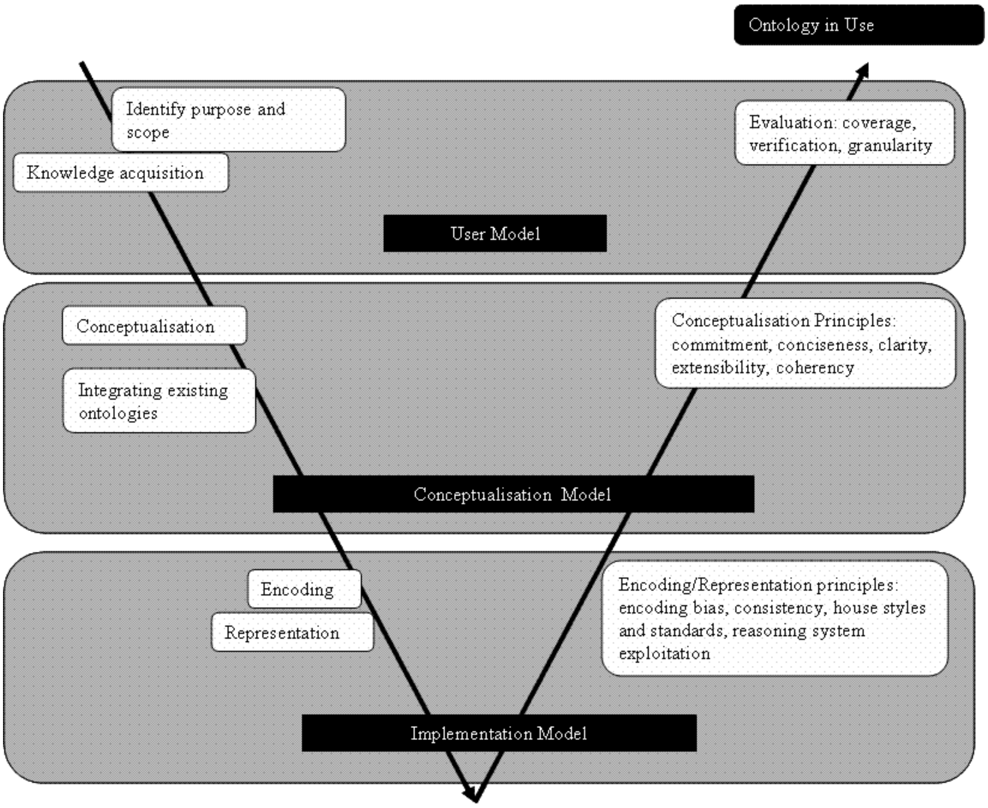
There are no standards or optimal methodologies; however there are some aspects of each one that fits in the desired development context and which have inspired the development of the ontology.

According to the previous requirements, Enterprise Ontology, TOVE, KACTUS and DILIGENT were selected in a first analysis because they are the most generic and simple ones with good documentation of the several stages which fit our intentions. Nonetheless, KACTUS is application-dependent (is built on the basis of an application knowledge base) [64] which is not quite suitable concerning the main drives for developing our ontology. On the other hand, TOVE recommends formalization of knowledge using formalisms such as FOL or Knowledge Interchange Format (KIF)[8] for building the ontology and this was not in our plans.

3.1.8 Other methodological approaches

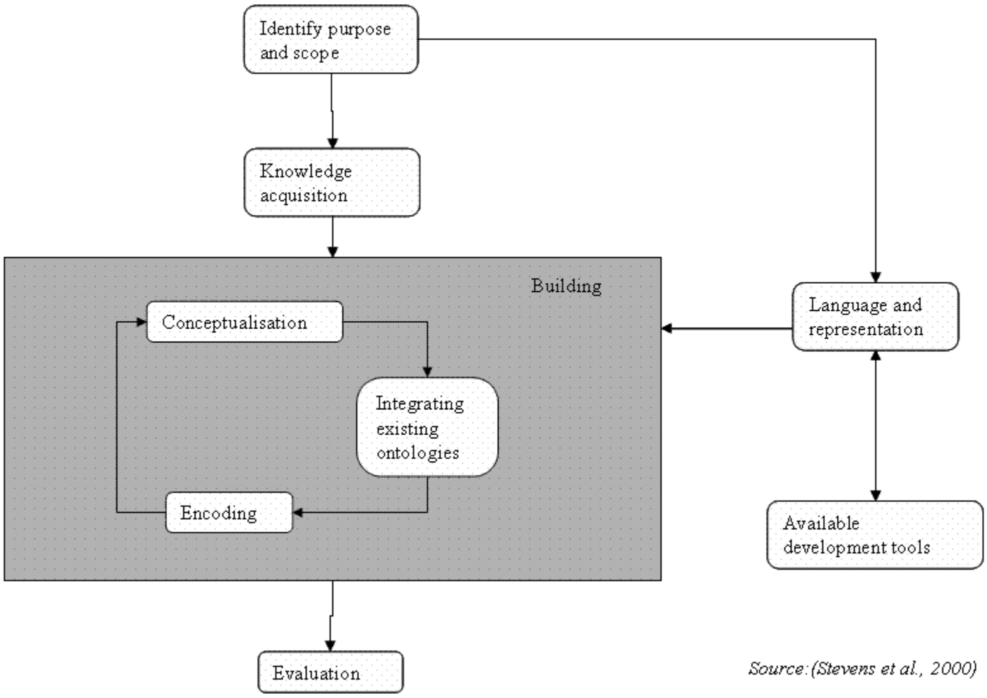
Figures 3.31 and 3.32, represents a methodology and life-cycle for building ontologies, inspired in the software engineering V-Process model. The left side illustrate the process in building ontologies and the right one the guide lines,_principles and evaluation. The overall process moves through a life-cycle, as depicted in the second figure [19].

Figure 3-31 V-Model methodology for building ontilogies



[8] KIF is a Logic-based formalisms. KIF is similar to First-Order Logic (FOL) but with an ASCII syntax

Figure 3-32 **The ontology building life-cycle**



In can be found other schemes representing methodologies for ontology building and its life-cycle, but, in summary, all strike to the same requirements and all of them demonstrate that ontology development is not a static process but an iterative task that crosses several stages.

3.2 Representing Ontologies with Concept Maps

An informal representation can illustrate information explicitly, but only through a formal way it is possible to share and interpret an ontology among computer systems. Well, this is where the Cmaps sin lies! Despite of the similarities with domain ontologies, Cmaps do not include a formal language, but just a set of recommendations for their construction; besides, Cmaps have some difficulty to represent large ontologies because several Cmaps shape only a part of the whole ontology. Furthermore adding asserted conditions (such as necessary, necessary and sufficient) is not possible.

Nevertheless, that lack of a formal language in knowledge representation is what makes Cmaps so easy to use for everyone.

Any way, a version of CmapTools called CmapTools Ontology Editor (more detailed in the next section) has been developed that allows users to work with ontologies. CmapTools COE keep the fundamental characteristics for concept construction without a restriction to a hierarchical model, since relations between concepts are transversals and arbitrary. COE keep also the interaction with the model allowing seeing, navigating and editing. The main change was the attempt to give to COE more interoperability and formalized notations meeting ontological agreements in order to allow the producing of information in more formats such as OWL and RDF, permitting as well reusing and sharing information. In this way COE came to cover a gap between the informal nature of conceptual maps and the formal nature of machine-readable ontology languages [4].

Ontology development is not a static process but an iterative task that crosses several stages. It is a structured process where several other processes take place in order to achieve knowledge elicitation, such as:

- Argumentative process
- Collaborative process
- Iterative process
- Evolution and evaluation process

This means that when creating an ontology, knowledge engineers argue about terms and relationships [19].Therefore collaborative interaction of arguments increase the amount of information attached to the concepts or relationships [19]. The arguments derived from discussions can be commentaries, files or even other maps. In general, discussions take place at forums, e-mail messages, chats, etc, which becomes necessary for knowledge engineer to filter the messages and formalize that knowledge. This task is certainly hard to perform and, for sure, not all information can be retrieved.

In summary it can be said that methodologies or tools for ontology building must address the **Collaboration Support** and **Argumentative Support.** These are features unquestionably necessary, aiming the new parameters of the ontology development discussed earlier in this chapter. Furthermore, pointed in [5] **Ontological Grounding Support** and **Guidance for Domain Experts** are essential requirements for ontological methodologies and tools. According to [5], domain experts are the central performer on ontology elicitation process, because they have the experience and knowledge about a specific area. However they have little or no training in using ontology modelling tools [5] and the most of the tools are build for software specialist or knowledge engineers (e.g. Protégé).

Grounding support can be achieved in two ways. Methodologies and tools can use fundamental design patterns of foundational ontologies to provide guidance to domain experts. The second approach is to use a foundational ontology as a base model for the definition of more specific domain [5].

Cognitive support[9] is another interesting and valid requirement in knowledge development domain. According to Garcia [5], cognitive support for ontology development is required in order to facilitate the useful/exchange of information and at the same time record the entire process. As argued in [5], cognitive support is used to leverage innate human activities such as visual information processing to increase human understanding and cognition of the problems. Conceptual maps perform at this level an important role.

Cognitive support is one of the most important requirements for knowledge engineering tools. In order to know if a specific tool has cognitive support is necessary to know which tasks need this kind of approach. Knowledge acquisition, modeling, verification and usage are the main tasks of knowledge engineering that cognitive support should address.

According to [9] leaded a research work where he went further on the specification of what kind tasks require cognitive support. Regarding surveys, reviews and their own experience the tasks were grouped in three categories for a simpler analysis:

- Navigation – navigation support for understanding, discovery and search.
- Modelling – modeling support in the task of constructing the ontology.
- Verification – visual checking support.

Within these analysis parameters of cognitive support, several tasks can be considered depending on the domain of interest in question. So, it's important to refer that the tasks listed below are neither standard nor casuals.

According to [9] and concerning navigation features, tools should provide:

- *Overviews and support top-down exploration of the ontology.*
- *Support slot-based browsing.*
- *Allow users to view query results.*
- *A mechanism for saving, annotating, and sharing views.*

On modeling features, tools should provide:

- *Graphical editing techniques*
- *Editing navigation*
- *Support ontology reuse*

Concerning to verification features, tools should provide:

- *Identify incoming relationships*

[9] Cognitive support is used to leverage innate human abilities, such as visual information processing, to increase human understanding and cognition of challenging problems. [91]

- *Incremental navigation*
- *Browse multiple and complex relationships at once*

3.2.1 CmapTool COE

*CmapTools Ontology Editor (*COE) software provides via Concept Maps a complete collaborative and argumentative environment, based on graphical direct manipulation and representation to facilitate the continuous exchange of information among domain experts. Looking at the actual state-of-the-art of ontology development tools, Concept Map COE is at the moment the tool that comes closer to the requirements specified.

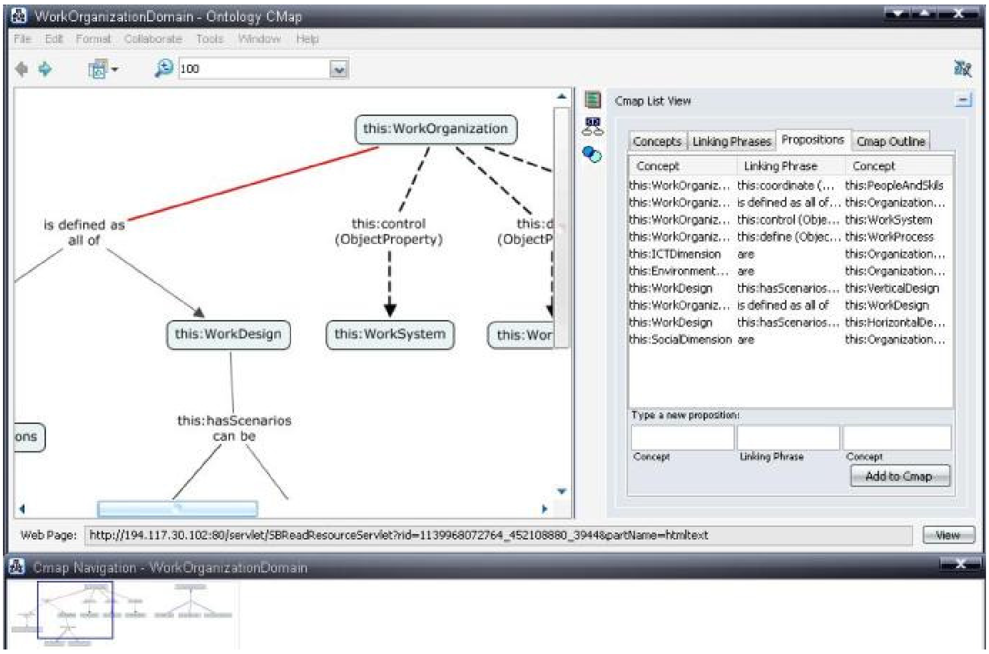
COE was developed in order to give *CmapTools* more interoperability and formalized notations meeting ontological agreements, that allows users to work with ontologies.

Cmap Tools COE can be used as:

- An ontology viewer;
- An ontology editor;
- A concept search engine.

COE uses concept maps to display, edit and compose OWL, in an integrated GUI combining Cmap display with concept search and cluster analysis. [12] COE can display any OWL or RDF ontology as a readable concept map, supports intuitive editing and construction of "ontology maps", allows uses to rapidly locate related concepts in published software ontology and outputs valid OWL/XML. The goal is to enable rapid and intuitive capture of machine interpretable knowledge by combining navigation, comprehension, selection and construction of knowledge in a single collaborative environment with an intuitive GUI. [12]

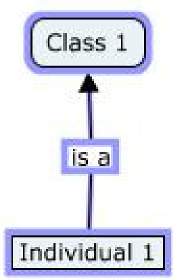
Figure 3-33 **CmapTools COE**

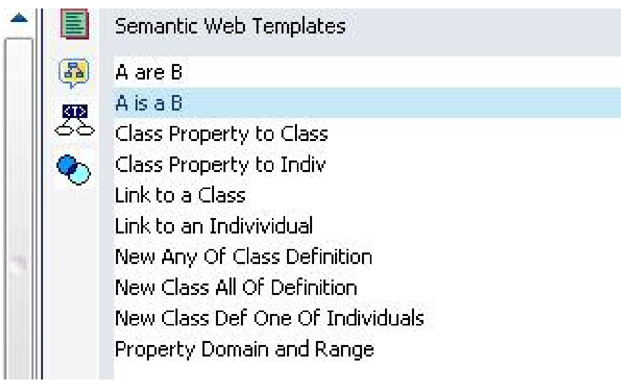


The majority of ontology-authoring tools need high level of technical skills to ontology development. In many cases it is necessary some OO (Object Oriented) sensibility. The friendly graphical interface from COE in opposition to text-based interfaces from other editors makes the mental processing of ontology, easier. The Protégé OWL, has already the capability of graphical visualization of the ontology but with a higher complexity of abstraction.

COE allows the capture of knowledge structures using templates which can be dropped directly onto ontology map canvas. The templates are graphical representations for commonly used OWL structures. With these templates the knowledge construction is much more rapid and easier.

If we want to say that "A is B" we can just drop the right template onto de ontology map as the next figure shows.





In OWL it will look like this:

```
<rdf:RDF
  <owl:Class rdf:about="http://localhost/this#Class 1"/>
  <rdf:Description rdf:about="http://localhost/this#Individual 1">
    <rdf:type rdf:resource="http://localhost/this#Class 1"/>
  </rdf:Description>
</rdf:RDF>
```

However when we want to export our ontology to OWL, is not always a peaceful process specially when we use templates for some constructors such as owl:intersectionOf (New Class All Of Definition template), owl:unionOf (New Any Of Class Definition template).

Concept Maps COE tool is not specific for knowledge representation languages. Its application is much more generic to cover new developments. COE is not only focused on the conceptualization and formalization of specific domain ontology, but on all network. COE is not a tool only for software specialists; it's a tool for all domain experts.

In fact, COE it's a very useful tool, but, regarding ontology building, Protégé is, so far, the most used tool. Indeed, Protégé is very powerful and capable software, however it lights much on a technical view and "disregard" a crucial aspect that is graphical direct manipulation of ontologies. In order to minimize this weakness, several plug-in have been emerging trying to endow Protégé for cognitive support on ontology development. All those efforts improved Protégé; however it is not possible yet to have a complete visual authoring feature, which means that Protégé do not have, so far, a direct manipulation feature over graphical representations.

3.3 Ontology development mediated by Cmaps: Practical experiences

Because Cmaps are, by their principles, well suitable for building domain ontologies, their use was leveraged in a quite few projects aiming at the creation of an environment allowing the construction and representation of domain ontologies. CmapTools COE tool, by its turn, has had a great acceptance as a privileged tool in a first stage on domain ontology development.

Cmaps are being used with success in teaching and learning domain as an intermediary for ontology creation but other areas in industry are also adopting this approach. One example of significant work done in this context is the CIMA project.

"The Common Instrument Middleware Architecture (CIMA) project, supported by the National Science Foundation Middleware Initiative, is aimed at "Grid enabling" instruments as real-time data sources to improve accessibility of instruments and to facilitate their integration into the Grid. CIMA middleware is based on current Grid implementation standards and accessible through platform independent standards such as the Open Grid Services Architecture (OGSA) and the Common Component Architecture (CCA)" [77].

The CIMA general interoperability goals are included in three main categories or levels:

- Hardware level
- Within a community
- Across communities

Within the last level (across communities) an ontology for interoperability of resources within a community of interest was developed. That core ontology, created for instruments and sensors consist of 74 concepts, 70 properties and individuals. Regarding the development process of the ontology, it was used Protégé and IHMC CmapTool instance editing and visual analysis of the ontology respectively.

Another significant example is given from the biological domain in a work conducted by Alexander Garcia. In this use case [32], is described the methodology used for ontology construction and how Cmaps were useful in all the process. In this particularly scenario [32], which involved domain experts distributed geographically, domain analysis and acquisition phases, were used different technologies that were not always integrated in protégé suite. In this scenario it was identified Cmaps as an important collaborative and knowledge acquisition tool for ontology development. In [32], was affirmed that "when developing an ontology involving geographically distributed domain experts, the domain analysis and knowledge acquisition phases may became a bottleneck due the difficulties in establishing formal means of representing and communicating knowledge". In order to develop the ontology for this specific case [32], it were used several protégé plug-ins such as OWLviz and Jambalaya until being decided the use of CmapTool in a first stage of the ontology development and the protégé in a second one. The choice of CmapTools instead of the others plug-is is due the direct manipulation capabilities of the graphical objects.

In fact, Cmaps were used in two stages. Initially it was used informal Cmaps, later, the Cmaps were semantic enriched (with CmapTool OWL – COE) and become more complex by formalizing the knowledge. According to [32], the Cmaps reveal to be an extremely useful tool for visualization and discussion, capable of representing basic knowledge elements in an easy way.

Cmaps' maturity was achieved with their use in the teaching and learning process. Nowadays and after Cmaps had been enriched when ontologies emerged, they have been even more used in the process of teaching-learning with their application with quite success in new concepts that have appeared such as **e-learning**.

According to [78], "A teacher now has to work as knowledge analyst or knowledge engineer making the skeleton of the studied discipline visible and showing the domain's conceptual structure. This structure is now called ontology."

Accordingly [78], an approach is presented for developing teaching strategies based on the principles of ontological engineering. "Object-Structured Approach is the method suggested to help the knowledge analyst (the teacher) to perform the most informal step of knowledge acquisition, concluding in prior conceptual structuring of the subject domain. This approach [78], highlights the need of a graphical editor to design the ontology. Taking this recommendation and add it to the fact that teaching-learning process is in the basis of conceptual mapping, the result of the operation could be "CmapTool COE".

I just presented three examples of different areas (industry, learning and a specific case in biology), however there are several other practical applications regarding the use of Cmaps in the ontology development process.

Even though CmapTool is to be an excellent tool in view of its scope, regarding ontology development, protégé is a much more powerful tool. The ideal would be that protégé could provide a true cognitive approach and a direct graphical manipulation in ontology construction. In the next section will be described some approaches in which the goal is to endow protégé with more easy to use and intuitive features for non-technical users.

3.4 Cognitive Approaches for Protégé

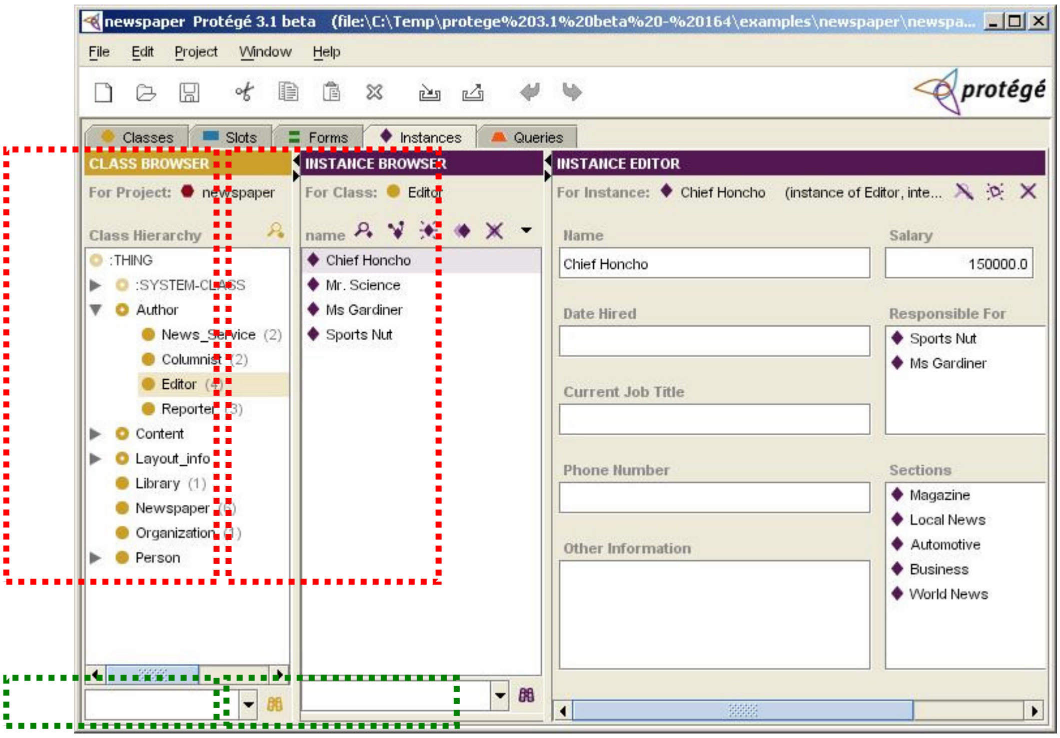
As mentioned before, Protégé is the most used tool for ontology building and it is in fact interesting to find out that new ways for guiding Protégé to be easily used among domain experts.

Having as background the knowledge engineering tasks presented in §3.1.2, we may conclude that Protégé offer some cognitive support for navigation, modeling and verification tasks provided by the protégé plug-ins identified in §2.3.4. Next it will be analyzed with some more detail how those tools (plug-ins) that grant some that support.

3.4.1 Protégé

Protégé itself has a tree navigation feature that allows users to navigate the class tree by opening and closing class concepts. Users can also use a search panel. As Figure 3.34 shows, Protégé provide Top-Down navigation and search visualization. However it's no sufficient and in this matter the plug-ins made the difference. Let's analyze some.

Figure 3-34 Class and Instance Tab with Browser



(Source: http://protege.stanford.edu/overview/pf-screenshots.html)

3.4.2 OWLViz

Figure 2.20 shows a screenshot of OWLviz.

OWLviz plug-in limit users to incremental navigation tasks and support browse multiple relationships at once. It does not reach the desirable level of cognitive support but increase it introducing graphical characteristics. The color scheme adopted by this plug-in is another important aspect in this area.

3.4.3 OntoSphere3D

Figure 2.22, 2.23 and 2.24 shows screenshots of OntoSphere3D.

Top-Down exploration is not the main strength of this tool, although it allows with rotation, panning and zooming several scenarios of incremental ontology navigation. The difference about OWLviz is the different navigation scenarios supplied by OntoSphere3D which is a step towards cognitive support in knowledge representation.

3.4.4 Jambalaya

Jambalaya is a tool developed by the University of Victoria's CHISEL software engineering group. Jambalaya is a suite of tools and views for viewing ontologies with graph metaphors;

Jambalaya provides several different mechanisms supporting navigation tasks, although browsing search results is not always simple. Like the other tools, Jambalaya has little or no support for editing tasks [9]. In the verification tasks, Jambalaya does fairly well; one can abstract information and search for arbitrary levels of interest, and identifying incoming relationships is easy as well [9]. There is no support for incremental navigation; there is rudimentary support for complex relationships: one can see which hierarchies a node belongs to, but there is no support for reification [9].

COE – Cmap based – is, in our opinion, the tool that more closely approaches to the ontology building requirements discussed earlier in this chapter. However, error checking and validating cmap-based ontologies still is a hard task to perform even resorting to the OWL templates provided by COE. Sometimes it is necessary to check the XML file – which is not pleasant to do – or to export the ontology to be open in Protégé to search for errors that at first sight seems do not exist when using COE with cmaps.

3.5 CM plug-in for Protégé

At some point during this work, we questioned ourselves about the possibility and interest of exploring cmaps capability within Protégé as some kind of plug-in providing Protégé with a nearly full cognitive support not only to construct ontologies more quickly and easily, but to input more responsibilities on domain experts and also to create a real collaborative development environment . During our researches we have found two projects that raised this question.

Alexander Garcia has mentioned in some papers the efforts that his research group has been making in order to an extension of Protégé – based on CM – that aims to provide an environment for ontology development within distributed, loosely-controlled, and evolving environments. "*Available tools do not promote collaboration, discussion, and exchange of ideas, all of which are critical components of any ontology development process in a geographically distributed community.*"[10] According to project goals "*it is necessary to provide cognitive support for the ontology development process in loosely centralized environments.*

[10] http://protege.cim3.net/cgi-bin/wiki.pl?ProjectsThatUseProtege#nid5L1

Conceptual maps should play a more central role in the ontology development process, not only during knowledge elicitation but also promoting interaction and access to the information"[1]

As far as we are aware, the project is in course. More specifically there are being made some usability tests of the plug-in i.e., the group is working together with domain experts in order to evaluate the plug-in. According to the information provided some aspects have been considered for the selection of our own domain experts: 1) they are geographically distributed, 2) they have to work with several initially unrelated but orthogonal ontologies, 3) they participate throughout the entire development of the plug-in 4) they constantly evaluate the development.

Despite of the lack of results it is quite relevant and interesting to know that soon we can make use of the two must used tools in ontology development not caring about the task of translating Cmaps to OWL structures in order to be able of incorporating them into Protégé.

Next our own experiences on using cmaps for ontology construction are described.

Chapter 4

Collaboration and Sharing Ontology Development in CODEwork@vo project

This chapter discusses the experiences obtained in CODEWORK project through the usage of CmapTools software both in the work organization ontology development and the way it was created in terms of concepts and knowledge representation.

4.1 CODEwork@VO project

CODEwork@vo is an acronym of **CO**llaboration and **DE**mocratic **work** design in **V**irtual **O**rganizations.

CODEwork project aims to answer the following research questions: "how to integrate business process design with task design in virtual organizations?" and "which management opportunities will bring this integration?”.

The main objective of CODEwork@vo is to create a framework for the design of work in virtual organizations (VO), integrating distributed business processes design with work task design. Furthermore, this framework will be supported by a computer application that will enable the management of the web of relationships resulting from the design process.

Amongst the several deliverables produced by the CODEwork, a model for mediation in virtual organization work design was developed. The guidelines for the achievement of the deliverable propose was highly based in the development of ontologies since they are the best forms to describe interrelationships between concepts describing both behavioral characteristics and operation characteristics of work. Furthermore, ontologies are an enabler for interoperability and collaboration. A work organization design ontology was developed in this particularly case.

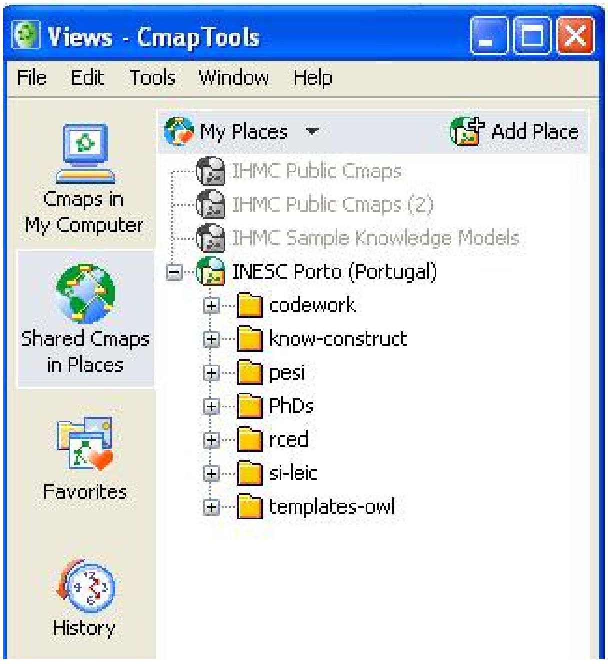
In the specific domain of work organization and in the research about the new forms of work organization, sociologists and managers and the domain experts. CODEwork project joined sociologists and engineers in an attempt to find out a work organization model considering both technical and social perspective.

A visual approach was need due the heterogeneity of backgrounds of the experts in domain conceptualization and ontology development. That visual approach is closer to the human understanding. At the same time, mechanisms of communication, discussion and argumentation were desirable which leaded to the selection of CmapTools COE to support the creation of the ontology in the work organization domain.

4.1.1 CmapTools COE software at CODEwork@VO project

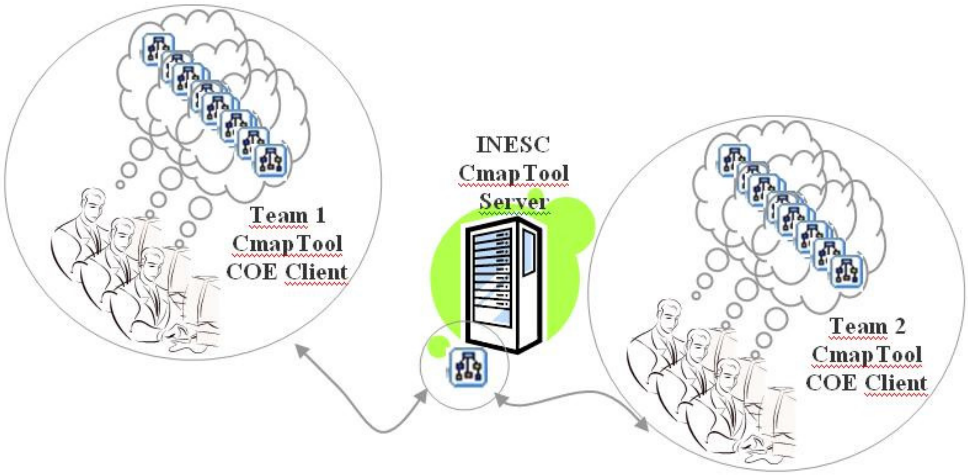
COE has a client-server technology. In a computer at INESC Porto was installed the server version which lodged ontology maps from several projects. The projects are represented by a folder with specific permissions parameters as figure 4.35 shows. Each element of the teams of each project had the client version installed. Locally, team members had its own working maps saved in the local hard drive, and at the same time, team members putted the maps in the server for appreciation and discussion. Figure 4.36 shows the basic network architecture of the COE.

Figure 4-35 Folders in the Server (Place)



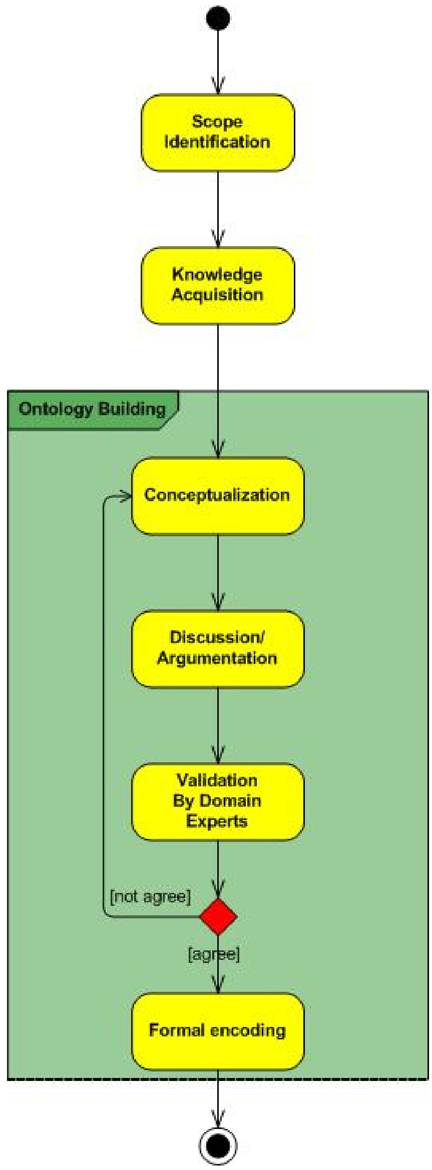
The CmapTools client program is used to construct and manipulate cmaps. Cmaps can be stored in the local hard drive or on a place (server). However, to share a knowledge model with users in other computers, the client program needs to communicate trough a computer network with a server running CmapTool server software [16].

Figure 4-36 Basic CmapTool network architecture



At the CODEwork project, the ontology development in the work organization domain had several steps; however we did not follow any methodology in particular. The next figure shows the process that we perform for creating the ontology.

Figure 4-37 CODEwork ontology development methodology



After an initial analysis and the definition of the ontology scope, it was submitted to approval by the research groups and eventually approved. Knowledge acquisition was the next phase, where all terms about the domain within the scope range were collected. Explicit definitions and descriptions were made and the competency questions were formalized in order to see if our ontology fulfilled the main scope. Later, Cmaps were created freely, without any formal restriction. The created Cmaps represented only a particular view of the problem. The next steps were to put the map on the server for collective discussion and validation. The team argued about the conceptual map trough notes, forum or other maps, files, links…, until it is reached a common understanding. After the cmap had been validated, it was converted in a formal notation. This formal notation was achieved by using COE templates.

Figure 4-38 CmapTools New Discussion Thread

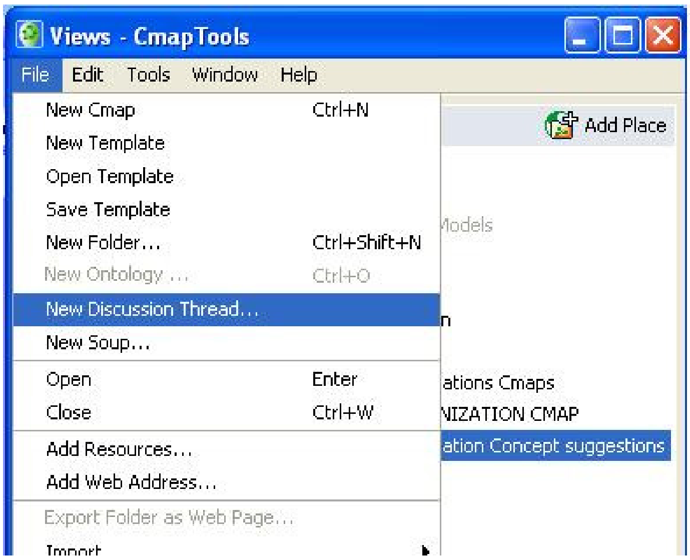
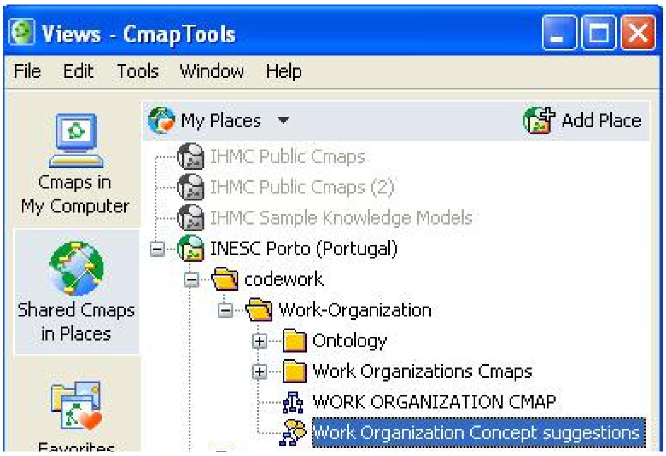


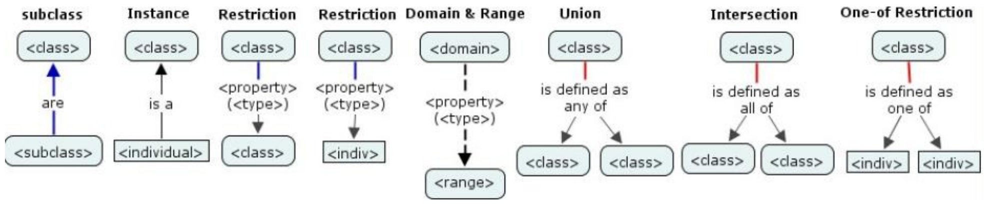
Figure 4-39 CmapTools Concept Suggestions



Templates for creating ontologies

COE is built on top of IHMC Concept Map Tools tool suite, so (almost) anything it can done do in normal concept mapping it can also be done in COE: adding nodes, moving things around, dragging and clicking, navigating, etc... Creating Ontologies in CmapTools COE is almost the same as drawing a concept map using CmapTools interface, but in order to understand the imported OWL ontologies, and produce Cmaps which will be output as OWL, users will need to learn, and use, some new conventions. For several of the OWL constructs - form a normal Cmap - it is necessary a number of steps to make the appropriate linkages and set of the box and line styles according with to specific convention [4]. To ease this, COE provides templates for commonly used structures such as union, intersection and restriction [4]. The use of templates in ontology creation is faster, easier and efficient, however to be possible the use of such templates, which are nothing more than basic pre-defined maps that can be drown into a Cmap, it is necessary to download them into the client computer and saved locally. The next figure is the graphical representation of the OWL Templates provided by COE.

Figure 4-40 OWL-specific language templates



Source: (Hayes et al., 2003)

Templates are dropped directly onto the ontology map canvas, where they appear as extensions to the concept map containing the bracketed 'reminder' phrases. The "subpropertyof" and "inverseof" properties of properties are constructed directly by dragging a link line between property labels [13], [29], [30].

Ontology creation in COE compel to learn by heart some conventions in the way a set of concepts and its relations are graphical represented. This learning process became easy by the use of templates, which technical knowledge of owl is low. Nevertheless the first contact with these conventions for ontology development may cause confusion to users less familiarized with ontology creation and OWL. Due to this fact, in the creation of knowledge models using (normal) Cmaps, after being validated, can be converted to the appropriated conventions.

Another validation phase is needed after all this process, but this time it requires a more technical discussion lighting on the aspects of the ontology itself such as the design principles. In the first validation phase the domain experts have been validated the concepts, relationships, scope and definitions. At this point ontology engineers should evaluate if:

- The domain was covered by the ontology.
- The propose of the ontology was achieved.
- The ontology provide information for the answers of the questions that leverage the ontology development
- The ontology can be (re)used and maintained

At this point should be also analyzed

- The classes and the class hierarchy.
- Rules slot Domain and Ranges

Once again templates are a good help for ontology engineers evaluate an Ontology Cmap.

The experience with COE in the CODEwork Project has demonstrated how easy knowledge elicitation and sharing could be through multidisciplinary teams with different backgrounds of knowledge and know-how and how a work model can effectively be achieved with tools having cognitive support. The use of Cmaps allow the project teams to achieve a knowledge model in a simple way favouring the know-how and experience of individuals. With the approved model COE allows easily to construct the corresponding OWL structures.

Regarding a collaborative work environment in ontology development there are some essential features that still missing in CmapTool COE which become important disadvantages. CmapTool doesn't provide version control as well as tracking changes, so careful is needed when changing the ontology. Our advice is to take the permissions features of CmapTool COE very serious and configure it in a responsible way. Only one person (ontology administrator) should accept the suggested concepts and make all the changes annotated just after a complete acceptance of the proposed changes by the domain experts. Each element of the team should keep their maps both locally and on the server.

4.2 Work Organization Ontology

Work organization is an up-to-date theme and all the turbulence resultant from de modernization of work pull the enterprises, social partners, governments, national organizations and worldwide organizations to act. In 1999 was created de European Work Organization Network (EWON) following the European Commission's presentation of green paper- "Partnership for a new organization of wok" in 1997 and the communication "Modernizing organization of work – A positive approach to change" in 1998. In order to standardize concepts, several papers produced by EWON reporting case studies, good practices and the new trends in work organization in Europe, as well as other institutes and organizations around the world, provided inputs for us to develop a generic ontology in work organization domain. It is public that the global economics instability, the new and constant demands from consumers, competitive pressures and the fast development of the technology, force enterprises to restructure through take-overs, new relationships with partners, intensive rationalization and development into service components among other things. We live in a "knowledge-based economy as it is necessary to develop and diffuse knowledge about and the use of new forms of work organization"[62] According to the green paper the drives for new forms of work organization are human resources, markets and technology and this is true in all work organization literature. The green paper also concluded that flexible work organization is a prerequisite for improving productivity and thus economic profitability. So, according with [61] the issue is not only about how work is organized but also how job is designed in terms of scheduling, remuneration, employment, job and task variety, skills, workplace, working time, management and organization. If we add the fact that the work organization is no longer enclosure within an organization and it often crosses the boundaries of organizations, professions, and countries will increase the complexity of the work organization scenarios. Sinha and Van de Ven (2005) [35] in Organization Science Journal made an approach to this complexity of conceptualizing work. The work organization is a domain of global interest, which due its knowledge-based characteristic there is a set of concepts and terms associated with objects of change within and between organizations. The relations within and between organizations are not static, they assume an acceptable range of values in a specific domain (work organization). The organization of work is a complex restructuring domain that must be firm in a formal process. For all this characteristics is legitimate to think about an ontology that could present in a simple way the action fields in the work organization domain and group the main formal concepts and it's correlation without the constraints of pre-defined reorganization models. Based on the main drives regarding the organization of work presented before, we collect a set of terms among some sources related with human resources and his work relations, in order to perform the conceptualization of the domain. In this context, I took the role of knowledge engineer, working both as a facilitator in the ontology construction by the involved research groups, and acquiring knowledge from the domain experts and use it to formalize the ontology and then apply it in an content management system. However, our approach followed the concerns pointed in [92] aiming at reduce or even withdraw the interference of the knowledge engineer in the ontology maintenance process and provide mechanisms that enable the domain experts to maintain the system and adjust it to changing knowledge requirements.

During the process of Knowledge acquisition, several were the consulted sources aiming to collect the most trusted information in work organization domain. It was collected information at the most important European Union (EU) organisms and some projects that they support. At the same time a research was made in order to identify "the best practices" beyond the European Union recommendations.

At the EU level and beyond the sources already mentioned, it was consulted some important recommendations and conferences [28], [42], [44] in the competency domain of the "European Agency for safety and health at work" (http://osha.europa.eu) and the "European Work Organization Network" of the European Foundation for Improvement of Living and Working Conditions. In this EU investigation was given a particular attention the project WORKS (Work Organization and Restructuring in the Knowledge Sociality) – www.worksproject.be - and conferences such as ICT, the Knowledge Society and Changes in Work Conference among others

Outside EU boundaries several institutes, governmental entities and non-profit organizations were consulted in order to clarify all the terms and concepts with the objective to find common descriptions and definitions. U.S. National Institute for Occupational safety and health (NIOSH - www.edc.gov/niosh) has special interests in Work Organization domain. Some data was collected to be engaged in a common ontology.

International labour organization (ILO) a UN agency, responsible for working conditions standardization and for promoting equal work was another important resource in the construction of the work organization ontology.

Human Resource Internet Guide (www.hr-guide.com), is an interesting web resource containing hundreds of pages with information related to human resources. Some concepts were found here very well defined and related.

All the collected information, terms and concepts from all these sources were combined and its similarities identified in order to achieve a standard vocabulary in the work organization domain.

4.3 Ontology Goal

The main objective of the development of the ontology is to help people to better understand work organization design and the new forms of work organization in order to better refine organization orientation strategies. This ontology will help the employees of the organization that have some decision making role regarding organizational design. The ontology will be also a standardization factor for a common understanding about work, avoiding communication gaps among individuals. This common understanding became even more useful once the work is performed beyond organizational boundaries. This ontology regards the consensus not only for employees of an organization but also for all interested people. Making use of other people know-how based on documented experiences on several articles and books we achieve an ontology for a common understanding on work organization design. Finally it is important to mention just one more single aspect; since the first efforts concerning to new forms of work organization, several terms referring to the same concept have been emerging. As time goes by this may cause some confusion even among specialists and an ontology is useful to clarify this situation.

According to some requirements on the process of ontology building, some guiding steps were adopted. The following steps were given:

- Specification of competency questions – questions that the ontology is able to answer.
- Ontology capture – concept and relations identification as well as the respective restrictions. At this point Cmaps were used
- Construction of dictionary of terms.
- Formalization of the ontology – for ontology formalization it was used CmapTools COE, which gives us an environment for ontology construction.

4.4 Ontology Scope

The domain and the intended use of the ontology were already described; however it is necessary to build the ontology boundaries defining the main subject areas that the ontology must cover.

According to the specified domain (work organization) it is necessary to understand some specific concepts related with the process of work design and organizational design in a human resource management perspective within and between organizations. Regarding the

European Union recommendations it is also important to be aware of the issues related with the social and physical environment around the worker. Finally, what is the ICT (Information and Communication Technology) support necessary for the work organization pool of concepts?

This work organization ontology aims to be generic without focus any particular task; so the detail level is not necessary to be very low; the intention was to capture the main concepts.

The main vehicle for the identification of the scope subject areas were the competency questions (CQs). CQs, which should be defined by domain experts, are informal queries specifying requirements that are in the form of questions and which must be "answered" by the ontology. The CQs may be later defined using a formal language but the in this specific case that wasn't done, although the CQs be a very useful and viable mechanism for ontologism evaluation.

In our approach, we didn't specify all the competency questions at once. The ontology was guessed extensive; hence, the first competency questions defined helped to create the subject areas that the ontology should cover. The main competency questions are shown in table 4.4:

Table 4-4 Work Organization Ontology Competency Questions

	Competency Question
1.	What are the things which define Work Organization?
2.	What are the relations between Work Organization, Work design, Organization Design and Job Design?
3.	What are the Work Design Scenarios?
4.	List all the work control procedures which are based on trust?
5.	What are the structural dimensions of organization design?
6.	What are the structural determinants of organization design?
7.	What are the design models of organization design?
8.	Are Human Resource Management design and Ergonomics the things which define the social dimension of work organization?
9.	What are the things which define ICT in an organizational context?
10.	What setbacks should I consider in each work design scenario?
11.	What modules should I consider in work system decomposition (Modularity)?
12.	How many design network approaches are?
13.	List all network design approaches?
14.	Should I perform all available job design approaches when designing a job?
15.	How should we implement task significance?

16.	How should we implement task autonomy?
17.	How should we implement task variety?
18.	How should we implement task identity?
19.	How can we use Job analysis information?
20.	Is job performance part of job analysis or part of job design?
21.	Is job specification part of job analysis or part of job design?
22.	Is job description part of job analysis or part of job design?
23.	What is the horizontal design main focus?
24.	What is the vertical design main focus?
25.	What are the things which define ergonomics?
26.	Is ergonomics of correction ergonomics?
27.	Is ergonomics of conception ergonomics?
28.	What approaches should we consider when regarding ergonomics?
29.	List all the things which the approach is based on cross-functional teams?

Judging from this list of questions, the ontology will focus on information about various **approaches** to job, work and organizational design and how some of that approaches **can be implemented**. The ontology will provide information about what work organization concepts need ICT for their effectively implementation and when trust should be considered.

4.5 Ontology High Level Description

Work organization is not only about the way jobs are designed, but also about the way they are performed and about who does it and in what conditions.

Work organization can be defined as a multilevel concept, divided in 3 levels of context: External Context, Organizational Context, and Work Context [43]. We cannot do much about external context. The most immediate thing we can do is to predict what will happen next and act according to it. The main management challenge is to organize/optimize all work and resources of an organization in order to endow enterprises the ability to easily adapt to market changes like a chameleon can adapt to his habitat, or like some kind of "plug and play" capability.

The figure 4.41 presented below, shows the conceptualization of work organization in a concept map. That cmap emphasize the following assumptions:

1. Work Organization can be defined as the result of work design process or organizational design process implementations.
2. Work organization has a direct impact over people and their skills, the processes and the overall work system[11], in view of the fact that it refers to the (re)organization of those concepts.
3. Regarding work design, we can consider two main different scenarios according to the boundaries of work system, thus we can have a work design process based in a vertical division of work, which the components of work are distributed inside organization among one or more hierarchical levels. By the other hand we have the case in which the components of work are distributed across many organizational units of one or many firms and, in these case, we have a work design process based in a scenario of vertical division of work. [34]
4. Organizational design has a set of structural dimensional, structural determinants and design models that drive the implementation process in an organization.
    a. The structural dimensions are [60]:
        i. Formalization level - The level to which expectations regarding the means and ends of work are specified, written, and enforced.
        ii. Complexity - The number of distinctly different job titles, or occupational groupings, and the number of distinctly different units, or departments.
        iii. Centralization - The location of decision-making authority in the hierarchy of the organization.
    b. Structural determinants are [60]:
        i. ICT – Information and communication technology in the organizational context
        ii. Size - This refers to capacity, number of personnel, outputs (customers, sales) and resources (capital).

[11] Work system is composed be 5 elements: individuals, tasks, tools and technologies.

iii. Environment – Is a very complex determinant. It comprises the relations and dependencies with other organizations, communities, customers, governments, etc. that are in continuous change and to which an organization has to adapt.

iv. Strategy – Comprises the vision, mission and competitive advantage strategies of an organization. Organizational strategy is concerned with envisioning a future for business, creating value, and building and rich a strong position in the marketplace.

c. Design models are [60]:

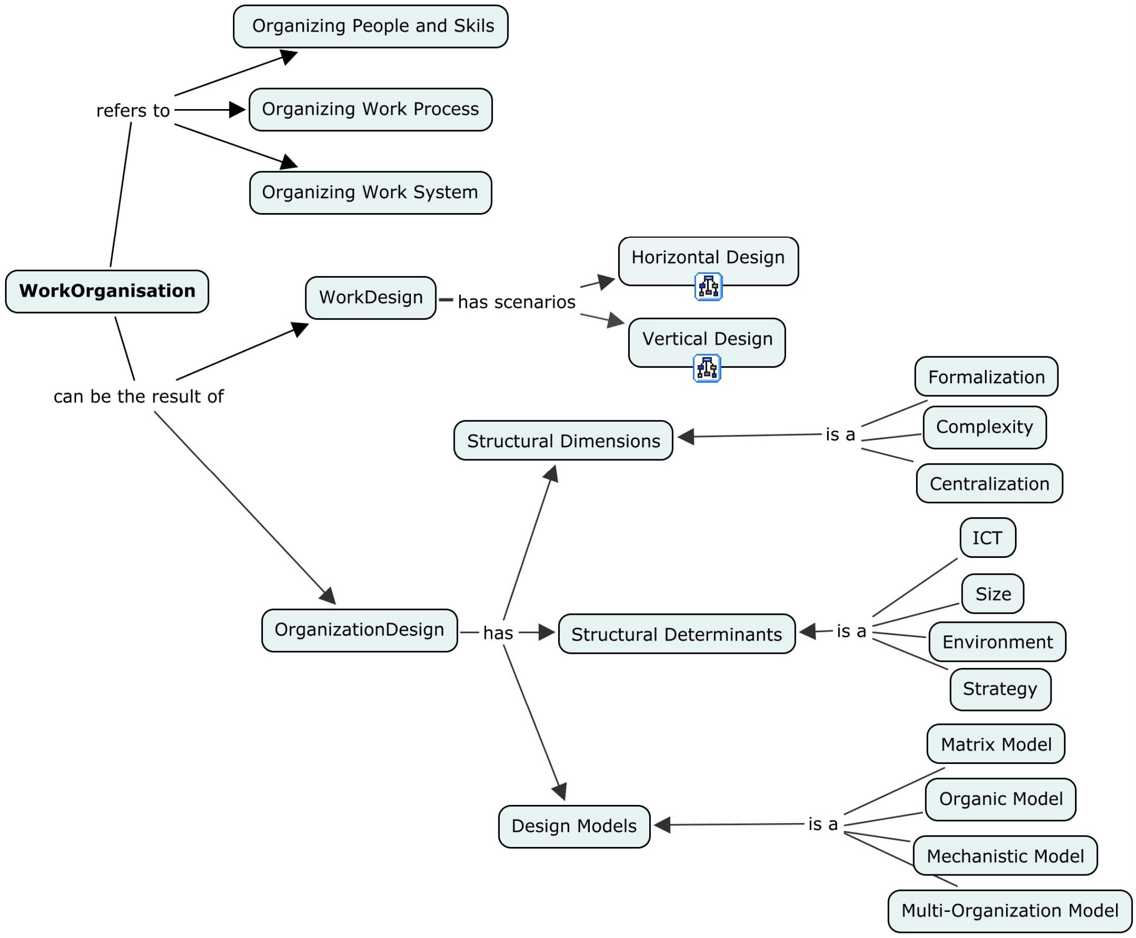
i. Matrix model - Organizations should combine functional and product departmental bases to maximize the strengths and minimize the weaknesses of each.

ii. Organic Model - Organizations should seek designs that maximize satisfaction, flexibility, and development.

iii. Mechanistic model - Organizations should function in a machinelike way to accomplish the organization's goals in a highly efficient manner.

iv. Multi-organization model - Attempts to minimize and, in some cases, eliminate vertical and horizontal structures, tightly defined work roles, and top-down control.

Figure 4-41 – High Level Work Organization Cmap



The figure represents a high level conceptual map part displaying the main work concepts relations.

Being work organization a wide concept we restrict the ontology domain to the following range concepts: work system, work process and people and skills. These are the central concepts according to the most recent proposals regarding new forms of work organization and the arguments mentioned in the section before.

Work design brings new and complex problems in both design scenarios. According with Sinha and Van de Ven (2005) [35], **Modularity**, **Hierarchical decomposition** and **Network** are the new concepts defused by the new trends of work organizations.

Figure 4-42 Work Organization - Horizontal Work Design Cmap

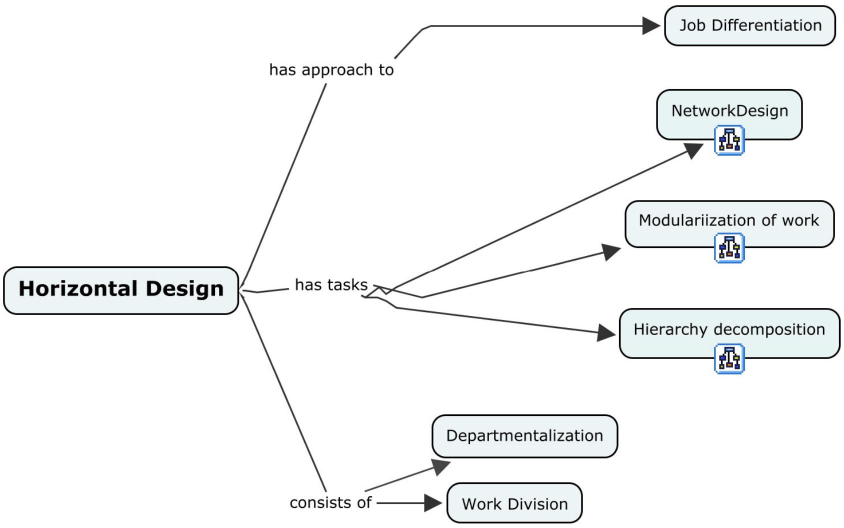
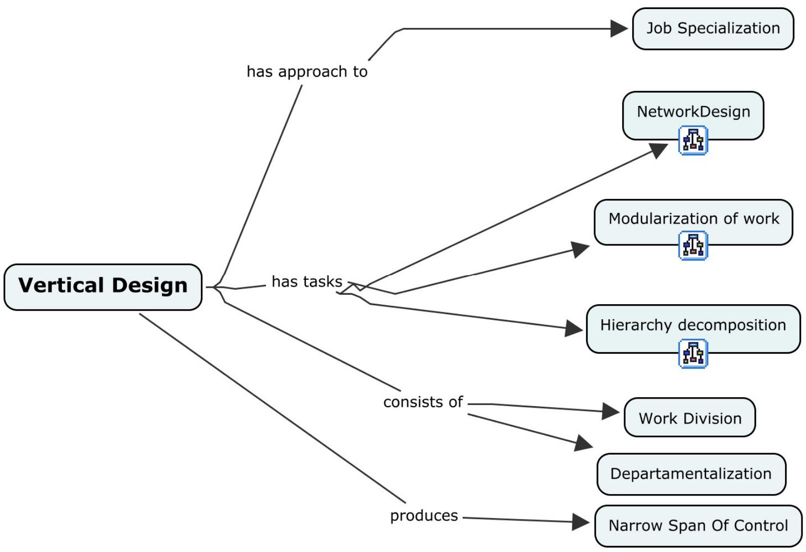
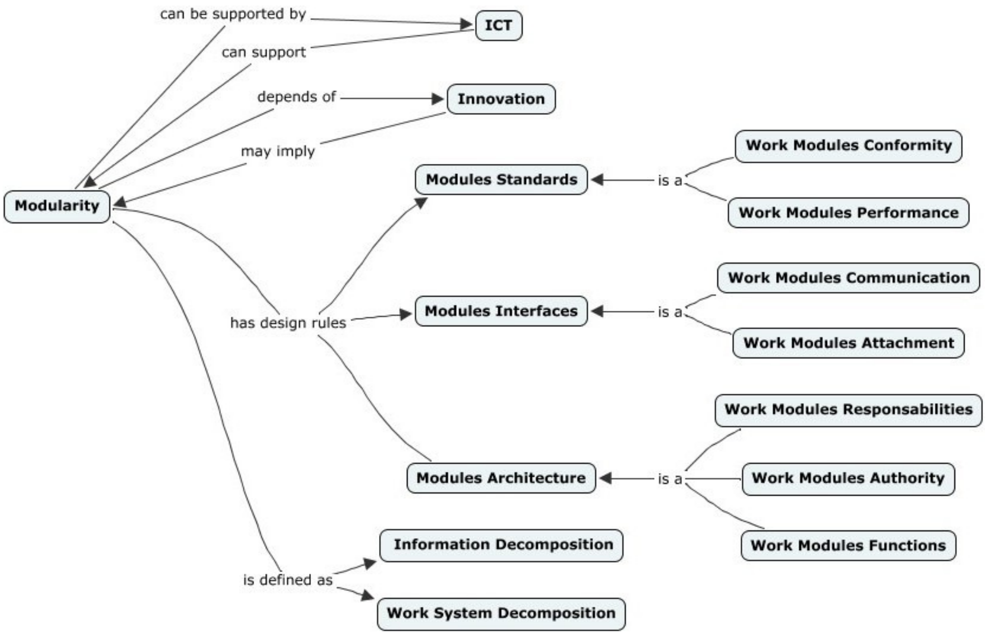


Figure 4-43 Work Organization - Vertical Work Design Cmap



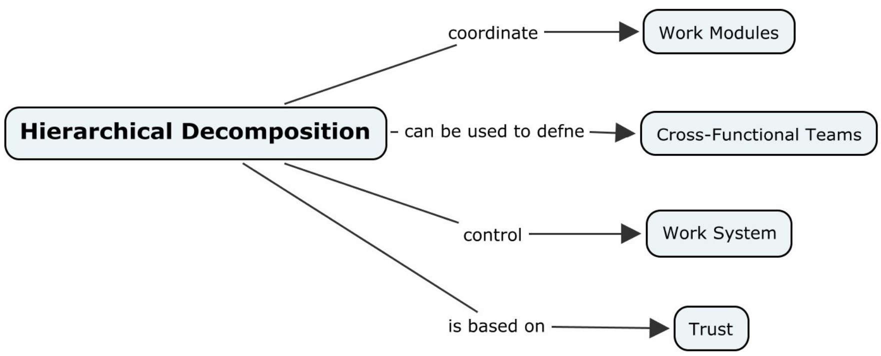
The main issue of **Modularity** is to decompose the work system into work modules reducing work system dependencies and increasing flexibility. Each work module has its own responsibilities and inside affairs. The goal is to have work modules with "plug and play" capacity among units and organizations.

Figure 4-44 Modularity Conceptual Map



Hierarchical decomposition is about coordination and authority among the levels of one or more organizations.

Figure 4.45 Work Organization - Hierarchical Decomposition Cmap

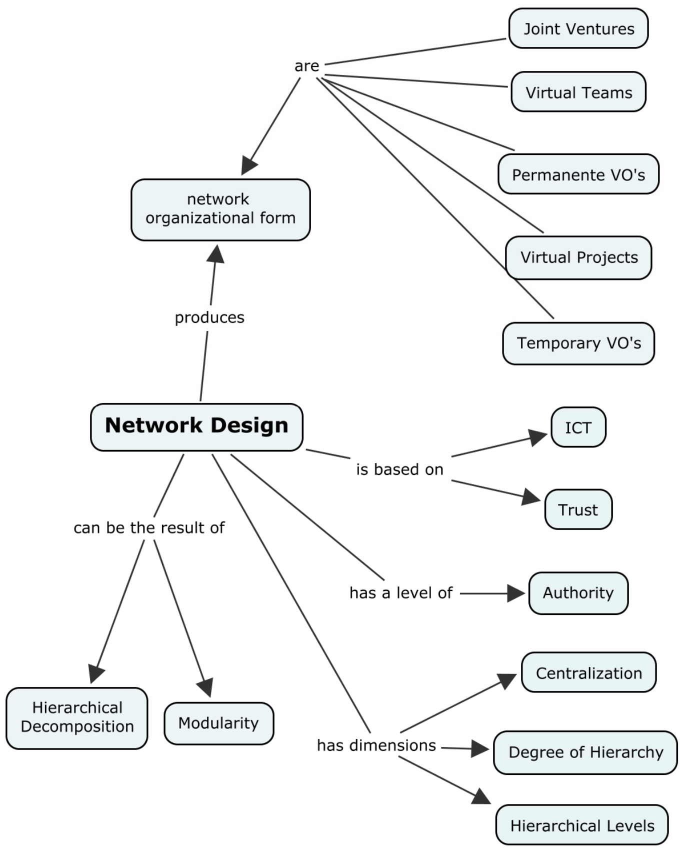


Network is a little more complex and has various approaches namely Virtual Teams, Virtual Projects, Temporary or permanent VO's. However it's necessary three fundamental requisites which are: trust, authority and Technology. [48],[49],[50].

According to Bernard R. Katzy [46], trust and ICT (Information and Communication Technology) constitute two of the three facilitators for development and implementation of VO (Virtual Organization). Market mechanisms is another facilitator concerning to the propositions on design and implementation of virtual organizations. Thus I highlight two propositions: a)

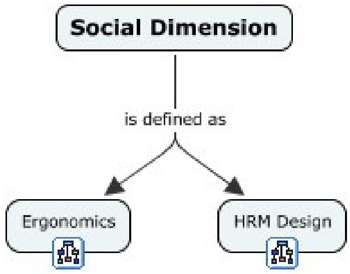
Recognized authority facilitates the rapid development and implementation of virtual organizations and b) Development and implementations of virtual organizations is limited to the level of trust among the co-operating partners *[46]*.

Figure 4-46 Work Organization - Network Design Cmap



In the three work organization dimensions, the social one has been increasing its importance in all the issues related with the way work is performed. Human Resources, Ergonomic, Job Design and Job Evaluation are terms with a broad social sense and they not only specify the tasks, duties, goals, requirements and responsibilities for workers but also the Environment, Job compensation, Job Enlargement, Job Enrichment , Job Rotation , Job Effort factors and ergonomic evaluation of jobs.

Figure 4-47 Work Organization - Social Dimension Cmap



With the development of the anthropocentric perspective of production systems and the increase of human factor importance, during the 80's the first efforts individual and group ergonomics appeared.

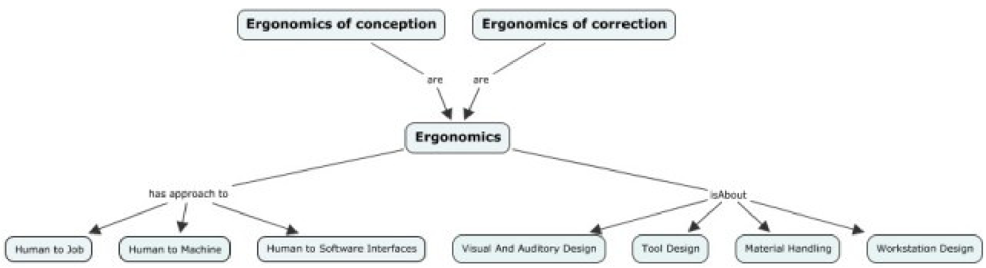
Ergonomics has nowadays a particular importance in the work organization domain.

Ergonomics is directly connected with the work health and assurance giving answers by anticipating possible problems related with workers. Ergonomics provides mechanisms for a constant adjustment of workers needs in their workstation.

The study of how the design of equipment affects how well people can do their work is performed in three perspectives:

- The worker and his job.
- The worker and the machinery.
- The worker and the software Interfaces.

Figure 4-48 Work Organization - Ergonomics Cmap



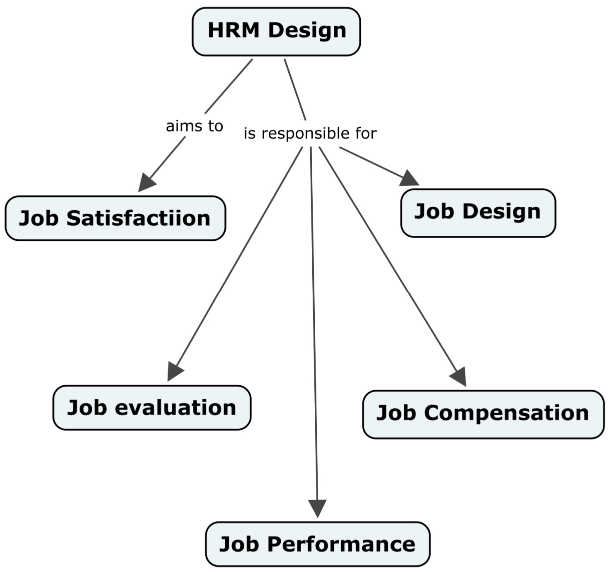
Organizations are very important and influent elements in society; they are responsible for helping maintaining and improve well-being of society as a whole. Employees are one of the must important pieces in the set of Organization's Social Responsibilities; they can't no longer being seen as costs, but as valuables assets. It is in this domain that Human Resource Management has the main responsibilities such as:

- Assistance in diagnosing organizational problems.

- Assistance in planning and implementation of job (re)design.
- Modification (if necessary) job descriptions, job specifications, and eventually adjusts pay rates of some positions.

Job design, job evaluation and compensation are important roles in all of these activities. For better knowledge of employees and their needs it is necessary to plan , design and evaluate their functions, responsibilities and benefits in the Organization.

Figure 4-49 Work Organization - Human Resource Management Design Cmap



The big issue about work is productivity.

Productivity is directly related with human behaviour, thus is conditioned by several factors, some are environmental issues, some are related with the employees and others are related with innate impulses. However all our behaviours are determined by the predisposition to reach an objective, thus motivation is the number one factor of human behaviour.

Motivation is in the base of several theories about human behaviour as a mechanism to satisfy successively our needs, more serious discussed since Abraham Maslow proposed, in 1943, the Theory of Human Motivation. Since then, several theories arrived.

By definition motivation *"has to do with a set of independent/dependent variables relationship that explain the direction, amplitude, and persistence of an individual's behaviour, holding constant the effects of aptitude, skill, and understanding of the tasks, and the constraints operating in the environment" [55]* .

According with several authors [50],[54] and according with the Hackman and Oldham's dimensions of motivational potential, job satisfaction is the core of Job design and work organization improvements, motivational approaches such as task identity, task variety, feedback, and autonomy and task significance are the main potential factors to achieve satisfaction at work. However there are other important key considerations to job satisfaction such as environmental issues, ergonomic issues, and organizational issues as described below.

As mentioned before, Job Design is one of the most important responsibilities/challenge of human resource management. Job Design tries to balance technological and organizational requirements with personal economic and/or social needs of the workers. Job Design use information of job analysis in order to perform the alignment of jobs and its contents to fill in different needs such as:

- Organizational needs related with:
- Efficiency and workflow.
- Ergonomic needs related with:
- Workplace safety.
- Physical demands, rest breaks, seating, climate...
- Efficiency and productivity.
- Employees needs related with:
- Variety and autonomy.
- Training and task significance.
- Feedback

In this sphere of action there are some design approaches to put into practice, namely:

- Job Rotation.
- Job Enrichment
- Job Enlargement
- Job Engineering

Figure 4-50 Work Organization - Job Design Cmap

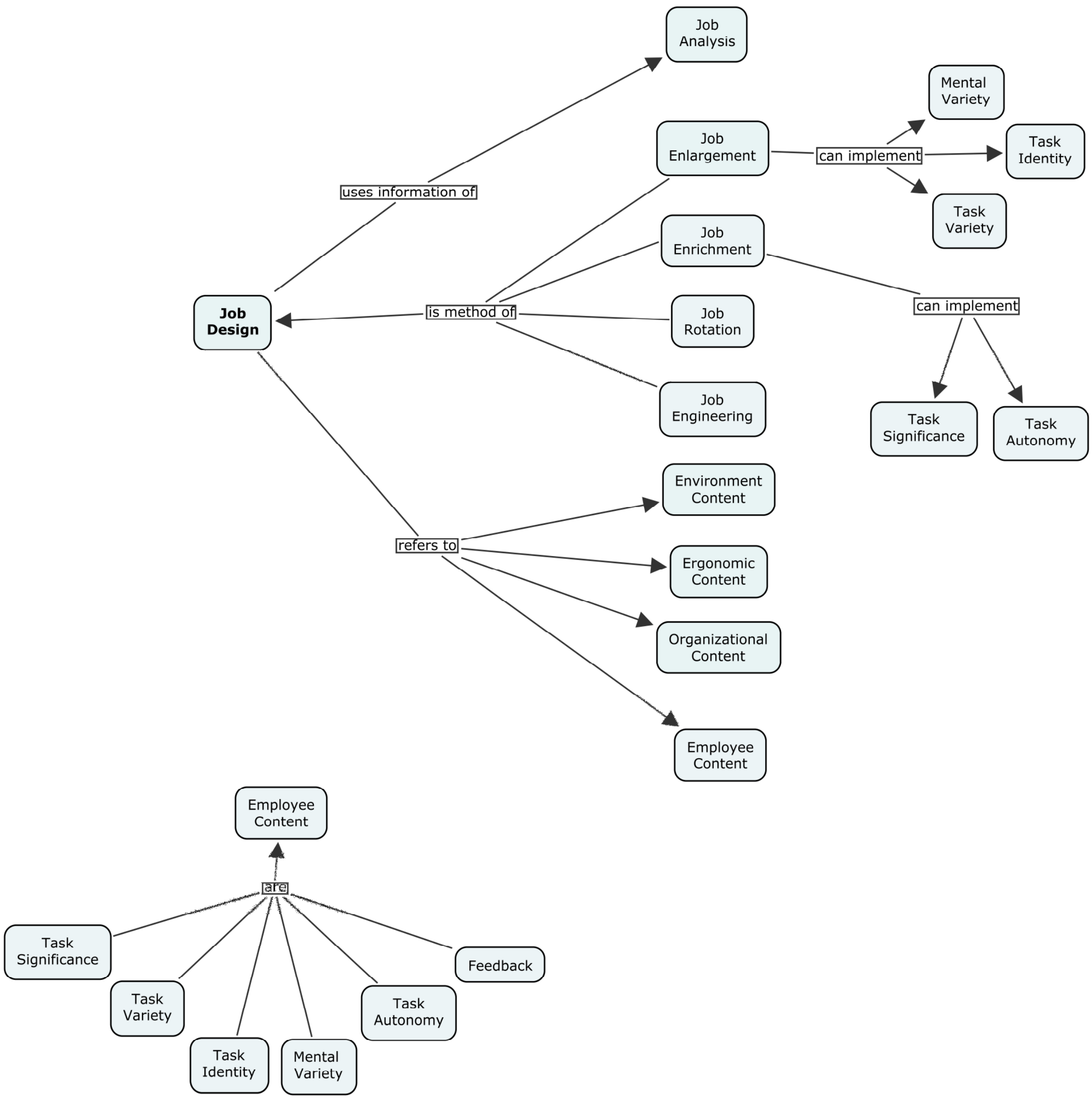
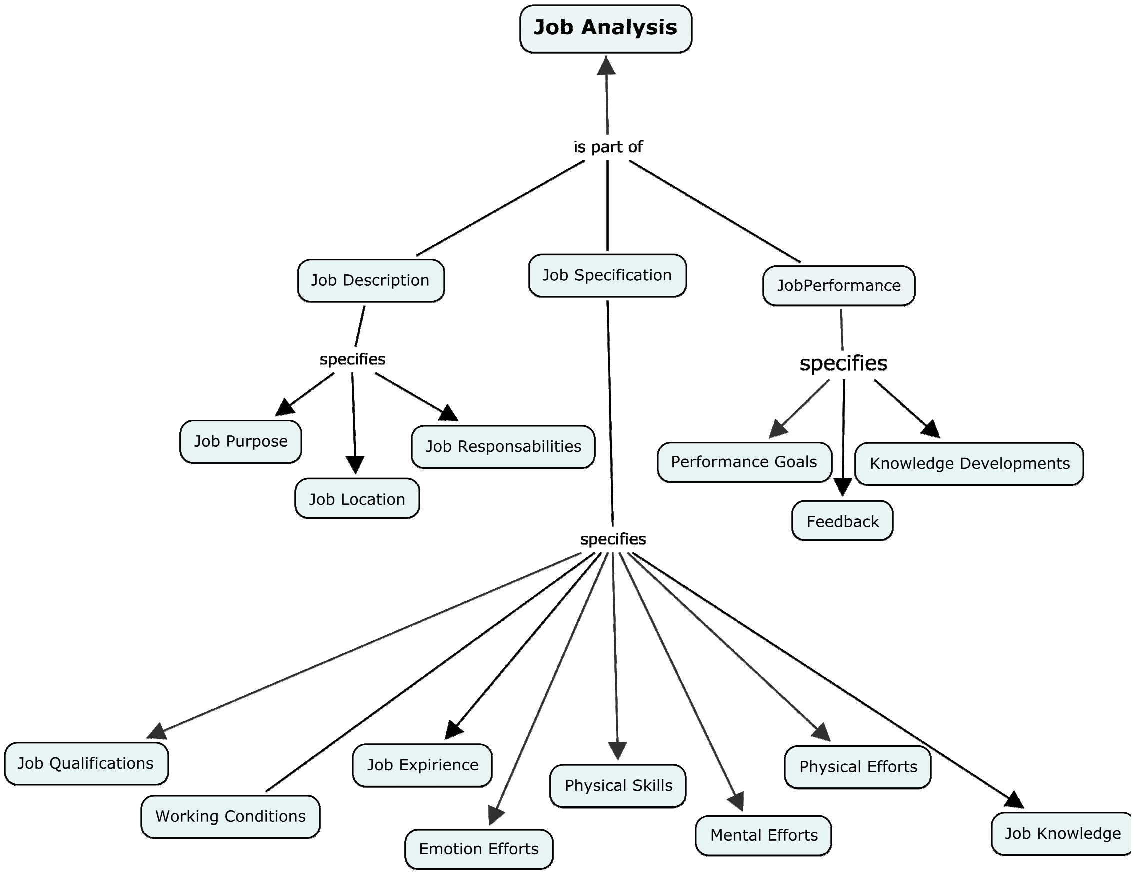


Figure 4-51 Work Organization - Job Analysis Ontological Cmap



As said before ICT is a facilitator for the development and implementation of virtual organization development increasing de power and range of development and implementation in the virtual organization and favouring market mechanisms.

Due its nature virtual organization is continually restructuring as a response to the opportunities in the market place creating innovative solutions. ICT have in this context a prevailing paper, because not only facilitates the search for new partners as well as extends the range of managerial design within and between organizations having a huge impact in almost every aspects of work.

Technological advances can no longer be treated as separately from other organization dimensions. However we are relearning it. The technological influence in work design is very important in the way the job is performed. The impact of ICT in work and workers should be evaluated according with the job design main goals.

Information technology refers to the computer mediated work where a task is accomplished trough the medium of the information system rather than trough direct physical contact with the task *[52]*.

An organization to have in one's possession the maximum of technology doesn't mean more efficiency, more flexibility and more celerity in business process execution.

It's a fact that technology has a significant strategic importance in organizational context, however technological restructuring should be aligned with the strategic goals of the organization and with the (re)organization of work.

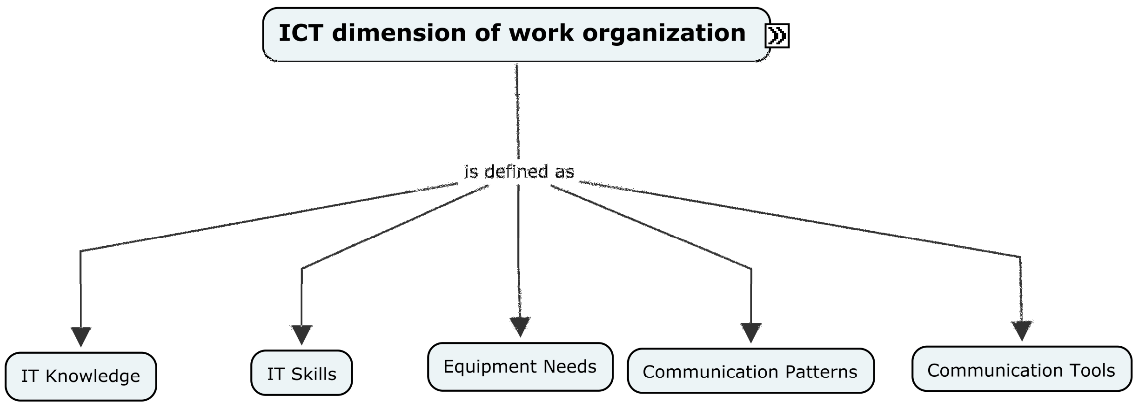
In the development and implementation of new information and communication systems it is necessary the capture of several requirements:

The technological knowledge needs to the development and implementation of extensible systems easy to maintenance and upgrade.

- Human resources IT Skills to operate with ICT.
- The equipment needs.
- Communication Tools
- Communication patterns, in order to allow an easy integration with other systems.

We must keep in mind that all this requirements innate to development and implementation of new information and communication systems should be adjusted with job design.

Figure 4-52 ICT dimension of work organization Cmap



It was presented the conceptualization of the work organization domain through concept maps, together with a brief theoretical explanation. The following images match to the translation of the conceptualization of the domain. These maps are concerning the graphical representation of the OWL specification of the domain, developed with the CmapTool COE and its templates.

Figure 4-53 – High Level Work Organization Ontology Cmap

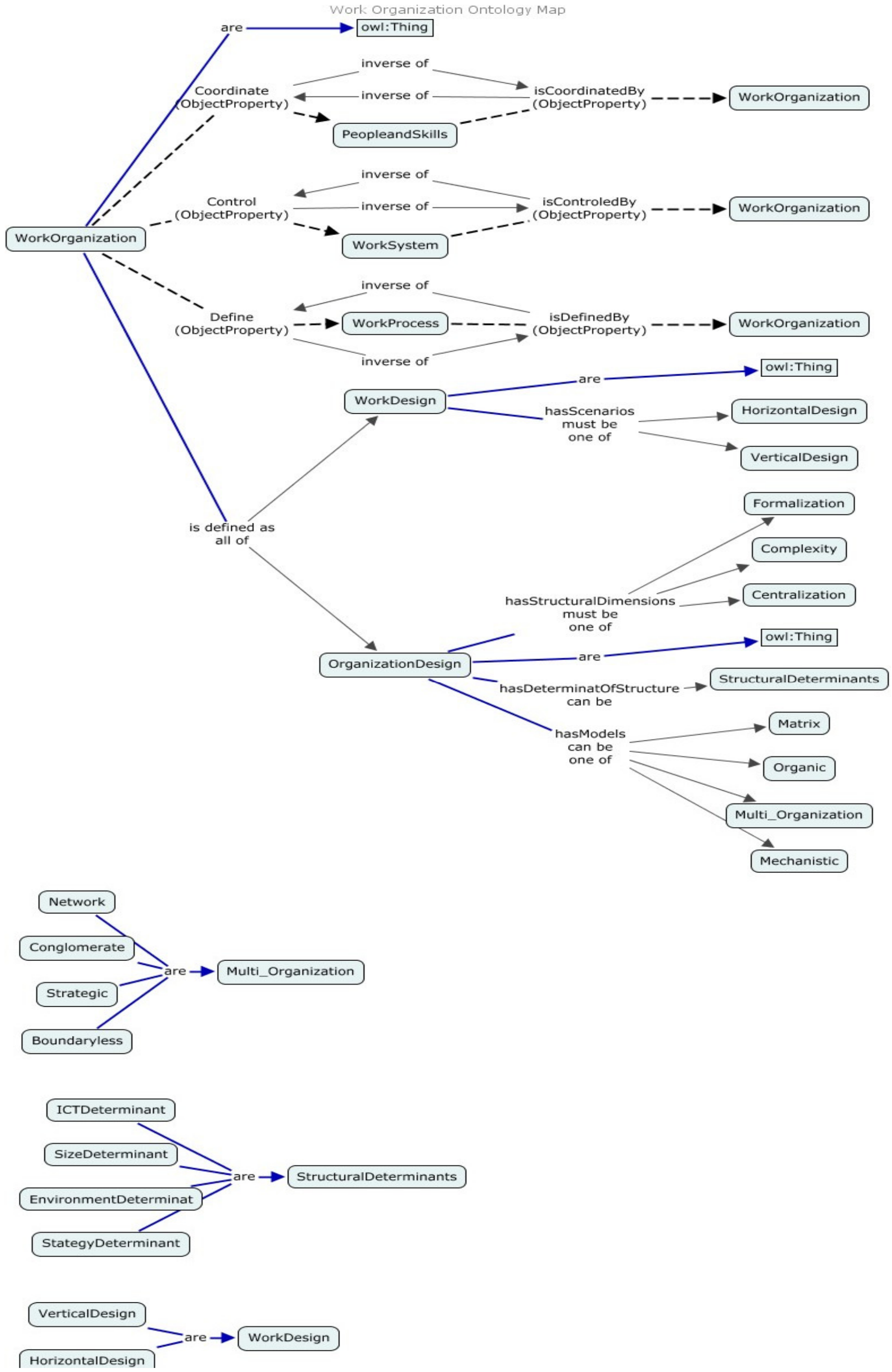


Figure 4-54 Work Organization - Horizontal Work Design Ontological Map

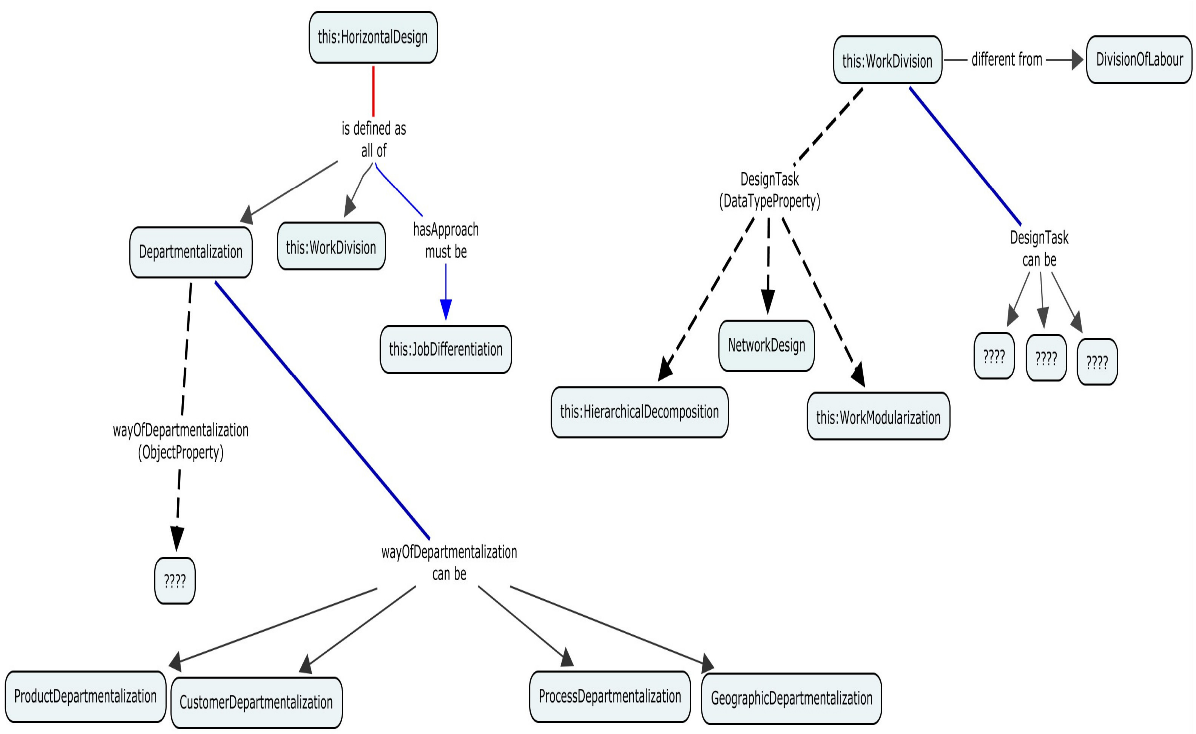


Figure 4-55 Work Organization – Vertical Work Design Ontological Cmap

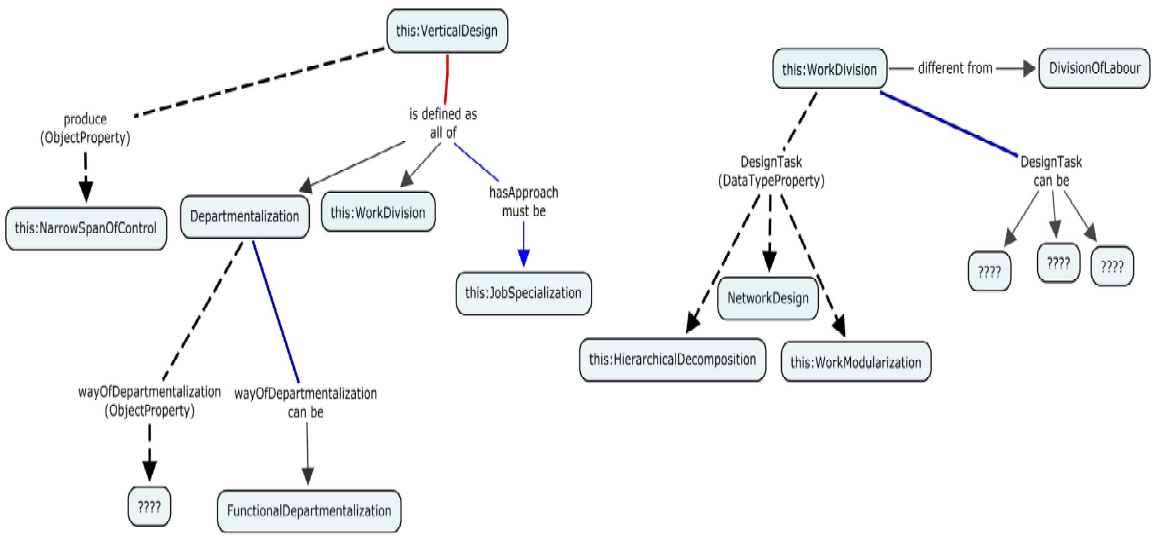


Figure 4-56 Work Organization - Modularity Ontological Cmap

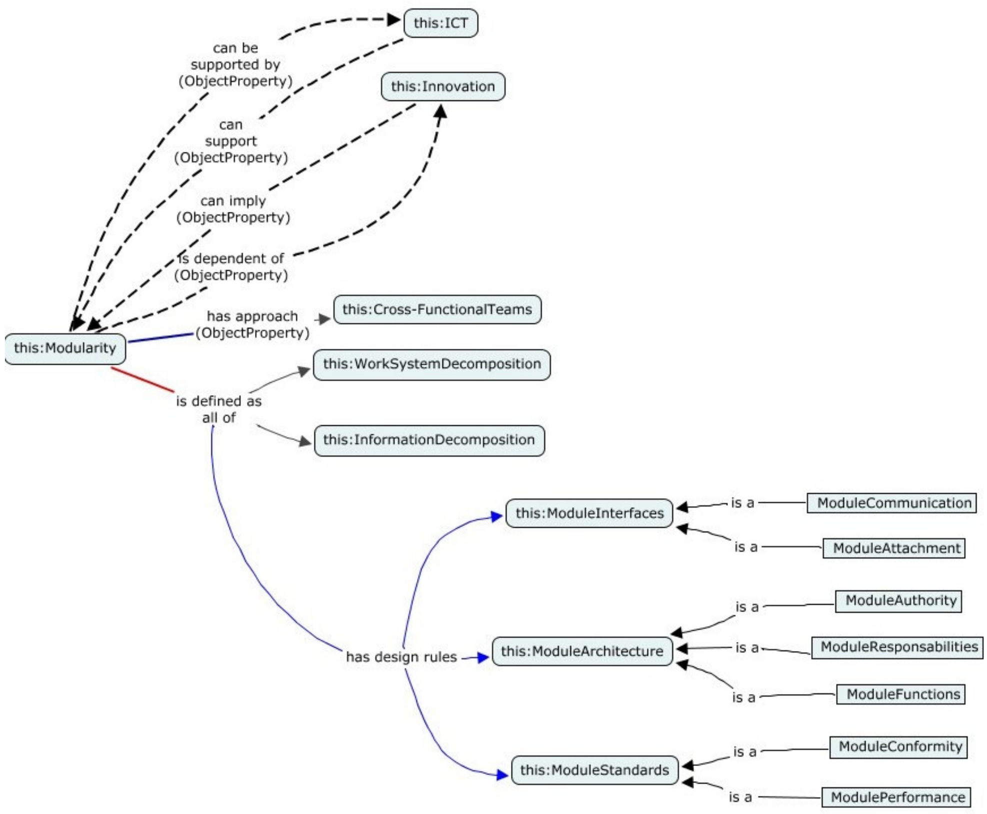


Figure 4-57 Work Organization - Hierarchical Decomposition Ontological Cmap

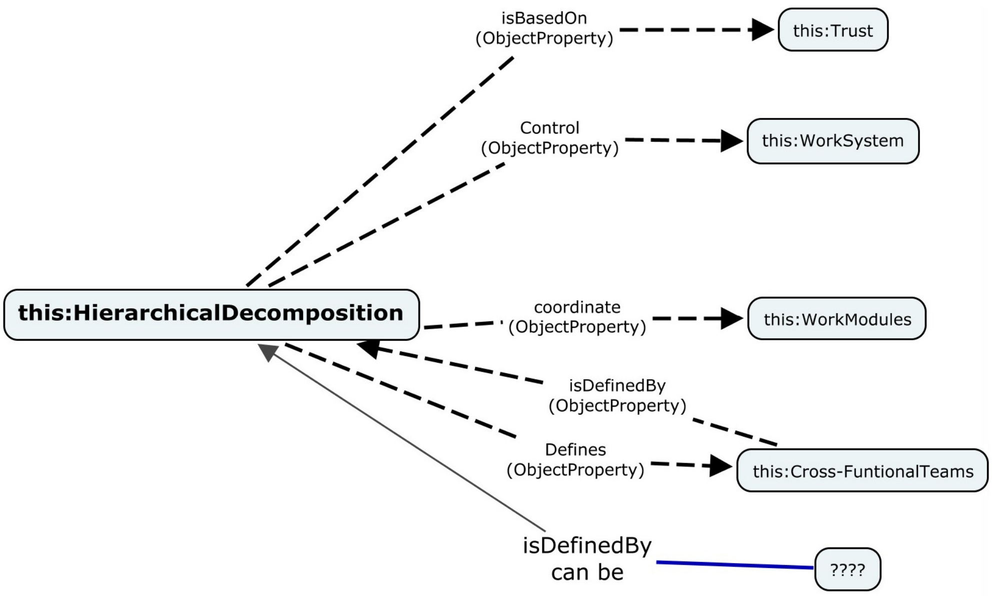


Figure 4-58 Work Organization - Network Design Ontological Cmap

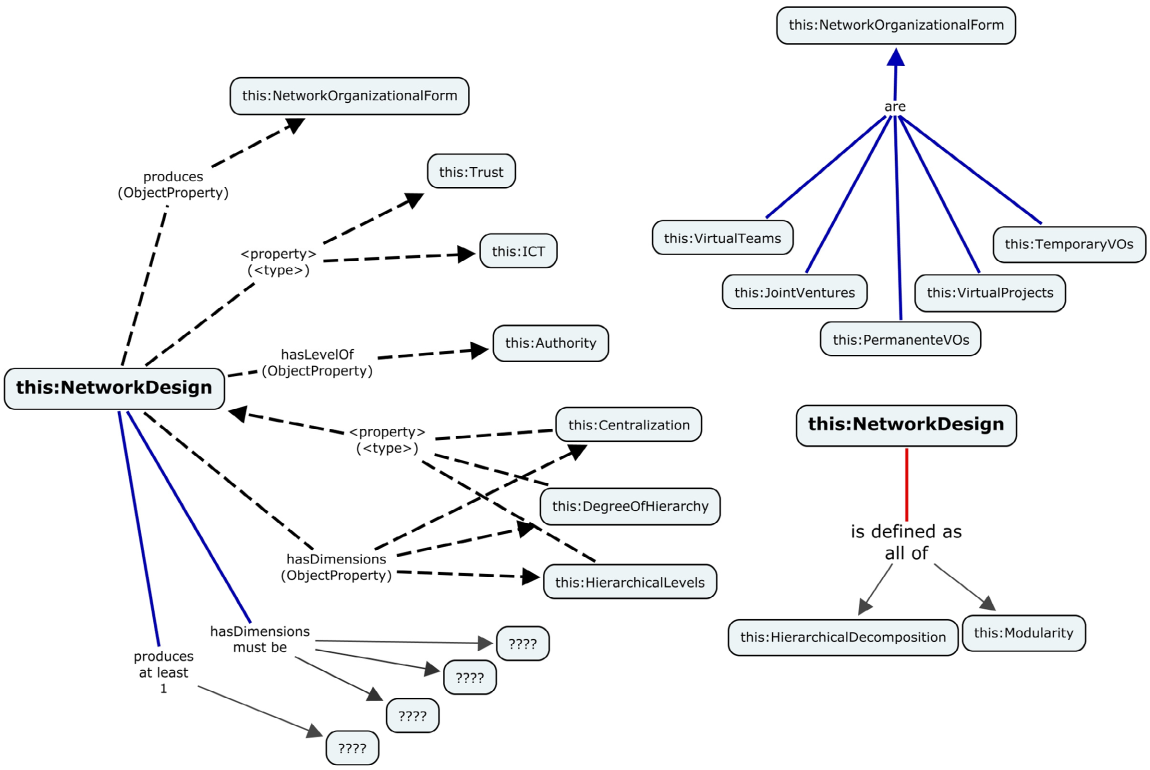


Figure 4-59 Work Organization - Social Dimension Ontological Cmap

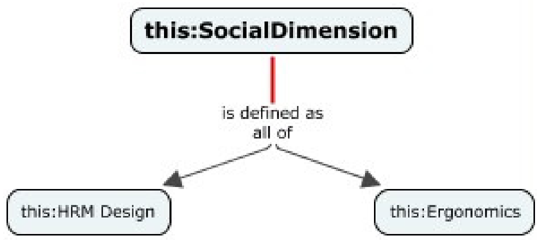


Figure 4-60 Work Organization - Ergonomics Ontological Cmap

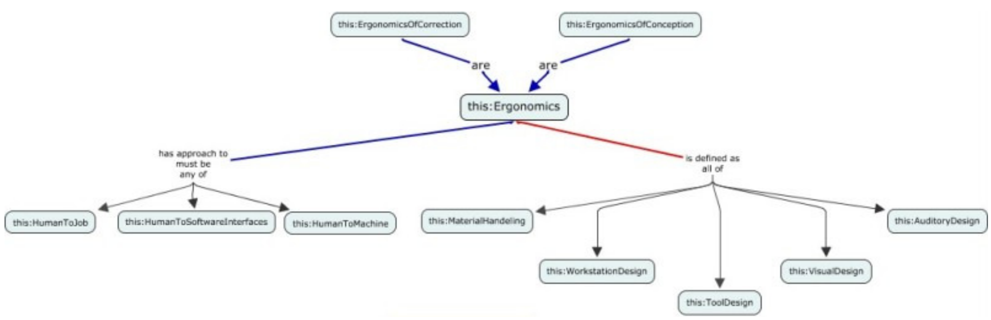


Figure 4-61 Work Organization - Human Resource Management Design Ontological Cmap

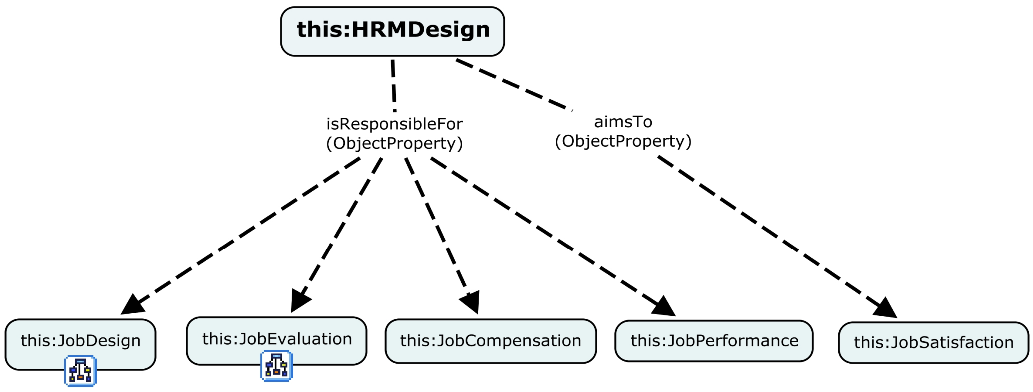


Figure 4-62 Work Organization - Job Design Ontological Cmap

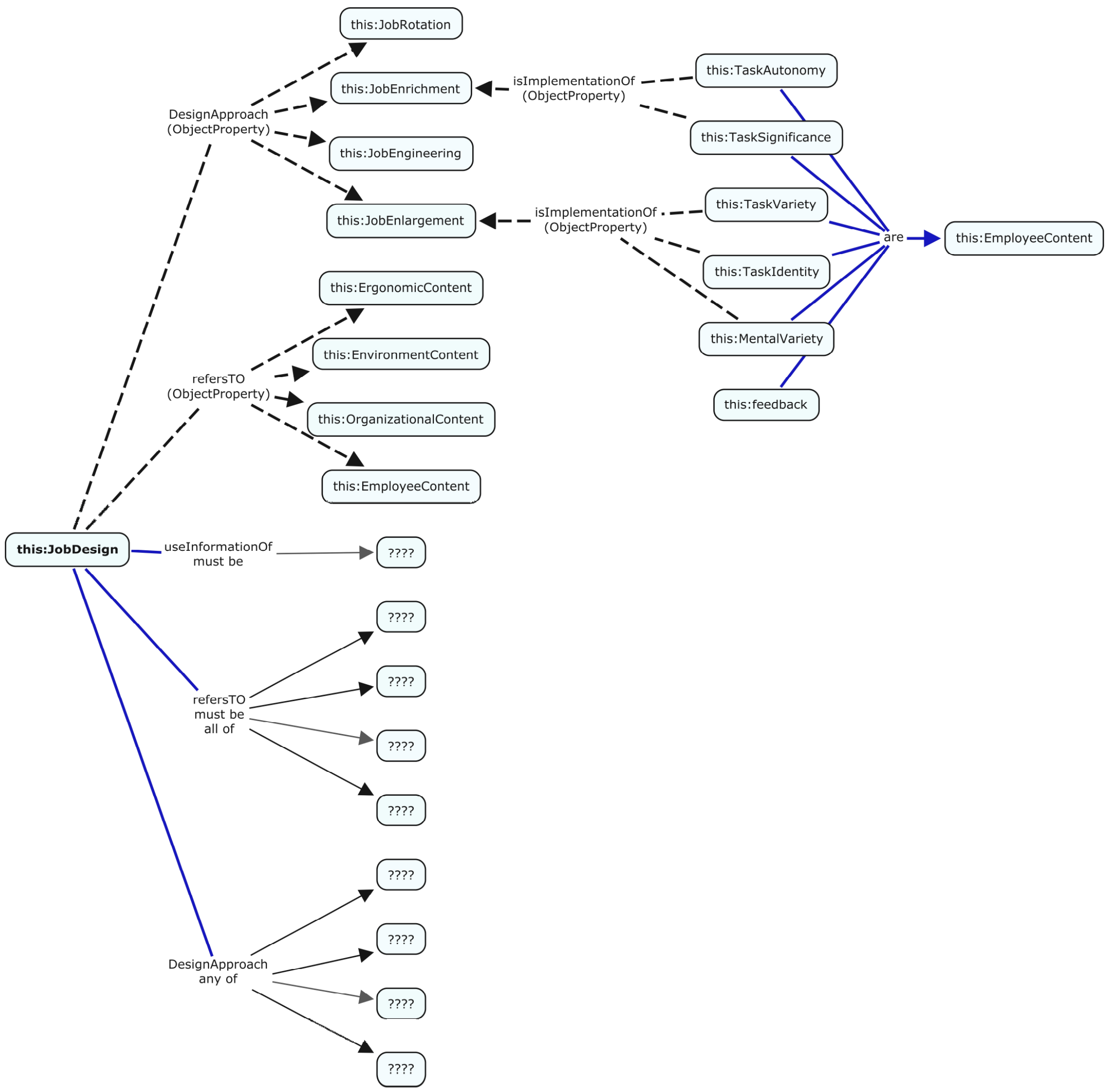


Figure 4-63 Work Organization - Job Analysis Ontological Cmap

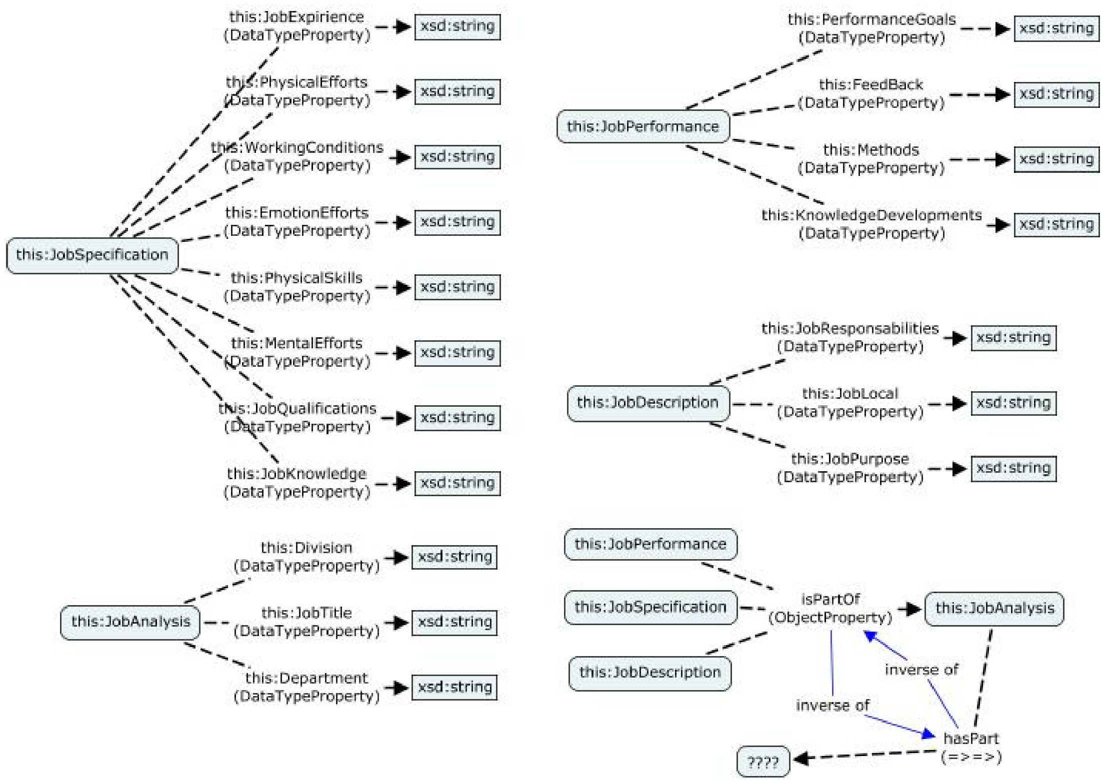
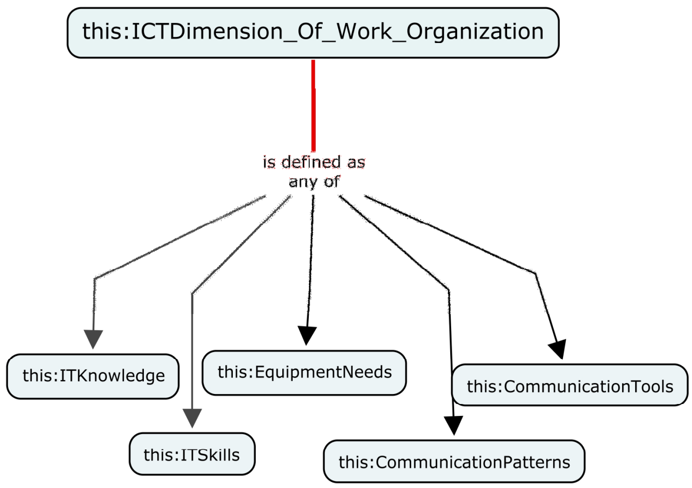


Figure 4-64 ICT dimension of work organization Ontological Cmap



An high level ontology on work organization domain was presented. The construction of this ontology in a collaborative environment allowed reaching a reasonably consensual knowledge model. The next step would be the use of the ontology in an organizational environment, e.g., within a knowledge management system.

The model presented here was built under a formal notation according with COE tool convention. For that reason it is possible to export this model into OWL language, which can be processed semi-automatically.

For the effective use of the developed ontology in a knowledge management system, it was necessary to achieve an architecture model allowing the easy translation of the ontology to the system and its maintenance without the interference of a knowledge engineer.

The next step is to find a way for doing that, with the minimum software development efforts.

Chapter 5

Using Ontologies to manage Plone content.

Technically, information management can be performed through the implementation of content management systems (CMS). CMS are the practical translation of Knowledge Management and Information Management approaches. CMS have been used for a long time just as content storage systems with a huge amount of non-related data. The concept of knowledge management joined to information management concept opened new horizons for CMS. Content is not knowledge for itself, knowledge is in the processes and practices that content management encircle.

James Robertson [85], a practitioner in the knowledge and content management area, has an interesting perspective how a CMS can be used to achieve the knowledge management goals. Robertson enumerates several organizational requirements or tasks for a CMS implementation, which can be performed appealing to knowledge management strategies, in order to meet knowledge management goals.

Restructuring and rewriting information.

Analyzing and reworking information helps to clarify knowledge needs in the organization.

Subject-matter experts or domain experts

Knowledge professionals should work as a team. Working as a team is a trick to get the knowledge that is in the head of the workers and that otherwise would be very difficult to get.

Content owner's identification

Content Owners must be identified for all information. This will provide a knowledge map of the organization which improves knowledge sharing. People inside organization will know exactly what to do and who they should ask for information.

Metadata

The use of metadata is the CMS' main asset.

Meta data is useful is a variety numbers of ways:

a. Tack content owners.

b. Capture relationships between information

c. This includes using approaches such as ontologies.

d. Capturing classification such as keywords.

e. Tracking the range of business.

Workflow

Workflow increases the manageability and Enterprise Content Management (ECM), and improves information quality control. The big advantage of workflows is the need of deeply know how the organization operates before setup workflow rules.

Barriers reduction

Here the issue is to have easy-to-use interface and easy ways for adding, sharing and evaluate new content.

Feedback

Collecting workers feedback is actually knowledge acquisition. However, that will not have the desired effect if the previous point (barriers reduction) is ignored.

Metrics

Metrics is about stats, search engines, etc. It must be provided easy mechanisms to access information efficiently. Tracking the words introduced in a search engine provided by the CMS it is for sure useful ways for knowledge capture.

Navigation

Search engines, cross-references, shortcuts, etc., can be provided.

It goes a long way between the knowledge capture (through content creation and organization) and its classification within the CMS. At the same time people with the content manager role not always participate in the process of knowledge capture, which means that not always the result of content organization is optimized. It became necessary to shorten the distance between conceptualization and implementation which demands for translation mechanisms between concepts and technical models or languages.

In this particular scenario and according to the architecture presented ahead, UML was used as the main vehicle in order to achieve a complete organization of the Plone content. With the use of ArchGenXML (see section 5.3.5), UML can be used to generate code for Plone Products based on ArcheTypes framework (see section 5.3.2). ArchGenXML parses the UML models in XMI format (e.g. .xmi, .zargo, .zuml), created with tools such as ArgoUML.

"However, UML is based on object oriented paradigm, and has some limitation regarding ontology development. These limitations can be overcome using UML's extensions (i.e. UML profiles), as well as other OMG's standards (i.e. Model Driven Architecture– MDA*).*

Currently, there is an initiative (i.e. RFP) within the OMG aiming to define a suitable language for modelling Semantic Web ontology languages in the context of the MDA." [79].

The architecture presented in this context scenario is not quite perfect, but it is an approach to model oriented software engineering. Beyond UML, other brand new standards are being developed in order to allow this kind of interoperability. In the next two sections we present why UML can be used for ontology development and the new MDA paradigm developed to satisfy new demands on systems interoperability.

5.1 Ontology Development and UML

Using UML as an ontology modelling language was firstly advocated by Stephen Cranefield [80], according with his approach UML object diagrams can be interpreted as declarative interpretations of knowledge, class diagrams provide a static modelling capability that is well suited for representing ontologies and if Semantic Web application is being constructed using object-oriented technology, it may be advantageous to use the same paradigm for modelling ontologies and knowledge. In particular, there are several characteristics of UML that, according with Cranefield [80], can be seen as advantages:

- UML has a very large and rapidly expanding user community;
- Unlike description logic formalisms, there is a standard graphical representation for models expressed in UML and XML-based formats;
- The Object Constraint Language (OCL) is powerful and allows the expression of constraints that cannot be described using description logic;

However UML does not whole accomplish the needs regarding ontology development, so Cranefiled [80] suggested the use of OCL with UML. OCL is a formal language used to describe expressions in UML models specifying conditions that must be guaranteed for the system operation. OCL means Object Constraint Language and it's used for restriction specification in diagram modelling architected by UML. An UML diagram doesn't supply all the relevant aspects of a specification; thus, there is the need for additional restriction definition about the model object.

5.2 Model Driven Architecture

In the software engineering domain, new paradigms emerged in a parallel evolution with the development of semantic web in a time where modelling has a new growing meaning in the development and interoperability of systems.

Model Driven Architecture (MDA) is the new paradigm resultant from an OMG effort and it's about a new approach to the development of the systems based on the idea of separating system functionality specification from implementation details. Main MDA objectives are:

- Portability
- Interoperability
- Re-utilization

"MDA is based on the four-layer metamodeling architecture, and several OMG's complementary standards; these standards are Meta-Object Facility (MOF), Unified Modeling Language (UML) and XML Metadata Interchange (XMI)." [79], [81]

D. Djurić et al. [79], [81] has mentioned that MDA provide a solid basis for defining metamodels of any modelling language, so it is the right choice to define an Ontology-modelling language in MOF. Such language can utilize MDA's support in modelling tools, model management and interoperability with other MOF defined metamodels.

The main characteristics of MDA are based in the OMG technologies MOF, UML, CWM, XMI.

UML

Among the duties that UML perform in MDA it is distinguished the separation between the abstract syntax and concrete syntax where the formal semantic models are defined. These models define the concepts that UML use associated to the abstract syntax. The graphical notation is the concrete syntax that allow expressing what is defined by the abstract syntax

XMI

XMI appears to finish with the difficulties during the MOF models to XML mapping. XMI is used as a standard for exchanging data between several applications and middleware.

MOF

MOF establish standards for modelling and constructors of data exchange used by MDA. Other standard models as UML and CMW are also defined by MOF standard constructors. These standards supply the fundamental base for interoperability and the inter exchange between models and metadata, trough the XMI.

CMW

CMW is the standard for data storage, it has a set of specific metamodels for databases, registry structure and XML as well as a set of several metamodels for data processing rules as OLAP, etc..

5.3 Goals

The main goal is to easily set up Plone CMS content based on OWL ontologies. The procedure described here shows how to generate a set of custom content type without writing a single line of code. A set o Plone objects will be personalized according with a specific domain of interest, specified by the ontology in order to become easier the organizational content management. To achieve that, it will be necessary to pull together some tools, in order to form the necessary environment for ontology to plone content type transformation.

In practice, to complete this process it's necessary to convert the ontology into UML class model using standard XML-base format XMI (XML Model Interchange), open it in ArgoUML (or similar product) for some corrections, run the ArchGenXML script, restart ZOPE and install product. Finally we have the newly generated product in Plone site.

5.4 Requirements

The first step is to install and configure the necessary packages on the system. The packages used in this scenario were:

- Plone 2.5.2 - http://plone.org/products/plone
- Protégé 3.2.1 - http://protege.standford.edu
- OWL plugin for Protégé - Protégé 3.2.1 full version already has OWL Plugin.
- UML Backend 3.2.1 plugin for Protégé - http://protege.cim3.net/file/work/plugins/uml-backend
- ArgoUML 0.24 - http.//argouml.tigris.org
- ArchGenXML1.5.2 - http://plone.org/products/archgenxml

All of the packages have installation instructions; however it is not wastefulness of time to give some particularly tips about it during the tools presentation.

5.4.1 Plone

Plone is a solid content type management system. It was chosen by eWEEK LABS in 2004 as one of the best 10 products of the year and the best option in web publishing portal. Actually, plone is very agile and adaptable in order to meet new challenges or to implement new ideas. Plone, provides excellent content creation and user collaboration features, as well as customization and corporate integration capabilities. In fact, interoperability, capability and usability are indicated as its stronger points. During the last years the plone community has growing at a good rhythm. Plone main features include an enterprise-level content management system, a solid workflow engine, a wiki role-based membership, and search engine. Sharing information in enterprise intranets becomes easier. Plone has reached a high level of utilization which motivates emergent projects using plone and for plone. Plone should be the first thing to install. Installing and running plone require a few easy steps. Plone has already a huge community and one of the best Plone advantages is actually the documentation and internationalization support. At plone site we can find several documentation such as manuals, how-tos and tutorials.

5.4.2 Archetypes (CMFTypes)

Archetypes is a framework for developing plone products. The main goal is to provide a common method for easily creating new content types without having to create products by hand. Archetypes automatically take care of generating edit forms, validation logic, view displays, and other details. Archetypes avoid a lot of hard work, allowing a large investment in business role definition and planning issues. Plone 2 already comes with Archetypes, thus isn't necessary to install it in a supplemental package.

5.4.3 UML backend plug-in

"The UML back-end plug-in provides an import and export mechanism between the Protégé knowledge model and the object-oriented modeling language UML. In particular, it enables the exchange of ontologies and UML class diagrams so that Protégé can be used in conjunction with traditional Software Engineering tools, including CASE tools and Integrated Development Environments." *[http://protege.cim3.net/cgi-bin/wiki.pl?UMLBackend]*

It is necessary to cite that some restrictions about this plug-in are known and were divulged. Issues related with export limitation in which only the parts of the ontology that can directly be expressed in UML are processed. Other limitation is the fact that this plug-in is based in UML 1.4.

The installation of UML plug-in is very simple. The efforts required are just the download of the file, and then extract it to the plugins directory of the protégé installation folder.

5.4.4 ArgoUML

ArgoUML is an open-source application that uses UML to modeling software. However, ArgoUML still under improvement. Currently ArgoUML includes support for all standard UML 1.4.

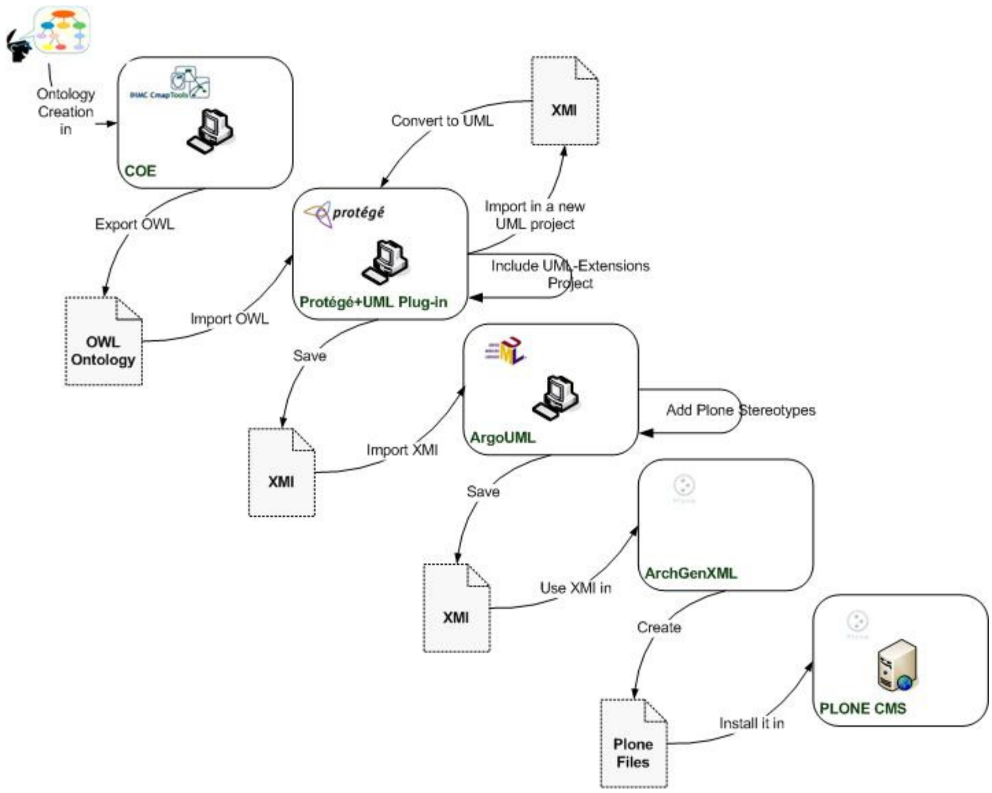
5.4.5 ArcheGenXML

ArchGenXML is a command line utility which auto generates code from a UML model in order to create a valid archetype schema. With ArchGenXML a lot of time is saved.

5.5 How-To

With the purpose of building a valid Plone Product or plug-in into PloneCMS, several steps are needed. The main idea is mapping an OWL ontology into a UML model encoded in XMI and use ArchGenXML to convert the XMI UML model into the files needed to install the pug-in in PLONE. Seems easy at a first sight but with all the iterations that the process entails it becomes a little more complex. The figured below illustrates the process and its steps.

Figure 5-65 OWL-to-Plone steps



5.5.1 The Ontology Developement

First step is to have a valid OWL ontology. There are several software tools for ontology development beside Protégé. Tools such as CmapTool COE, OntoEdit and OILEd among others can be used for ontology creation. In this particularly case, I have chosen CmapTools COE to develop my ontology. COE has a set of templates for a graphical ontology construction approach that facilitates the creation of the base of ontology. In order to demonstrate how we achieve a full functional Plone Product, it was used an ontology for Job Analysis concept.

In Practice, the result of a job analysis activity is a written document with reference to the main characteristics of a certain job. In that document it can be found the job description, which is an outline of the main tasks of a job, a set of summarized outlines about the essential and desirable qualities to fit the job vacancy and the work performance expected from an employee. Thus, the next figure represents an example of what could be the contents of a job analysis study.

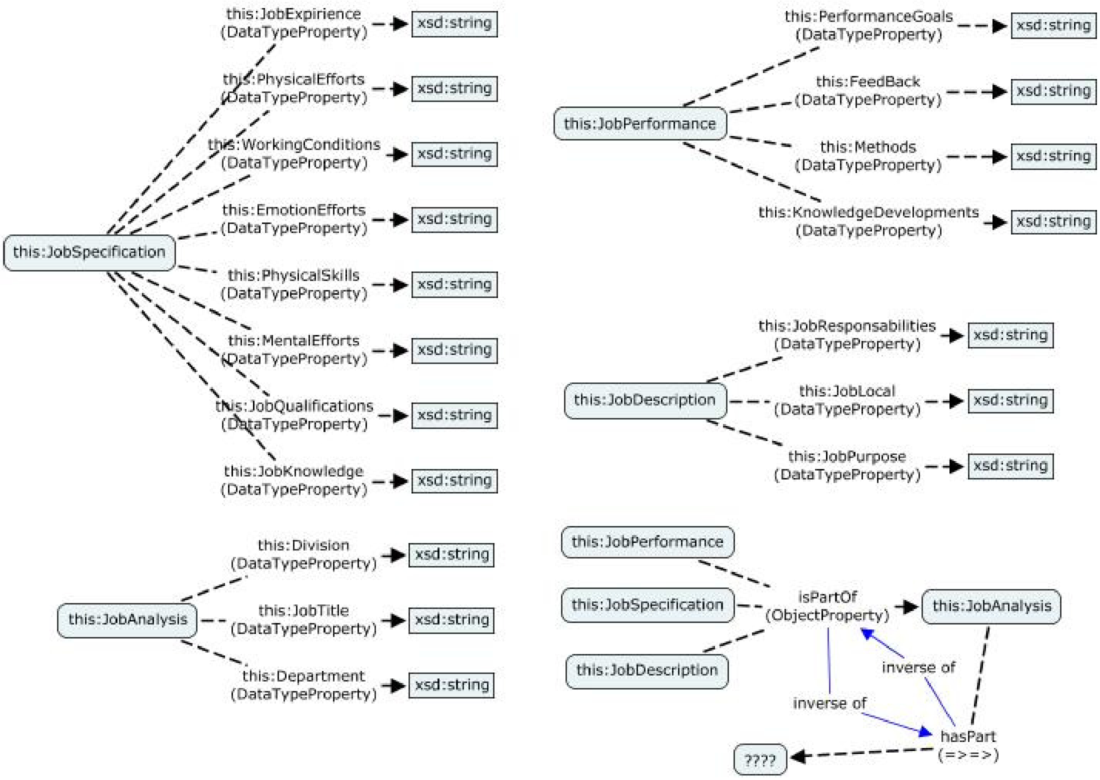


Figure 5-66 Job Analysis Ontology

The figure above is a graphical representation of the Job Analysis ontology, which was developed with the proper specific notation of CmapTool COE from a set of available templates that when combined allows the graphical representation of all OWL elements. The map can be exported to an OWL standard file by selecting the “OWL Ontology File...” option in the “File” Menu of the program.

The Job Analysis ontology was the started point for all the next steps needed to mount this ontology into plone.

5.5.2 OWL Ontology to UML

The process of converting an OWL ontology into UML is achieved by loading the OWL files into Protégé. After load it we simple need to save the project (Project > Save) and then convert it into UML (Project > Convert Project > UML). Now we have an UML project based on the previous ontology files. Nevertheless, it is necessary to load new UML project back into Protégé in order to include the UML-Extensions project provided by the UML backend plug-in, to accomplish that it is necessary to perform the following steps:

a. Create a new project (Project > Build new Project > UML).

b. Load the UML saved project files back in.

c. Add the UML extensions project (see the figures below).

Figure 5-67 Protégé – Manage Included projects.

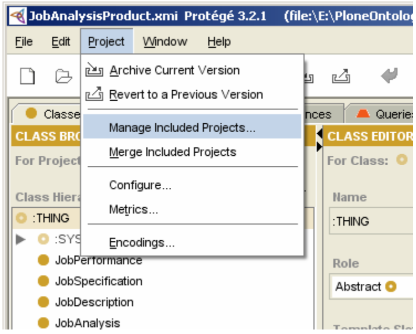


Figure 5-68 Protégé – Include UML-Extensions Project (1).

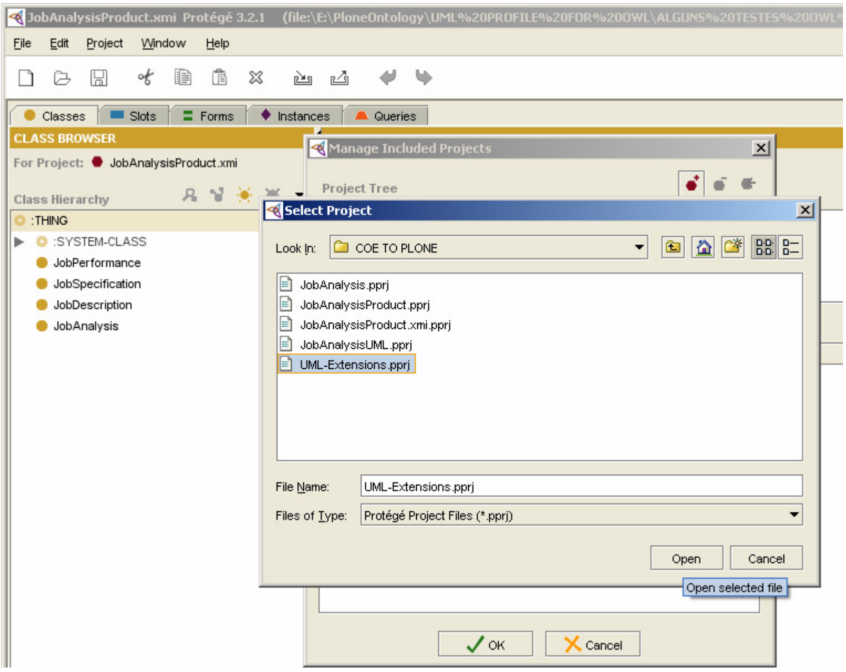
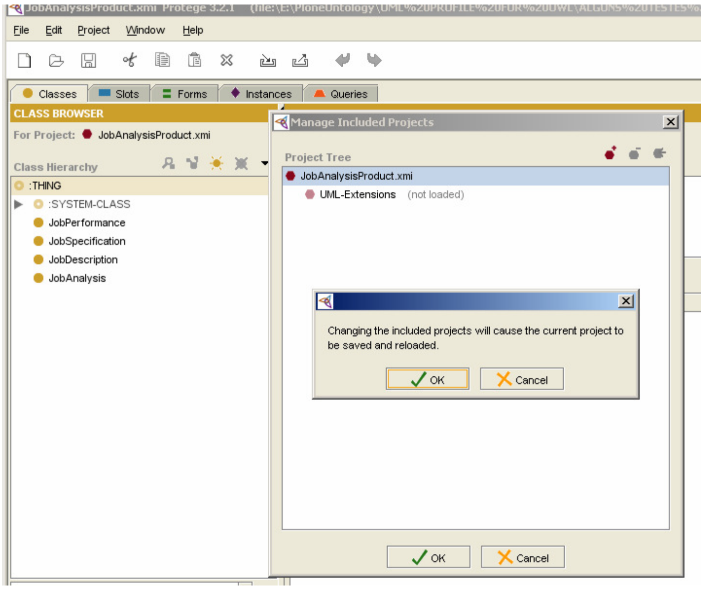
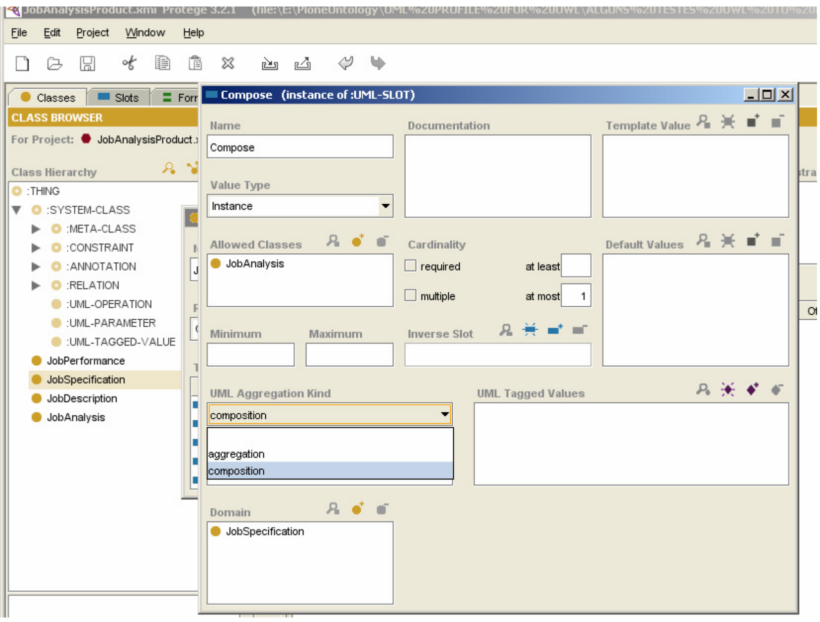


Figure 5-69 Protégé – Include UML-Extensions Project (2).



The UML extensions project is necessary to endow protégé with extra UML slots in order to be possible to specify relationships such as aggregation (see figure below) and composition, which are not possible to specify in OWL.

Figure 5-70 Defining composition with UML slots in Protégé



In this particular case it was added a composition relationships between Job Analysis concept and Job Specification, Job Description and Job Performance.

Before saving the project again it is necessary to perform some more steps:

a. Remove OWL, RDF meta-information within Protégé

b. Delete extraneous OWL and RDF classes, move entity up to subclass of :THING

c. Make sure to also delete :THING->SYSTEM-CLASS->META-CLASS->CLASS->STANDARD-CLASS->all OWL, rdfs classes

d. Remove extraneous OWL, RDF meta-properties

e. The project is ready to be saved (Project->Save as...) in order to resave the modified xmi file.

Each time is created or converted a protégé OWL project into an UML project, a set of correspondent XMI files are created. The XML Model Interchange Language (XMI) defines a standard way to serialise UML models. Using XMI, it is possible to transfer an UML model from one tool into a repository, or into another tool for refinement or the next step in your chosen development process. In simple words, an UML class diagram can be stored in an *.xmi* file.

But it is not all that simple. There are some drawbacks regarding this approach that we must be careful with, namely:

- All cardinality constraints on associations are lost and must manually re-create in EA diagram.
- "Comment" is used for all kind of comments, instead of "Documentation" for classes and "Description" for attributes.
- Enumeration (controlled vocabularies) must be manually input

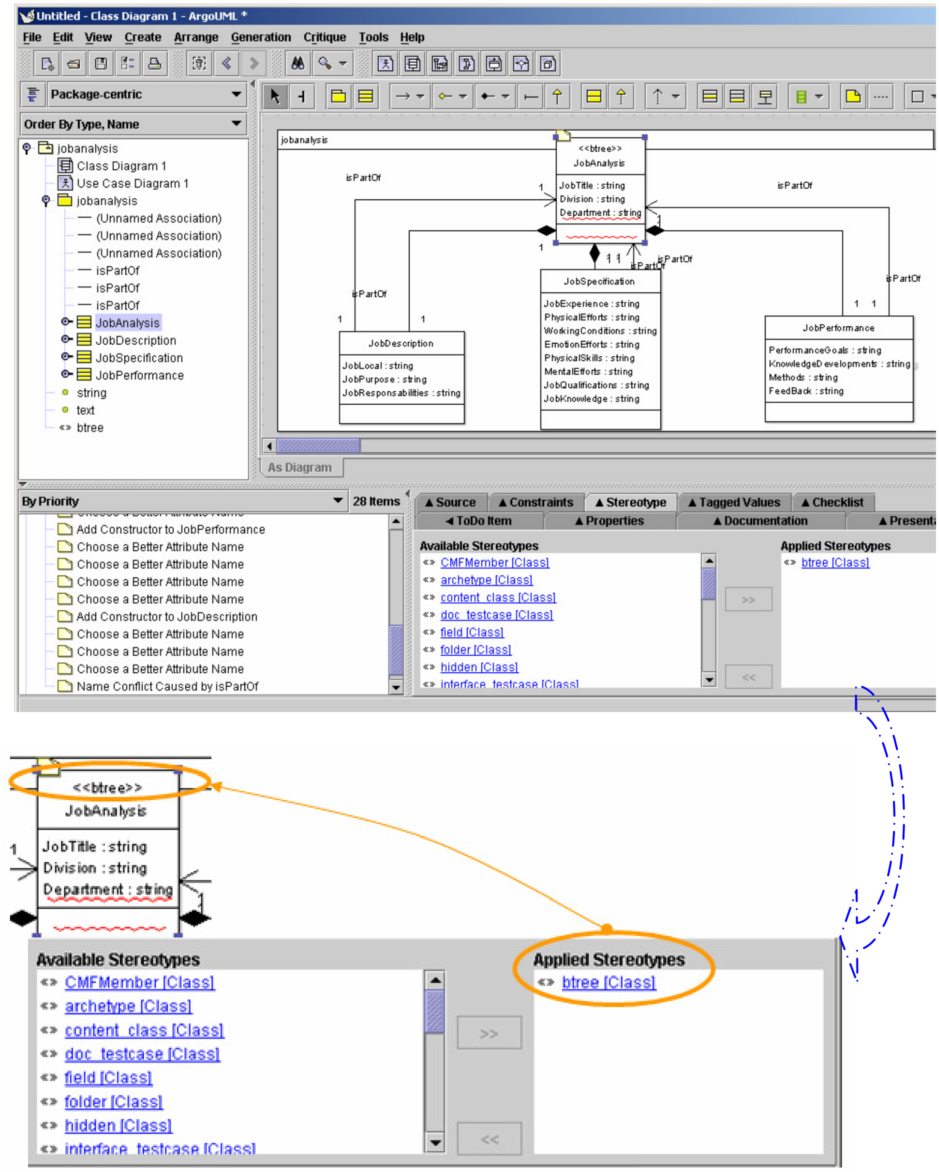
5.5.3 UML to Plone Schema

At this stage we have an UML ontology stored in an .xmi file.

The next step is to import the xmi file into ArgoUML(File > Import XMI). ArgoUML supports the ability to load a "definition" file with predefined tagged values and stereotypes. This makes it possible to have all the custom tagged values and stereotypes used by a Plone project available in dropdown menus within ArgoUML.

The next figure shows the Job Analysis ontology as an UML class diagram in ArgoUML. It can also be seen the available Plone stereotypes.

Figure 5-71 UML Class diagram of Job Analysis ontology

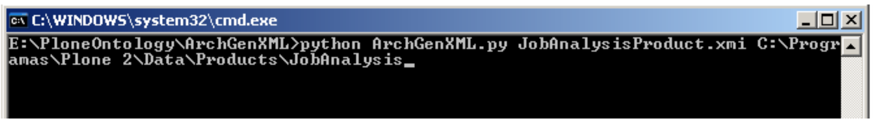


As shown in the previous figure, it was attached a <<btree>> stereotype to the JobAnalysis class which means it will be generated a folderish object using a BTree folder for support of large amounts of content.

5.5.4 Creating a valid Plone product

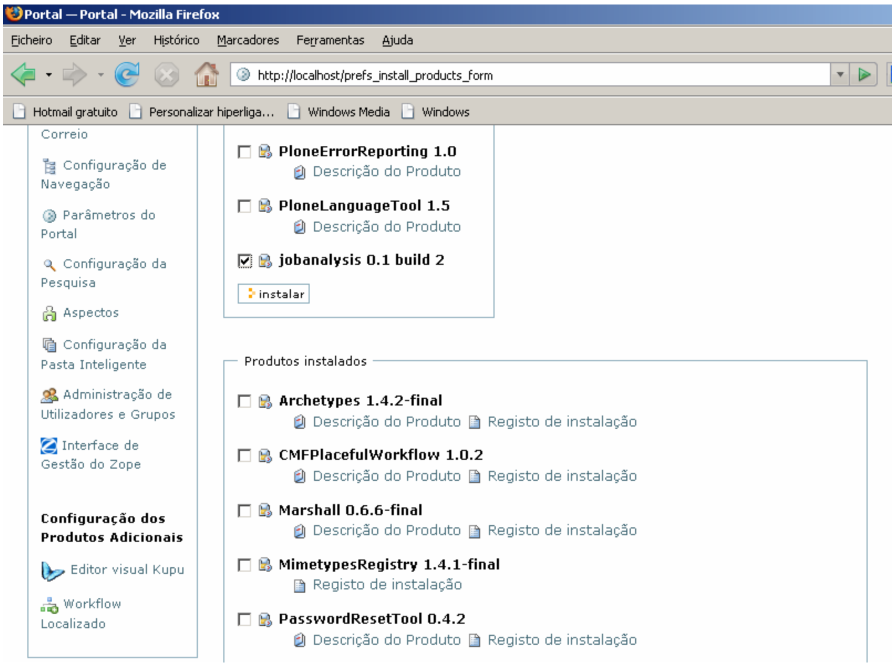
Subsequent to have the UML class diagram saved in an xmi file, we are now ready to build a JobAnalys product for plone. But, to accomplish that goal it is necessary to use the ArchGenXML utility as the next figure shows.

Figure 5-72 Creating a Plone product through ArchGenXML



After performing the operation shown in the previous figure, it is needed to restart ZOPE and then access to Zope administration interface for the final install of the new JobAnalysis Plone product as the next figure exemplifies.

Figure 5-73 Installing a Plone product



Restart ZOPE again is advisable in order to be possible the full use of the new Plone product that was initially conceptualized collaboratively using CmapTool COE.

The final step is to present how to use the JobAnalysis Product in Plone. Note that the way product will be used is directly dependent of the ontology created.

5.5.5 How to use JobAnalysis in Plone CMS

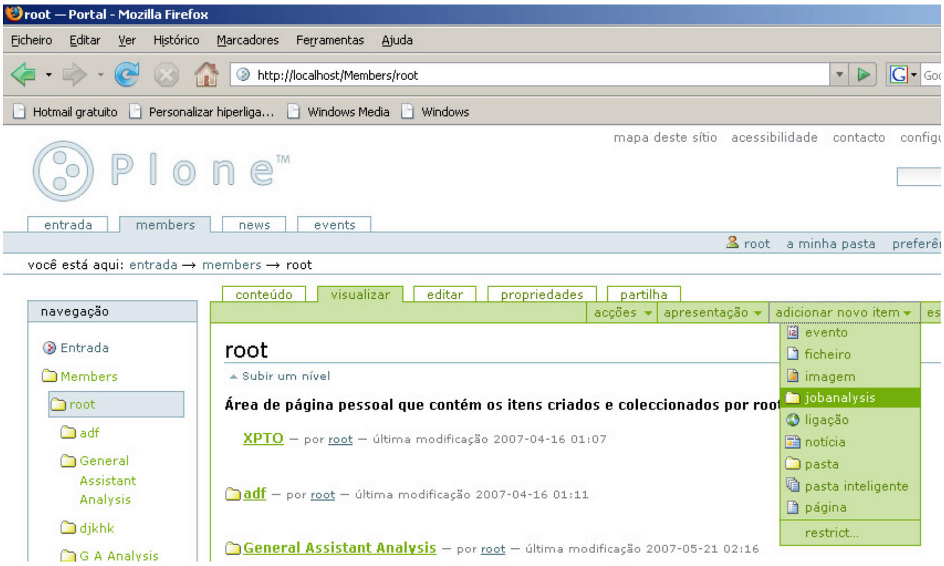
Scenario:

Lets suppose that a consultant is about to start analysis a specific job and will use the CMS of the organizations to impute his results.

Step 1:

The first thing to do is create a new Job analysis thread as shown in the next figure.

Figure 5-74 Inserting the brand new JobAnalysis plug-in.

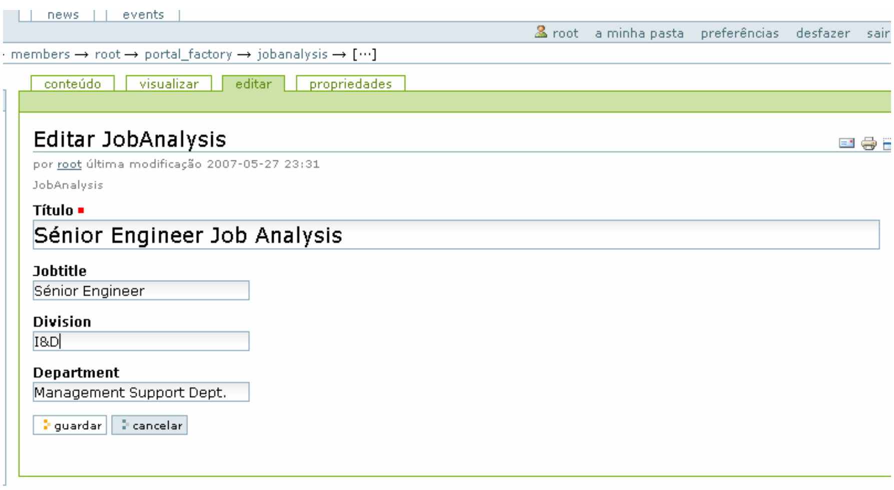


Step 2:

Immediately after creating a new job analysis thread the consultant will be prompted with some information to be provided such as (see figure below):

- The Title of the thread
- The title of the job
- The division where the job belongs to
- The department where the job belong to

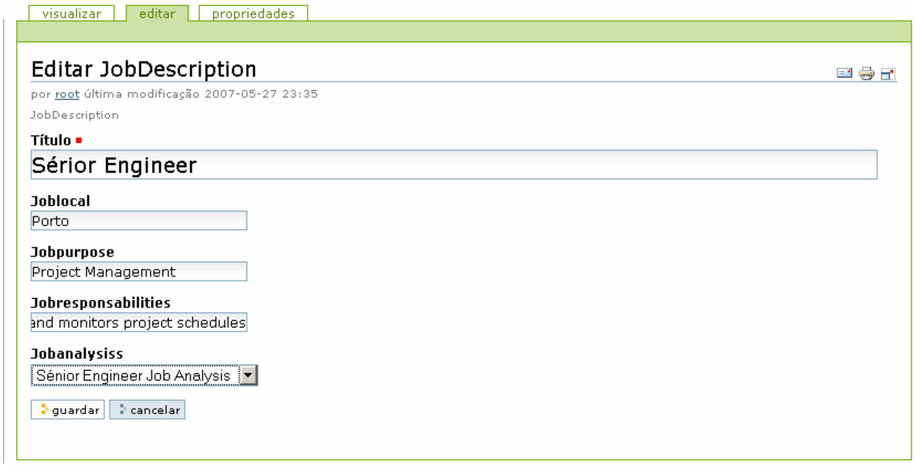
Figure 5-75 Adding content to JobAnalysis



Step 3:

After providing all the information about the job under analysis, the consultant must also carry out the specification of all the concepts that perform the complete analysis of the job as specified in the ontology and which entails job description, job specification and job performance.

Figure 5-76 Adding content to JobDescription

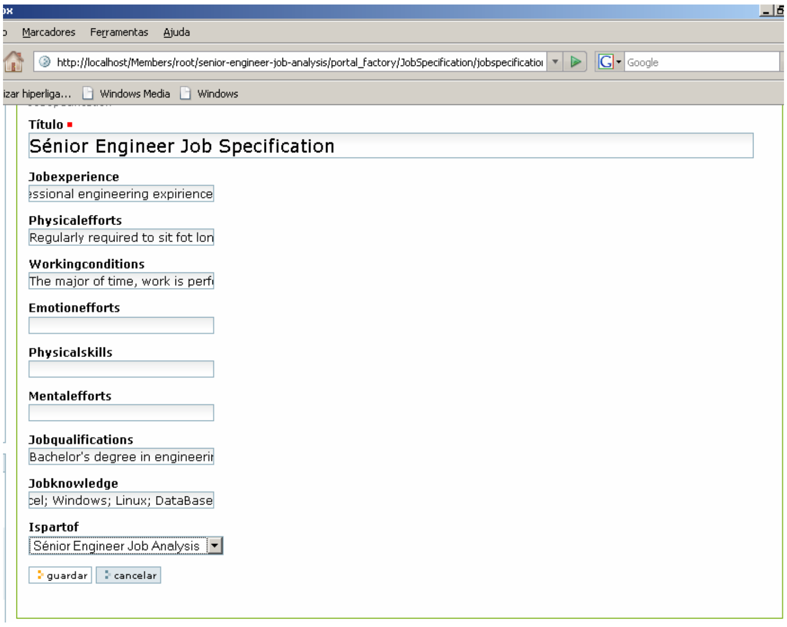


Note that a relationship between job analysis and job description was establish before, thus it must be specified at this step, the job that this description refers to, which is, according to the previous figure, performed by selecting an item from a dropdown list.

Step 4:

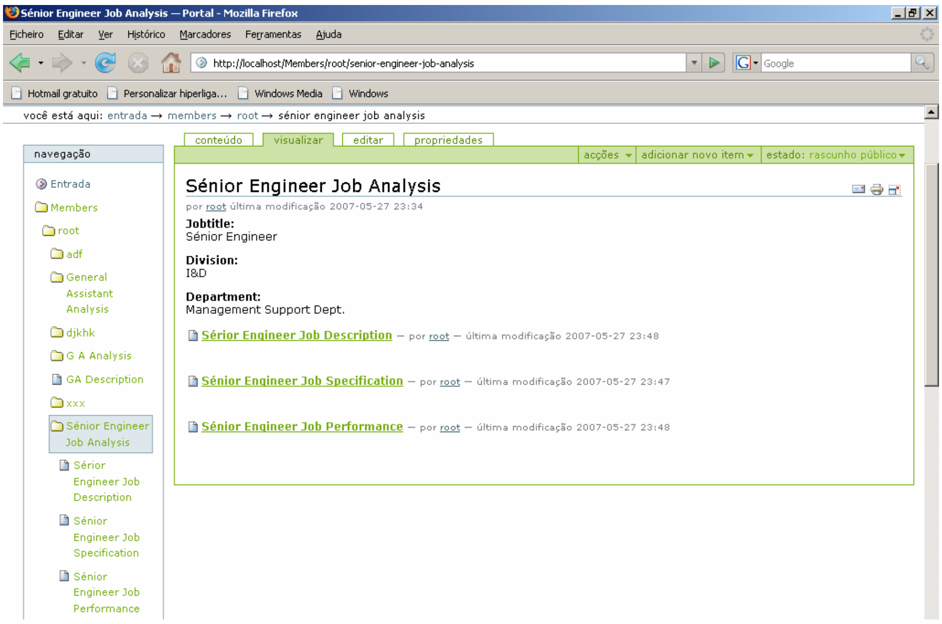
The same process must be performed for job performance and job specification.

Figure 5-77 Adding content to JobSpecification



At the end we can easily navigate through the result of the Job analysis process consulting the specification, description and performing requirements for a specific job.

Figure 5-78 Visuslizing Job Analysis Content



Despite of the volume of iterations that are necessary to obtain a full functional plug-in conceptualized in a CM is an interesting practical approach of ontological applications. In this example we can easily define how the work organization should be, and then we can use that knowledge to build a content organization in a CMS, that will support consultants and/or even collaborators in the design of the work, activities, specifying the duties and responsibilities and collect real information about their working conditions.

Chapter 6

Conclusion and Future Work

In this thesis it was described an approach to manage knowledge about work organization making use of ontologies.

New forms of work organization are a channel for productivity improvement. Organization of work involves an interaction between work design, individuals and groups. For this reason work organization requires the participation of all people because the tasks are not any more started and finished by the same person and in the same department or in the same organization. It is in this context that ontologies play an important role

It stand firmly that work organization design is a multidisciplinary process which involves several teams with different know-how, collaborating for the same propose, that is, to find the best way of performing work without it became strain.

Ontologies can expose behavioral characteristics and operational characteristics of work. They are used in a particular domain of discourse enabling a common sharing and understanding of that domain, which is in this case work organization design. Through ontologies it becomes easy to involve several teams in the process of change, finding a common model, sharing knowledge and point-of-views.

Ontology is a formal specification of a domain which demands the knowledge about some technical requirements and specific representation conventions. The knowledge is not only in the ontology engineers heads, thus it is highly recommended the specification of a knowledge model supported by simple knowledge representation formalisms that allow domain specialists to discuss together about a certain domain in order to be obtained a set of concepts and its relations according with the experience, know-how and surveys of the right people without having the concern about knowledge representation technical issues.

In the first step of knowledge acquisition, Cmaps do fairly the job due its informal characteristics. However an easy knowledge formalism is not sufficient by itself to support the construction of a knowledge model. Thence came into existence the CmapTools Software by IHMC providing a complete collaborative knowledge representation environment based on CMs. Easy to use and to understand CmapTools conquered a great number of adherents.

By definition ontology has formal constraints, so it is necessary to transform cmaps into a formal notation. Here we have found the first problem. It is not a simple task to perform even having a knowledge model for guidance. Protégé appears to be a solution since it is the most used software for ontology building; however protégé looses on its visualization features. In any case Protégé has undergone several revisions and today, and thanks to its plug-in architecture, it offers a variety of visualization solutions, where Jambalaya and recently Map2OWL are the closer to COE features and propose.

CmapTools COE appear as another solution. Since cmaps are the best choice for the first stage in ontology development COE offer an integrated environment to formalize the ontology. However, and by experience acquired on CODEwork project, cmaps are not easy to validate due its informal characteristic besides, the OWL exported is not fully compatible with other tools such as Protégé for example. cmaps can be validated, but through empirical methodologies such as validating with other cmaps and by conceptual mapping principals itself. In this context CmapTools can validate links but not the knowledge structures. This issue persists when using COE for ontology edition; we can draw formal knowledge structures that subsequently are not compliant when exported.

Combining cmaps and Protégé is a new demand for matching the potentialities of both tools. At least one project by Institute of Molecular Bioscience of the University of Queensland, Australia, is in progress to achieve a cmaps Plug-in for protégé. Meanwhile, and at the end of our research, we were noticed about the new Map2OWL protégé plug-in which allows the direct construction of cmaps within protégé. Despite of being less powerful than CmapTool software it is very interesting.

Step-by-step with this alliance between cmaps and Protégé, we consider important to expend some investigation on COE, in order to develop a framework that implements a solid methodology for cmaps translation into COE formal conventions. This translation should be interactive and iterative with structure validation mechanisms. This approach will reduce the need of ontology engineers and will increase the number of collaborators that can participate on the ontology development.

We also foresee cmaps and Ontologies as a solution for today's' new programming challenges. Massive access to the internet by all kind of people trying to rapidly retrieve information from several different sources either for professional or personal use, ask for new ways to develop and present web content. It is necessary the development of easy to use systems allowing inexperienced computer users for robust and queriable access for their own. By other hand systems must be designed in order to allow that complex operations related with technical issues and content sets, can be done more efficiently.

A possible future approach could be based on concept maps as a tool to set up a CMS for implementing e.g., a web portal. At the same time, it would be interesting to have the reverse synchronous standard mechanism in which the content created in a CMS could be used to add new instances to the OWL ontology. Perhaps CMS could be easily used to maintain and evaluate ontologies. Interoperability Semantic Web could be the answer at a short/medium term.

The goal is to provide users a rich visual interface to organize content and set up some workflow roles and document sharing capabilities mediated by CMAP OWL ontologies allowing a suitable integration with Plone. At the end, we should have a complete hierarchical structure of folders where we can navigate through content by keywords and their relations. The experiments conducted so far and presented in this thesis leads to many iterations and import and export operations using Plone's archetypes which are able to import any schema specified in the form of an XMI file output by any UML modelizing editor.

A more simple and efficient method is needed in order to integrate an ontology developed in CmapTools into a plone portal. The work to be done involves the research of the architecture of the Plone CMS/Portal, configuration of "plugin" components to add functionality, including new content types and workflows and utilization of the Plone Ontology Pugin in particular.

All the work mentioned and further intended research, is aligned with the new trends of Knowledge Management requirements, achieving an information management system not only from a technological point-of-view but in a multidisciplinary one.

Apendix A

Work Organization Ontology Glossary of Terms

Organization Design

Organization Design is the process of choosing and implementing structures that best arrange resources to serve organization mission and objectives *[59]*.

Is the specialization of Work Organization in the work level context. Is the management of decisions and actions that result in a specific organizational structure *[60]*. Organizational design build and the organizational structure in order to achieve organizational goals. According with Shermerhorn [59], organization design tries to integrate people, information and technology.

Work Design

Work Design is the process of designing in advance all the features of a task or job, specifying the way the worker will do his/her work inside or between organizations according with the social, environmental and technological dimensions of the organization. *[45]*

Work System

A work system is a system in which human participants and/or machines perform business processes using information, technologies, and other resources to produce products and/or services for internal or external customers *[64,],[56]* .

Vertical Design

Is a of work design specific case.

Is the process of work division where all the resources, knowledge and authority stay in one or more hierarchical levels inside the organizations. This process of dividing work must have into consideration the hierarchical decomposition, the modularity of the division of work and the network division of work. [35]

Horizontal Design

Is a scenario of work design. Is the process of work division, where the work system can be distributed trough several organizational units of one or more organizations, where each one supplies one module to the network. [35]

Modularity

Modularity is a process of designing work in which the work system is decomposed into simple work components or modules as independent as possible, communicating one another only trough standardized interfaces within a standardized architecture. At the sane time and in order to reduce the work system complexity the process of information decomposition must occur otherwise the work system would be unmanageable. [35], [47]

Cross-Functional Teams

A Cross-Functional Teams is composed of individuals with different skills and backgrounds form a team to bring a wide range of viewpoints to accomplish some task. They collaborate on common work issues and learn from one another. Through this, they acquire greater knowledge of the complexities of business issues and decision-making processes *[58]* .

Teams are created to knock down "walls" separating departments *[59]*.

Work Module Interfaces

Is a set of communication patterns and connections among work modules or components inside and between units of one or more organizations. Describe in detail how work modules will interact, including how they fit together and communicate. [47]

Work Module Standards

Is a set of rules for work modules definition and decomposition in order to achieve a satisfactory level of work division, as well as a set of performance evaluation patterns. Modules Standards test modules conformity to design rules and measure the modules performance relative to other modules. [47]

Work Module Architecture

Is about the way each work component or module is organized in order to be recombined with other components with the minimal changes in the overall system. Each work module has a set of responsibilities in all system and a set of specific functions. Each module has also “hidden” and “visible” functions, which means, some of the functions are under exclusive control of the module. This is the work module authority. [47]

Hierarchical Decomposition

Hierarchical Decomposition is the process of definition of coordinating roles and responsibilities for work across hierarchical levels of an organization or network of organizations.

Network Design

Network design is a complex process derived from the "interaction effects of modularity and hierarchical decomposition" [41]. Network design is a systematic process of formulation of a common network strategy and socialization by trust between organizations and different units within an organization. Network design is the definition of the social mechanisms for coordinating and control all the entities (units or organizations) involved in the overall work network. There are three dimensions of structure important for Network design: Centralization In network design we must have in consideration three distinct dimensions, degree of hierarchy, centralization, and hierarchical levels. Degree of hierarchy is reflected by the degree to which relationships in a network are directly or indirectly reciprocal. Reciprocal relationships indicate teamwork, while an abundance of unreciprocated relationships are seen in more hierarchical networks. Centralization reflects the extent to which a network or group is organized around its focal point. It is a measure of integration or cohesion of the group. A centralized network may reflect an uneven distribution of knowledge such that knowledge is concentrated in the focal points of the network. Finally, hierarchical levels reflect the number of levels one must go through in order to obtain information. An existence of hierarchical levels indicates that members must go through someone rather than directly obtain information from the source. [41]

Joint Ventures

Joint venture is a legal organization formed between two or more parties undertake economic activity together for mutual profit. Both parties contribute equity sharing revenues, expenses and enterprise control. Joint ventures can involve any type of business transaction and involves individuals, group of individuals, companies or coorporations.

Virtual Teams

Is a group of people who interact through independent tasks guided by a common propose that works across space, time and organization boundaries with links strengthened by webs of communication technologies. [33]

Temporary VO's

Temporary Virtual Organizations are decentralized, team-based organizations with distributed structures a set of independent organizations that share resources and skills to achieve its goal, with no common past and with a low level of probability of working together again. [41]

Permanent VO's

Permanent Virtual Organizations are decentralized, team-based organizations with distributed structures a set of independent organizations that share resources and skills to achieve its goal, where each of them had a common history of collaboration and a high probability of working together again in the future. It means that enterprises know each other well. [41]

Trust

The concept of trust in the virtual organization domain has been largely argued as I have been aware trough my researches. It can be distinguished two different views about trust: the traditional view and the concept of swift trust. According with Rousseau, trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intention or behavior of another, fits the reliance/risk trade-off that characterizes inter-organizational alliances. In all definitions of trust is mentioned the difficulty to establish trust without some face-to-face contacts. According to the roles of trust by Charles Handy, "trust need touch" and "a shared commitment still require personal contact to make it real" [49]. Charles Handy said that the more virtual organization, the more people need to meet in person. Another vision of trust is based in the concept of swift trust developed by Meyerson for temporary teams. Swift trust deemphasizes the interpersonal dimension and is based initially on broad categorical social structures and later on action. Because members initially import trust rather than develop trust. According to this view is possible to maintain trust at distance.

Authority

Is the control relationship between Organizational Agents (OA). An Organizational Agent is an individual member in the organization *[51]*. In the VO's context we can add that an OA could be an organizational unit of one or more organizations. For an organization Agent to have authority over another implies that the first one is able to extract commitment from the second one to achieve a goal that is defined as part of the 2nd Organization Agent's organization roles. In order to extract that commitment OA1 gas to be related directly or indirectly by a communication-with-authority link relation *[51]*.

H.R.M Design

Human Resource Management study and establish guidance for relationships between people and his/her job and workplace, as well as the relationships between workers in order to satisfy the organization goals and workers well-being. Human Resource Management has the main responsibilities such as:

- assistance in diagnosing organizational problems.
- assistance in planning and implementation of job (re)design.

- modification (if necessary) job descriptions, job specifications, and eventually adjust pay rates of some positions.

Job Design

Job design is the process of determining the arrangement of the features of a job or group of tasks to satisfy economic and social needs of each member of the organization. The job design is performed based on job analysis.

Job Rotation

Is about rotating tasks among workers and have workers move from one task to another according to a schedule. Job Rotation is the practice of periodically moving workers around a number of tasks in order to increase variety.

Job Enrichment

Job Enrichment is an increase in vertical loading of a job by combining it with some tasks and responsibilities normally performed by others to make the job more challenging and to increase responsibility.

Job Enlargement

Is the process of adding more tasks to a job. Job Enlargement is an increase in horizontal loading of a job by incorporating other tasks requiring similar levels of skill to increase variety.

Job Engineering

Is the process of maximizing efficiency through time and motion studies and man-machine interfaces.

Task identity

The degree to which job requires completion of a whole and identificable piece of work.

Task variety

The degree to which a job requires a variety of challenging skills and abilities.

Task significance

The degree to which the job has a perceivable impact on the lives of others, either within the organization of the world at large.

Autonomy

The degree to which the job gives the workers freedom to independence in scheduling work and determining how the work will be carried out.

Feedback

The degree which the worker gets information about the effectiveness of his ou her efforts, either directly from the work itself or from others.

Job Analysis

Job analysis is the systematic study of a job to discover its specifications and skill requirements. *[35]* Job analysis is the determination of the essential characteristics of a job in order to produce a job specification. *[45]*

[45] According with Robert J. Harvey job analysis may be defined as the collection of data describing job behaviors performed by workers and characteristics of the job environment. Specifically, there are three major defining characteristics of job analysis: First, job analysis methods should have as their sole goal the description of observables. Second, a job analysis should involve the description of work behavior, not the personal characteristics or performance effectiveness of the employees who perform the job. That is, the work itself is being described, and this description must not be influenced by whether the employees currently hired to perform the job are doing so successfully. Third, job analysis data must be verifiable and replicable. According to this definition of job analysis, a fundamental distinction exists between the description of job activities and the specification of the Knowledge, Skills, Abilities, and Other characteristics thought to be necessary for successful job performance (also known as competency identification).

This is accomplished by gathering task activities and requirements by observation, interviews, Employee log, or other recording systems. Job analysis also has important implications for fair employment practices. In order to demonstrate job relatedness in employee selection, the criteria utilized should be directly related to, or embody constructs associated with job performance.

Job Specification

Job Specification is a written statement of the essential characteristics of a job including necessary qualifications, duties, responsibilities and degree of authority of the job holder. *[45]* Job demands are in the basis of Job Specification concept which can be designed as a set of summarized outlines about the essential and desirable qualities which would make the ideal person to fit the job vacancy.

Job specification describes the demands for the human factors required. Those factors could be grouped in competency factors, effort factors and working conditions factors where each one would have specific demands. For instance, problem-solving demands or leadership demands are competency factors; by other hand the type of environment (aggressive, cold…), concerns to working conditions. Hobbies, for example, are considered demands for human effort factors.

Job Effort

Job effort is a collection of Lifestyle Characteristics and the physical and mental demands of the job.

Job Competency

The needful knowledge, skills, experience and ability are the minimum factors related with competency that must be defined for each job.

Working Conditions

The environment description, the job working hours the salary survey and pay structure are all factors concerning to the working conditions of a specified job and are the minimum issues that must be written about it.

Job Descriptions

Job Description is a written outline of the main tasks of a job. *[45]* Job Description is a recognized list of functions, tasks, accountabilities, working conditions and competencies for a particular occupation or job. *[35]*

Job Performance

Job Performance is the work performance expected from an employee on a particular job

Objectives or targets for employee efforts

Criteria for measuring job success

Job Evaluation

"Formal Process used to assign salary rtes to jobs, not to employees." *The Guide to Using the CMS for Human Resource Applications* (...)

Compensation

"Compensation is a systematic approach to providing monetary value to employees in exchange for work performed. Compensation may achieve several purposes assisting in recruitment, job performance, and job satisfaction." *HR-Guide.Com*

Ergonomics

Ergonomics is the study of a person's work related to the tools and machines he/she uses to accomplish the task of work. Ergonomics is also the study about the time involved in work.

Ergonomics is the science of fitting workplace conditions and job demands to the capabilities of the working population.

Ergonomics of Conception

Ergonomics of conception is oriented for the creation of workplaces and tools adaptable to the workers.

Ergonomics of Correction

Ergonomics of Correction is oriented for the adjustment and improvement of the work systems already implemented.

Work Station Design

Work Station Design is the process in which is defined how the set of resources that forms a productive unit at a particular location is arranged in tune with environmental factors such as noise vision thermal and chemical factors. Work station design define the drives for worker to perform the tasks regarding his posture, the force used to do the tasks, repetition and breaks and material handling avoiding work-related musculoskeletal disorders.

Tool Design

Tool design is the process of establishing the anatomical principals of tool conception in order to reduce unnecessary efforts, discomfort and injuries in task performing avoiding work-related musculoskeletal disorders and performance deterioration.

Material Handling

Is the Process in which is defined the best way to operate, use and transport equipment, tools and other materials in order to achieve the best performance with the less effort avoiding work-related musculoskeletal disorders and performance deterioration.

Visual and Auditory Design

Ergonomics is not only about anatomical issues. Visual and auditory factors are environment factors that must contribute for performance deterioration and stress.

Bibliography

[1] H. Yao and L. Etzkorn, "Automated conversion between different knowledge representation formats," Knowledge Based Systems, vol. 19, pp. 404-412, 2006.

[2] J. F. Sowa, "Semantic Networks," 2006.

[3] J. Novak and A. Cañas, "The Theory Underlying Concept Maps and How to Construct Them," Florida Institute for Human Machine Cognition, 2006.

[4] P. Hayes, T. C. Eskridge, T. Reichherzer, R. Saavedra, M. Mehrotra, and D. Bobrovnikoff, "COE: Tools for Collaborative Ontology Development and Reuse," 2006.

[5] A. Gruber, R. Westenthaler, and E. Gahleitner, "Supporting domain experts in creating formal knowledge models (ontologies)," presented at I-Know'06: 6th International Conference on Knowledge Management, Graz, Austria, 2006.

[6] A. Garcia, A. Norena, A. Betancourt, and M. Ragan, "Cognitive support for an argumentative structure during the ontology development process," in 9th Intl. Protégé Conference. Standford, California, 2006.

[7] G. Almeida, L. d. Oliveira, and S. Aluísio, " A terminologia na era da informática.," in Cienc. Cult., vol. 58, 2006, pp. 42-45.

[8] U. Petersen, H. Schärfe, and P. Øhrstrøm, "Conceptual Graphs," Aalborg University - Department of Communication, 2005.

[9] N. Ernst, M.A.Storey, and P. Allen, "Cognitive Support for Ontology Modeling," International Journal of Human-Computer Studies, vol. 62, pp. 553-577, 2005.

[10] A. Díaz and G. Baldo, "CO-Protégé: A Groupware Tool for Supporting Collaborative Ontology Design with Divergence," in 8th International Protégé Conference. Madrid, Spain, 2005.

[11] S. Koshman, "User vs. System Generated Similarity Measures with a Visualization Graph-Based web Interface," in www 2004 Conference Workshop. New York, 2004.

[12] P. Mika and H. Akkermans, "Analisys of the sate-of-the-Art in Ontology-based Knowledge Management," Vrije Universiteit, Amsterdam, Project Deliverable 14/02/2003 2003.

[13] P. Hayes, T. Eskridge, R. Saavedra, T. Reichherzer, M. Mehrotra, and D. Bobrovnikoff, "Collaborative Knowledge Capture in Ontologies," in Semantic Integration Workshop. Sanibel, Island, 2003.

[14] J. Davies, D. Fensel, and F. v. Harmelen, "Towards the Semantic Web," in Ontology-driven Knowledge Management: John Wiley & Sons, Ltd., 2003.

[15] H. Alani, "TGVizTab: An Ontology Visualisation Extension for Protégé," presented at Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering, Sanibel Island, Florida, USA, 2003.

[16] A. Cañas, G. Hill, A. Granados, C. Pérez, and J. D. Pérez, "The Network Architecture of CmapTools," Institute for Human and Machine Cognition (IHMC) 2002.

[17] J. F. Sowa, "Conceptual Graphs Standard," 2001.

[18] J. D. Heflin, "Towards the semantic web: knowledge representation in a dynamic, distributed environment," in Department of Computer Science, vol. Doctor: University of Maryland, 2001, pp. 236.

[19] R. Stevens, C. Goble, and S. Bechhofer, "Ontology-based Knowledge Reresentation for Bioinformatics," Briefings in Bioinformatics, vol. 1, pp. 398-416, 2000.

[20] S. Pepper, "The TAO of Topic Maps," presented at XML Europe 2000, Paris, France, 2000.

[21] J. L. Gordon, "Creating Knowledge Maps by Exploiting Depend Relationships," Knowledge Based Systems, vol. 13, pp. 71-79, 2000.

[22] F. Baader, "Logic-Based Knowledge Representation," in Artificial Intelligence Today: Recent Trends and Developments, vol. 1600/1999, Lecture Notes in Computer Science: Springer Berlin / Heidelberg, 1999 pp. 13.

[23] D. Wang, "Knowledge Representation in Health Sciences," 1999.

[24] N. Guarino, "Formal Ontology, conceptual analysis and knowledge representation," Internation Journal of Human - Computer Studies, vol. 43, pp. 625-640, 1995.

[25] T. R. Gruber, "Toward Principles for Design Of Ontologies Used for Knowledge Sharing," in International Workshop of Formal Ontology. Padova, Italy, 1993.

[26] U. Petersen, H. Schärfe, and P. Øhrstrøm, " "Online Course in knowledge representation using conceptual graphs.", Aalborg University - Department of Communication, at http://www.huminf.aau.dk/cg/index.html, 2005.

[27] IMF, Enterprise Restructuring and Work Organization Research Paper, 2001.

[28] Larsen, H., New Forms of Work Organisation, Conference on new forms of work organisation, EU Presidency, Roskilde, Denmark, 2003.

[29] Hayes, P., Saavedra, R., Reichherzer, T., A Collaborative Development Environment for Ontologies, Semantic Integration Workshop, Sanibel Island, 2003.

[30] Hayes, P., Thomas, C., Saavedra, R., Reichherzer, T., Mehrotra, M., Bobrovnikoff, D., Collaborative Knowledge Capture in Ontologies, Third International Conference on Knowledge Capture, Banff, Canada, 2005.

[31] Cañas A., Lake D., Wilson D., "Managing, Mapping, and Manipulating Conceptual Knowledge", 2004.

[32] Garcia A., Sansone S., Rocca-Serra P., Taylor C., Ragan M., "The use of conceptual maps for two ontology developments: nutrigenomics, and a management system for genealogies", 8th Intl. Protégé Conference Protégé, 2005.

[33] Lipnack & Stamps (1997), Virtual Teams: Researching across space, time and organization boundaries with technology. New York: John Willey and Sons.

[34] Knublauch H., Fergerson R.,Noy N., Musen M., "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications", 3rd International Semantic Web Conference, Hiroshima, Japan, 2004.

[35] Sinha, K., Van de Ven, A., Designing Work Within and Between Organizations, Frontiers of Organization Science, Laguna Beach, CA, 2005.

[36] Schwind et al., "Canadian Human Resource Management - A Strategic Approach", Mc Graw-Hill 7th edition, 2005.

[37] A. Bosca, D. Bonino, P. Pellegrino, "OntoSphere: more than a 3D ontology visualization tool," SWAP 2005, the 2nd Italian Semantic Web Workshop, Trento, Italy, December 14-16, 2005, CEUR Workshop Proceedings, ISSN 1613-0073, online http://ceur-ws.org/Vol-166/70.pdf.

[38] A. Bosca, D. Bonino, "OntoSphere3D: a multidimensional visualization tool for ontologies," 17th International Conference on Database and Expert Systems Applications-DEXA 2006, 5th International Workshop on Web Semantics (WebS? 2006), Ed. IEEE.

[39] OntoSphere3D? - e-Lite Reserach Group(2006). Acceded on May, 20th 2006, at: http://ontosphere3d.sourceforge.net.

[40] ALMEIDA, Gladis Maria de Barcellos, OLIVEIRA, Leandro Henrique Mendonça de and ALUISIO, Sandra Maria. A terminologia na era da informática. Cienc. Cult., Apr./June 2006, vol.58, no.2, p.42-45. ISSN 0009-6725.

[41] Ahuja, M., Carley, K.(1998). Network Structure in Virtual Organizations, Jounal of Computer-Mediated Communication. Acceded at http://jcmc.indiana.edu/vol3/issue4/

[46] H. Kern, Principles of Organizational Design Part I, Harris Kern´s Enterprise Computing Institute, 2003.

[42] Peter R.A. Oeij and Noortje M. Wiezer, New work organisation, working conditions and quality of work: towards the flexible firm?, European Foundation for the Improvement of Living and Working Conditions, Dublin, Ireland, 2002, pp. 55-67.

[43] S. Sauter, Brightwell, W., Colligan, M., Hurrell, J., LeGrande, D., Lessin, N., Lippin, R., Lipscomb, J., Murphy, L., Peters, R., Keita, G., Robertson, S., Stellman, J., Swanson, N. e Tetrick, L., The Changing Organization of Work and the Safety and Health of Working People, Knowledge Gaps and Research Directions, National Institute for Occupational Safety and Health, 2002, pp. 1-43.

[44] Dhondt, Steven and Kraan, Karolus and van Sloten, Guurtje. Work organisation, technology and working conditions. European Foundation for the Improvement of Living and Working Conditions, 2002.

[45] Job Design and Work Organization, Accel-Team.com, http://www.accel-team.com/work_design/index.html, 2005.

[46] Bernhard, R., Katzy, Design and Implementation of Virtual Organizations, Thirty-First Annual Hawaii International Conference on System Sciences, IEEE Computer Society, Hawaii 1998, pp. 140.

[47] Langlois, R. (2002). "Modularity in Technology and Organizations." Journal of Economic Behavior & Organization. 49 (2002)(1 (September)): 19-37.

[48] Crossman, A. and L. Lee-Kelley, Trust, commitment and team working: the paradox of virtual organizations. Global Networks, 2004. 4(4): p. 375-390.

[49] Handy, C., Trust and the virtual organization. Harvard Business Reviews, 1995.

[50] Javenpaa, S. and D. Leidner, Communication and Trust in Global Virtual Teams. Organization Science, 1999. 10(6): p. 791-815.

[51] Fox, M., et al., An Organization Ontology for Enterprise Modeling, in Simulating Organizations: Computational Models of Institutions and Groups, M. Prietula, K. Carley, and L. Gasser, Editors. 1998, AAAI/MIT Press: Menlo Park CA. p. 131-152.

[52] Wright, P., et al., Impact of information Technology on Employee Attitudes: A Longitudinal Field Study. 1997, Center for Advanced Human Resouces Studies - Cornell University.

[53] Larsen,K. and C. McInerney?, Preparing to work in the virtual organization. Information & Management, 2002. 39 (6): p.445-456.

[54] Slocum, J, Job Redesign: Improving the Quality of Work Life . Experiential Learning and Simulation, 1981. (3): P.17-36.

[55] Campbell, D. & Pritchard, R., Motivation theory in industrial and organizational psychology, 1976, In M. D. Dunnette (Ed.), Handbook of industrial and organizational psychology. P 63-130. Chicago: Rand McNally.

[56] Robertson, M. MACROERGONOMICS: A WORK SYSTEM DESIGN PERSPECTIVE, SELF-ACE 2001 Conference "Ergonomics for changing work", Montereal, 2001, pp. 67-77.

[57] Alter, S. The work system method for understanding information systems and information system research. Communications of the Association for Information Systems, 9(9):90–104, 2002. http://cais.isworld.org/articles/default.asp?vol=9&art=6

[58] L. Budd, D. Amaral, R. Bergquist, P. Longanbach and A. V. Skiver, Getting Results Through Learning, in H. R. D. Council, ed., Federal Human Resource Development Council, 1997.

[59] J.R. Schermerhorn, Management 8th, John Wiley & Sons, Inc, 2004.

[60] J. Ivancevich, R. Konopaske, M. Matteson, Oeganization Behavior and Management 7 e/d, McGraw?-Hill/Irwin, 2005.

[61] E. S. K. Yu, "Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering," in 3rd IEEE International Symposium on Requirements Engineering (RE'97): IEEE Computer Society. Page(s):226 - 235, 1997.

[62] W. M. Coalition, "Workflow Management Coalition - Terminology & Glossary," Workflow Management Coalition, United Kingdom WFMC-TC-1011, February 1999.

[63] M. Gruninger and M. S. Fox, ""Methodology for the Design and Evaluation of Ontologies"," in Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal,CA, 1995.

[64] M. Fernández López, "Overview Of Methodologies For Building Ontologies," in IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5), Stockolm, Sweden, 1999.

[65] S. Pinto, S., York Sure and Christoph Tempich (2004). OntoEdithttp://dionisio.inescporto.pt/twiki/bin/edit/PHDSALS/OntoEdit?topicparent=PHDSALS.CarlaRevisao empowering SWAP: A case study in supporting DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies (DILIGENT).

[66] Y. Sure and R. Studer, ""On-To-Knowledge Methodology - Employed and Evaluated Version," Institute AIFB, University of Karlsruhe 2001.

[67] B. Frankovič and I. Budinská, "The Role of Ontology in Building of Knowledge Systems for Industrial Applications", in SAMI 2006 - 4th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence, Herlany, Slovakia, 2006.

[68] M. Laclavík, "Ontology and Agent based Approach for Knowledge Management." vol. Ph.D: Institute of Informatics, Slovak Academy of Sciences, 2005, p. 121.

[69] M. Smialek ,L. Balcerek and T. Biedka, "The knowledge-based content management application design methodology" , **ICONS - Intelligent Content Management System** Consortium, **www.icons.rodan.pl**, 2004

[70] D. Jones, M. Bench-Capon, T. J. M. and Vis ser, P. R. S. (1998): Methodologies for Ontology Development, In: Proceedings IT&KNOWS Conference, XV IFIP World Computer Congress, Budapest.

[71] M. Uschold, M. King, M. Moralee, S. Zorgios, Y.: The Enterprise ontology. In Uschold, M., Tate, A., eds.: Special Issue on Putting Ontologies to Use. Volume 13 of The Knowledge Engineering Review. Cambridge University Press (1998) also available from AIAI as AIAI-TR-195.

[72] B. Swartout , R. Pati, K. Knight, and T. Russ, "Toward Distributed Use of Large-Scale Ontologies," in SPK+96 - The 10th Workshop on Knowledge Acquisition, Banff, Canada, 1996.

[73] J. Mylopoulos, "An overview of Knowledge Representation," in Proceedings of the 1980 workshop on Data abstraction, databases and conceptual modeling, Pingree Park, Colorado, United States, 1980, pp. 5-12.

[74] T. Gruber, "Ontology (Computer Science).", acceded in 2007 at http://tomgruber.org/writing/ontology-definition-2007.htm.

[75] H. Kim, "Developing ontologies to enable knowledge management: integrating business process and data driven approaches", AAAI workshop on Bringing knowledge to Business Processes, 20-22 March, 2000.

[76] D. Calvanese, M. Lenzerini, and D. Nardi., "Unifying class-based representation formalisms.", Journal of Artificial Intelligence Research, 11:199-240, 1999.

[77] NSF Initiative, "Common Instrument Middleware Architecture

(CIMA)," acceded in 2007 at www.intrumentmidlleware.org.

[78] G. Tatiana, K. Michael, and V. Veremiev, "TEACHING STRATEGIES AND ONTOLOGIES FOR E-LEARNING," International Journal "Information Theories & Applications", vol. 11, p. 6, 2004.

[79] D. Gašević, V. Devedžić, and D. Djurić, MDA Standards for Ontology Development, in International Conference on Web Engineering (ICWE2004). 2004: Munich, Germany

[80] S. Cranefield and M. Purvis. UML as an Ontology Modelling Language. In Proceedings of the Workshop on Intelligent Information Integration, volume 23 of CEUR Workshop Proceedings, Stockholm, Sweeden, July 1999.

[81] D. Djuric, D. Gasevic, V.Devedzic, "Ontology Modeling and MDA", in *Journal of Object Technology*, vol. 4, no. 1, January-February 2005, pp. 109-128.

[82] P. Hayes, "Knowledge Representation." acceded in 2007 at http://www.cs.umbc.edu/771/papers/hayesp.html.

[83] K. Russell, "Topic maps.", in ComputerWorld, vol. 38, no.40, p.46, 2004

[84] A. Wrightson, " Topic Maps and Knowledge Representation.", 2001, acceded in 2006 at http://www.ontopia.net/topicmaps/materials/kr-tm.html.

[85] J. Robertson, "Where is the Knowledge in a content management system?," in *KM Column*, 2003. http://www.steptwo.com.au/papers/kmc_wherek/index.html.

[86] J. Plotkin, "Conceptual Systems Vs. Physical Systems," 2003, acceded in 2007 at http://ratjed.com/index.php?systems.

[87] Alice Lam "Tacit Knowledge, Organizational Learning and Societal Institutions: An Integrated Framework". Organization Studies. May 2000. FindArticles.com. 25 Dec. 2007. http://findarticles.com/p/articles/mi_m4339/is_3_21/ai_65379605.

[88] A. Seufert, "Towards the Continuously Learning Organization through Knowledge Networking," in *33rd International Conference on System Sciences*, Hawaii, 2000.

[89] B. Chandrasekaran, J. R. Josephson and V. Benjamins, "Ontology of Tasks and Methods," in *KAW'98, Eleventh Workshop on Knowledge Aquisition, Modeling and Management*, Banff, Canada, 1998.

[90] J. Sowa, "Preface to Knowledge Representation," in *Knowledge Representation: Logical, Philosophical, and Computational Foundations* Pacific Grove, CA: Brooks Cole Publishing Co., 2000.

[91] A. Walenstein, "Foundations of cognitive support: toward abstract patterns of usefulness." In: 14th Annual Conference on Design, Specification, and Verification of Interactive Systems (DSV-IS'2002), Rostock, Germany, 2002.

[92] A. Bultman, J. Kuipers, F.Harmelen, "Maintenance of KBS's by Domain Experts: The Holy Grail in Practice", in Thirtheenth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Experts Systems IEA/AIE'00, R. Loganathana and G. Palm and Mali Ed., Springer Verlag, no. 1821, 2000.