Università degli studi di Palermo
Dipartimento di Ingegneria Informatica

Dottorato di Ricerca in

Ingegneria Informatica

A KNOWLEDGE MANAGEMENT SYSTEM

for Organizational Activity Support

Advisor
Prof. Giuseppe Lo Re

Ph.D. Candidate
Patrizia Ribino

Director of Ph.D Course
Prof. Salvatore Gaglio

A thesis submitted for the degree of PhilosophiæDoctor
XXI Ciclo - SSD: ING-INF/05

Ph.D Thesis

Patrizia Ribino

15 February 2011

To my son Michele
I love more than myself.

Sommario

L'attuale realtà imprenditoriale è fortemente influenzata dalla dinamicità dei mercati e dai continui progressi tecnologici. Queste tendenze trovano pratica realizzazione nel modello di organizzazione *flessibile*, che punta a massimizzare la capacità di rispondere con efficacia alle sfide della complessità ambientale. La *conoscenza*, considerata un nuovo fattore di produzione, diventa un elemento chiave nei processi aziendali. Sempre di più, negli ultimi anni è cresciuta la consapevolezza delle imprese dell'effettivo valore di una corretta gestione della conoscenza. L'uso di strumenti propri del *Knowledge Management* nelle organizzazioni è divenuto una pratica comune.

Una caratteristica fondamentale della conoscenza, inoltre, è l'essere strettamente legata alla capacità di compiere azioni. Solo *chi conosce* è infatti capace di prendere le giuste decisioni ed agire di conseguenza. Prendere decisioni riguardanti sistemi complessi (come per esempio, gestire attività organizzative e processi industriali o controllare dispositivi robotici in ambienti dinamici) è un compito che, molto spesso, va oltre le capacità cognitive umane. Questo è dovuto al fatto che le variabili che influenzano il sistema sono, generalmente, soggette a complesse interdipendenze. Per questo motivo predire il risultato finale può risultare piuttosto complicato. Il giudizio di un esperto umano, dunque, si discosta dalla decisione ottima al crescere della complessità dei processi decisionali. In quelle situazioni in cui la precisione è fondamentale, la qualità delle decisioni è molto importante. Una sfida per la comunità scientifica è infatti riuscire ad elaborare tecniche e modelli per superare il limite umano.

Nella tesi presentata vengono affrontati essenzialmente due grossi problemi riguardanti le organizzazioni dell'Information and Communication Technology: il riuso del software e la selezione dei progetti aziendali.

Il riuso del software (*Software Reuse*) non è semplicemente un problema tecnico ma anche e soprattutto un problema di gestione della conoscenza. Il *Riuso* è comunemente definito come un ulteriore utilizzo o un ripetuto uso di un artefatto. Un nuovo prodotto può, quindi, essere realizzato utilizzando una serie di elementi (nel caso in esame, possono essere componenti software o hardware) prodotti in precedenza. Gestire in maniere efficiente la conoscenza aziendale permette, per esempio, di trovare possibili candidati per il riuso da un'apposita repository.

La selezione dei progetti aziendali (*Project Selection*) riguarda la scelta della migliore tra le alternative possibili sulla base di un'analisi costi/benefici. Per decidere quali tra i progetti proposti è più conveniente sviluppare, occorre tenere in considerazione determinati fattori. Ogni progetto, infatti, ha una propria complessità e specifiche caratteristiche, per esempio vantaggi e svantaggi, benefici tangibili e non, costi, impegno di risorse umane e cosi via.

La presente tesi propone un sistema per la gestione della conoscenza che affronta diversi aspetti del *Knowledge Management*, dalla rappresentazione della conoscenza ai processi decisionali (*Decision Making*). In particolare, è mostrato come le ontologie sono applicabili ed effettivi mezzi per supportare la rappresentazione della conoscenza; come sia possibile ricercare componenti software riutilizzabili utilizzando un sistema esperto basato su regole; ed infine come le reti Bayesiane e i sistemi Fuzzy possono integrare conoscenza utile per il supporto alle decisioni in condizioni di incertezza.

Il modello di ragionamento incerto che propongo tiene in considerazioni sia la vaghezza e la soggettività del giudizio umano che l'aleatorietà di alcuni eventi che sono intrisecamente legati al mondo degli affari. Per questo motivo, sono state implementate tecniche di ragionamento fuzzy, tramite le quali il sistema deduce la complessità di un progetto software considerando una serie di fattori che influenzano un progetto. Inoltre, la realizzazione di una rete bayesiana permette di stimare la fattibilità di un dato progetto a partire dall'evidenza derivata dal ragionamento fuzzy.

Il lavoro di ricerca condotto in questi anni di dottorato ed in questa tesi illustrato, ha portato alla realizzazione di *Kromos*, un sistema prodotto in collaborazione con il Dipartimento di Ingegneria Informatica dell'Università di Palermo e di *Sicilia e-Innovazione*, una società della Regione Sicilia finalizzata all'informatizzazione degli uffici della Pubblica Amministrazione.

Abstract

The modern business world is characterized by dynamic markets and continuous technological advances. To cope with these trends, organizations must become more flexible. The *knowledge*, considered as a new factor of production, becomes a key element in business processes. In the last few years, the enterprises awareness about the worth of a correct knowledge management is grown exponentially. The use of Knowledge Management tools within the organization is became a best practice.

The knowledge, additionally, is strictly linked to the capability to perform effective actions. Who knows is able to make a correct decision and to act consequently. Making decisions concerning complex systems (e.g., the management of organizational activities, industrial processes or the control of robotic device in complex environment etc...) often is a task that exceeds human cognitive capabilities. This is because many variables of the system are involved in complex interdependencies and predicting the total outcome may be very difficult. The human intuitive judgment and decision making become far from optimal to grow of complexity of the decision process.

In many situations the quality of decisions is important, overcoming the deficiencies of human judgment is an important issue in the scientific community.

Two main problems concerning ICT enterprises are deeply addressed in this dissertation: Software Reuse and Project Selection.

Software Reuse is not only a technology problem but fundamentally a knowledge management problem. Reuse can be defined as further use or repeated use of an artefact. A new product is created by taking applicable assets from the asset base. A correct knowledge management allows finding candidate assets for reuse from asset base.

Project Selection concerns the choose of the best among alternative proposals on the basis of cost-benefit analysis. In order to decide which of the proposed projects should be selected, a number of factors must be considered. In fact, each project has its own complexity and includes environmental advantages and disadvantages, tangible and intangible benefits, costs, allocation of human and hardware resources and many others.

In this thesis, I present a novel fusion of Artificial Intelligence techniques in order to cope different aspects of knowledge management from knowledge representation to decision making. I show how the ontologies are applicable and effective means for supporting knowledge representation, how to find reusable software components by means of a rule based expert system and how the Bayesian networks and Fuzzy systems can be integrate knowledge to support decision processes under uncertainty.

I proposed a model for uncertainty reasoning, in order to cope not only to the unpredictability of some events that are intrinsically linked to the market environment, but also to overcome the vagueness and subjectivities of human judgments. This model is based on a fuzzy reasoning, which allows evaluating the complexity of an ICT projects unifying the contribution of several factors that complicate a project, and on a Bayesian network able to estimate the feasibility of a project on the basis of the evidence derived from fuzzy reasoning.

This research was applied to the realization of Kromos, a product of collaboration between the Computer Engineering Department of Palermo University and the Sicilian local Government ICT society, *Sicilia e-Innovazione.*

Acknowledgement

I am very grateful to my advisor Prof.Giuseppe Lo Re who gave me the opportunity to conduct this research, for his constant encouragement and guidance in my academic effort.

A special thanks to Prof. Salvatore Gaglio, director of my Ph.D School, for his extremely fruitful technical discussions, for his encouragement and good teachings.

I am indebted to my many Ph.D colleagues for providing me a stimulating and fun environment in which learning and growing and for giving me a healthy dose of distraction from my work. I would like to express my deepest gratitude for their direct and indirect contribution to this dissertation.

I am especially grateful to Antonio Oliveri, Orazio Farruggia, Rosamaria Barone and Liliana Lo Presti for their good company, understanding, patience and affect that I will never forget.

I would like to thank my dear friends, Enza Lo Bue, Rosanna Mirabile and Fabiola Di Bella who are like sisters to me.

I would like to thank my family for the emotional support whose I enjoyed throughout my studies. I thank my mother, Anna, whose morning telephone conversations cheer my days, and my father, Nicolò, who is always in competition with my mother for getting the phone. I thank my husband, Diego, for his love and all the little sacrifices that he made for me in the last years.

Finally, most important of all, I thank my son, Michele, whose birth has enlightened my life.

Contents

List of Figures

Glossary

Best Practice

A process or methodology that has been proven to work well and produce good results, and is therefore recommended as a model. iii

Data

Facts, measurements or observations 6

Database

A base which contains data and information extracted from the government databases, external databases, decision makers. personal databases. It also includes summarized data and information most needed by decision makers. 8

Decision Support System (DSS)

An integrated set of computer tools that allows a decision maker to interact directly with to make unanticipated decisions. 17

Expert System

The branch of artificial intelligence that develops computer programs to emulate a human expert in a specific application domain. 18

Explicit Knowledge

Explicit knowledge can be written down and captured in books, reports and databases. It is relatively easy to transfer from one person to another and does not necessarily require a face-to-face conversation 8

Individual Knowledge

Individual knowledge can be defined simply as knowledge possessed by the individual. This knowledge is most often tacit unless the individual possesses explicit knowledge that is not shared with anyone or any organization other than the individual. 9

Information

Information is interpreted data or data in context. 6

Information and Communication Technology(ICT)

A term encompassing the physical elements of computing, including servers, networks, and desktop computing, that enable digital information to be identified, created, stored, shared, and used. 9

Intangible assets

The non-physical resources and rights of an organization. The intangible assets of an organization are those assets not traditionally accounted for in the financial balance sheets of an organization - assets such as brands, patents, copyrights, knowledge, know-how and customer loyalty. Many organizations now believe that these assets represent such a significant proportion of the value of the organization that they should be included in the balance sheet. 34

Intellectual Capital

Knowledge that is of value to an organization. 11

Know How

The ability to cause a desired result. This may be the most valuable Knowledge element of all. It is usually used to refer to knowledge and experience that has been recorded in context so that it can be used to make decisions and solve problems efficiently. 8

Know What

The known facts. This kind of knowledge is normally closer to information, as it can be more easily codified and/or easily communicated 8

Knower

The Knower is the person or people who are responsible for knowledge, a knowledge domain, or set of documents. The knower or knowledge owner is responsible for keeping the knowledge and information current, relevant, and complete. 6

Knowledge

Knowledge is information in action or information transformed into the capability for effective action. Taking action and building experience turns information into knowledge. 6

Knowledge Base (KB)

A special kind of database for knowledge management, providing the means for the computerized collection, organization, and retrieval of knowledge. Also a collection of data representing related experiences. 18

Knowledge Based System(KBS)

A system that helps decision makers by providing artificial intelligence assistance and knowledge information based on the experience of experts in related fields in decision-making processes and problem-solving processes. 12

Knowledge Management

Systematic approaches to help information and knowledge emerge and flow to the right people, at the right time, in the right context, in the right amount and at the right cost so they can act more efficiently and effectively 5

Knowledge Worker

A staff member whose role relies on his or her ability to find, synthesize, communicate, and apply knowledge. 73

Knowledge-based Economy

A knowledge driven economy is one in which the generation and exploitation of knowledge has come to play the predominant part in the creation of wealth. 9

Kromos

A Knowledge Management System for ICT societies 33

Ontology

The science or study of being. A formal, explicit specification of a shared conceptualization. A shared and common understanding of some domain that can be communicated across people and computers An ontology describes the subject matter using the notions of concepts, instances, relations, functions and axioms. Concepts in the ontology are organized in taxonomies through which inheritance mechanisms can be applied. 13

Operational Knowledge

The knowledge and information required by employees in order to deal effectively with day-to-day problems. 57

Organizational Knowledge

Organizational knowledge is the capability members of an organization have developed to draw distinctions in the process of carrying out their work, in particular concrete contexts, by enacting sets of generalizations whose application depends on historically evolved collective understandings. 9

Project Selection

Choosing the best among alternative proposals on the basis of cost-benefit analysis 59

SECI Model

A model of how knowledge is created through the different interactions between explicit and tacit knowledge. 10

Software Reuse

Software reuse is the process of creating software systems from existing software rather than building software systems from scratch. 49

Tacit Knowledge

Tacit knowledge is essentially impossible to write down. It resides mostly in people's heads. It is hard to transfer from one person to another. It must be gained by dialog and personal experience. 8

[1]

[1] Many definitions come from BSI Knowledge Management Vocabulary

Introduction

Organizations are the pillars of human societies. They can be seen as living entities composed of individuals whom execute a set of activities. These activities are the result of the collective work of the members of the organization in order to achieve an organizational objective. To perform each individual activity it is essential to own appropriate domain knowledge.
In addition nowadays most companies work in complex application contexts which create huge amounts of information. The whole knowledge used by an organization for its functioning forms the so-called "*organizational knowledge*". Moreover, the markets and their competitive pressure are continuously growing, the organizations must react quickly to changes in their domain; information loss may lead to a missed opportunity. In each environment, it is essential an organizational infrastructure that coordinates and supports the activities of the members and manages the intellectual capital of the enterprise - *the knowledge*.

For this reason, organizations are constantly searching for new solutions to adapt to new conditions in order to survive in these increasingly competitive environments. It appears very useful for companies to have awareness of their own information, contained in documents, enterprise processes, acquired experiences and so on. The great amounts of data impose the adoption of new computer-based information systems which enable the storage of structured data and the automation of the information-processing activities of the organization.

Enterprise Knowledge Management classifies knowledge as an organizational asset for competitiveness and survival. It studies methods and computer technologies to increase value, reuse and access of knowledge. KMSs represent information systems which manage organizational knowledge with the purpose of increasing the productivity of knowledge operators.

The work I am presenting in this thesis was carried out during my Ph.D, and it concerns the modeling of a Knowledge Management System to decision support and knowledge representation. My research was applied to the realization of Kromos, a product of collaboration between the Computer Engineering Department of Palermo University and the Sicilian local Government ICT society, *Sicilia e-Innovazione.* In this thesis, I will show how the ontologies are applicable and effective means for supporting knowledge representation, how to find reusable software components by means of a rule based expert system and how the Bayesian networks and Fuzzy systems can be integrate knowledge to support decision processes under uncertainty. In particular I will discuss the creation of an expert system, focusing on two issues concerning ICT enterprises: the Selection of Projects and Software Reuse.

Publications Parts of the work in this thesis have been published in several referred conference proceedings in the fields of Artificial Intelligence and Knowledge Management.

- P. Ribino, A. Oliveri, G. Re, and S. Gaglio, *A Knowledge Management System Based on Ontologies* in Proceedings of the 2009 International Conference on New Trends in Information and Service Science. IEEE Computer Society, 2009, pp.1025 - 1033.
- A. Oliveri, P. Ribino, S. Gaglio, G. L. Re, T. Portuesi, A. L. Corte, and F. Trapani, *Kromos: Ontology based information management for ict societies*, in 4th International Conference on Software and Data Technologies, ICSOFT, 2009, pp.318 - 325.
- P. Ribino, A. Oliveri, G. Re, and S. Gaglio, *A Knowledge Management System Using Bayesian Networks* in Ai*Ia 2009: Advances in Artificial Intelligence: XIth International Conference of the Italian Association for Artificial Intelligence, Reggio Emilia, Italy, December 9-12, 2009,Proceedings. Springer, 2009, p. 446.

Other publications

- Conti, V. and Milici, G. and Ribino, P. and Sorbello, F. and Vitabile, S., *Fuzzy fusion in multimodal biometric systems*, in Proceedings of the 11th international conference, KES 2007 and XVII Italian workshop on neural networks conference on Knowledge-based intelligent information and engineering systems, 2007, pp.108–115.

Structure of the Thesis This thesis is organized in 6 chapters. The first two chapters are a guided tour of relevant literature. The remaining chapters describe the general architecture of the system.

Chapter 1 - Organizational Knowledge Management In this chapter a brief introduction about the Knowledge Management as a new research field is given. Concepts such as knowledge and organizational knowledge are defined. Finally the Knowledge Management Systems and a brief state of the art are argued.

Chapter 2 - Expert Systems for Decision Support This chapter gives a brief overview of the theoretical foundations of the expert systems for decision support. In the several sections will be reviewed the main features of an expert system and its application for decision support. Particular attention is paid to the rule-based expert system and to reasoning with uncertainty. It reviews, specially, the principles underlying the application of the Bayesian probability theory and Fuzzy logic to treatment of uncertainty in decision support systems. Moreover, it exploits the potentiality of the probabilistic tools (such as Bayesian belief networks and influence diagrams), and fuzzy logic tools in order to built decision support systems.

Chapter 3 - Kromos: A KMS for ICT Societies This chapter presents an overview to the features of Kromos Platform, a KMS developed for an ICT company of Sicilian government, and its architecture. In particular we rough out the main components of the system that are the Document Management component and the Decision Support component.

Chapter 4 - An Ontology-based Expert System for Reuse The chapter shows an Ontology-based Expert System to enterprise knowledge reuse that has been integrated in Kromos Platform. In order to deal with the issues of the enterprise knowledge management, the approach, illustrated in this chapter, separates knowledge content into project's knowledge, which describes relevant concepts of software's projecting and into domain's knowledge, which represents the structure of government's offices. In addition, an ontological representation about the enterprise knowledge and an inference engine to extract new knowledge are presented.

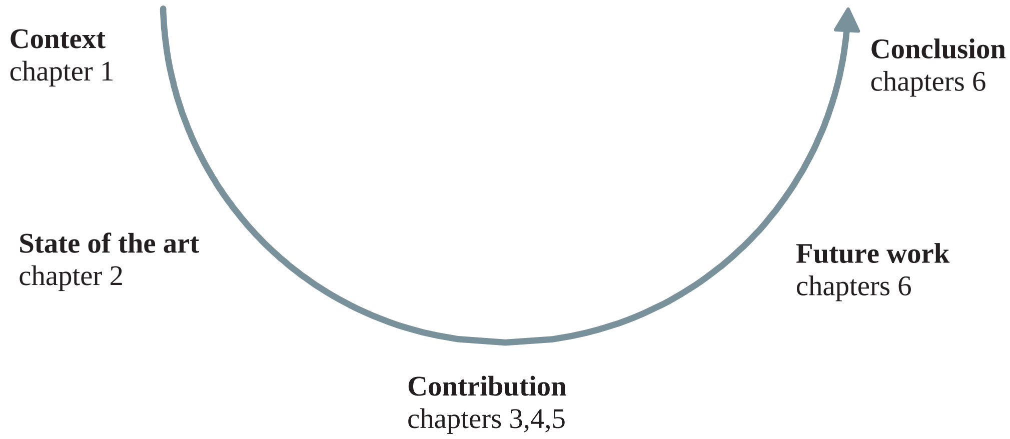


Fig. 1: Structure of the thesis.

Chapter 5 - Optimizing ICT Project Planning Process In this chapter is introduced the Enterprise planning process as an important issue in business environments. When a new project must be developed, the project managers have planning the activities required to reach own goals. In order to achieve these objectives, the managers must be taking into account some issues. In this chapter is given particular attention to optimize Project Planning Process in Information and Communication Technology (ICT) organizations. Especially are taken into account some factors that can influence this process, defining a decision support system that performs two kind of reasoning: fuzzy reasoning and a probabilistic reasoning.

Chapter 6 - Conclusion and Future Works Finally, in this chapter I draw conclusions and show my future directions.

As shown on the figure 1, the dissertation follows a logic progress from generic concerns to specific contributions. I have tried to give different information to make this document reasonably self-consistent for a linear reading. I sincerely hope that the readers of these pages will find useful information to build new knowledge that is, what is more, the high-level goal of this work.

Patrizia Ribino

Chapter 1

Organizational Knowledge Management

In this chapter a brief introduction about the Knowledge Management as a new research field is given. It will be defined concepts as knowledge and organizational knowledge. Finally the Knowledge Management Systems and a brief state of the art are introduced.

1.1 Introduction

The modern business world is characterized by dynamic, changing markets and continuous technological advance. To cope with these trends, organizations must become more flexible. The *knowledge*, considered as a new factor of production, becomes a key factor in business processes.
Knowledge Management(KM) is intended as an integrated approach to reach organizational goals by placing particular attention on knowledge. KM supports and coordinates the creation, transfer and application of individual knowledge in business activities. The major benefits of KM for organizations include:

- Improved capacity for organizational learning and a greater potential for action.
- Knowledge-based value creation processes.
- Increased competitiveness.
- Long-term security and survival.

The following sections present briefly some key concepts related to knowledge management field, introducing principally the main subject at issue, the knowledge and its features.

1.2 Knowledge

In order to fully understand what actually the Knowledge Management is, it is essentially comprehend the concept of knowledge and its features.
Knowledge is defined by Davenport and Prusak (1998) [1] as ‵*a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and information. It originates and is applied in the mind of knowers. In organizations, it often becomes embedded not only in documents or repositories but also in organizational routines, processes, practices, and norms″.*

A common mistake is to identify with the term knowledge what are only *information* or merely *data*. This usually leads to consider the KM as a simple data manager or information system. In order to avoid confusion it is necessary to understand the evolution process that leads from data to become knowledge (Fig:1.1).

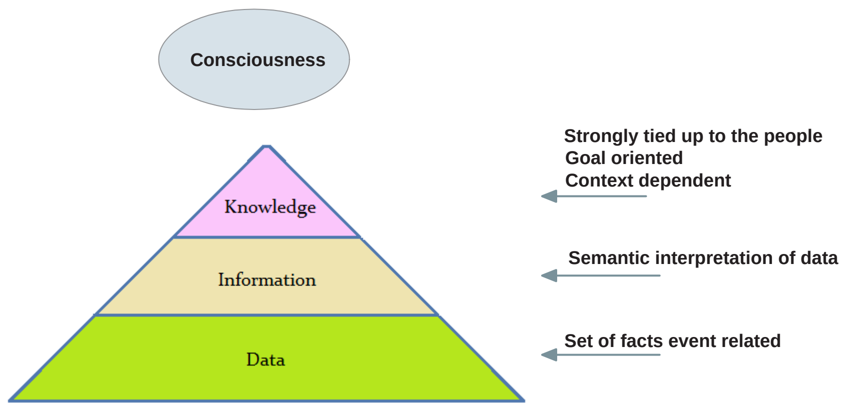


Fig. 1.1: Knowledge pyramid.

The *data* are symbolic representations of facts events related without meaning, so they not constitute a basis for any kind of action. When a datum is linked to a context, the context enriches the datum of a meaning. The data thus turned into *information* . The concept of *knowledge* is much wider than the previous. The knowledge comes from the information and evolves

by means of individuals. It is the result of experience and it is strongly related to expert skills. The shift from information to knowledge involves two processes: the selection of information and integration with the domain of knowledge already gained. The selection process of the information is handled by the objective of the knowledge. So the information are reviewed and retained in proportion to help us to achieve the goal of knowledge. The selection process is followed by the integration process, through which the information is integrated into the knowledge domain of the individual.
Unlike the information, knowledge has implications for decision-making, in other words who *knows* is able to act in certain circumstances within a given context. Human beings use their memory to perform any kind of actions. The stimuli from environment produce cognitive processes in the brain, which change the state of the memory. So, individual knowledge can be defined as the set of all possible memory states linked to possible actions.

Generally, the knowledge shows the following characteristics (Fig:1.2):

1. it is created dynamically through changing of the memory states.

2. it is intrinsically linked to people.

3. it is a prerequisite for human action.

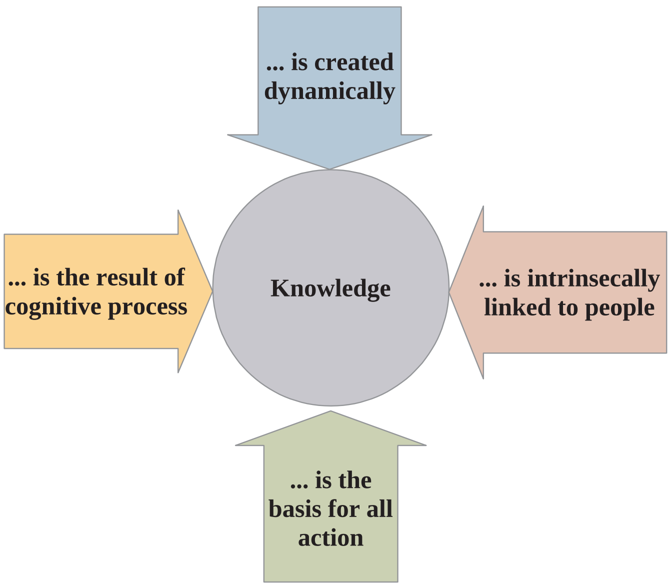


Fig. 1.2: Proprieties of knowledge.

1.2.1 Knowledge Categorization

It is possible to highlight the different features of the knowledge viewing from different perspectives. In particular we can see the knowledge from point of view of psychology, articulability and knowledge holder.

The psychology perspective allows subdividing the knowledge in declarative and procedural knowledge. While *declarative knowledge* refers to facts and objects, *procedural knowledge* is based on the way to perform cognitive processes and actions. Declarative knowledge is also described as knowledge of something or *know what*, while procedural knowledge is also described as process knowledge, or *know-how*.

The articulability perspective look at the awareness of the *knowledge holder*, also called *knower*. In other word, only who is conscious of the owned knowledge will be able to communicate it. People generally know more than they can consciously say. This has the result that the knowledge can be split into explicit and tacit knowledge.

Explicit knowledge is knowledge consciously understood so that the knower can talk about. This kind of knowledge can be also divided into structured, semi-structured and unstructured knowledge. This differentiation is due to the inherent nature of information, which is generally a set of content and structure. The content is the object of the communication. It can be described by means of natural language if we want to represent a text, by means of visual language if we want to represent images or by means of audio language to represent sounds. The structure, instead, is the organization of the communication that makes explicit the main features of what we want to represent. For this reason, commonly we can say that:

- The knowledge is unstructured when it has very little structure and much content, such as a novel.
- The knowledge is structured when it has a lot of structure and content rigidly codified, such as databases .
- The semi-structured knowledge is a compromise of the two previous solutions, such as the Web.

Tacit knowledge , on the other hand, is knowledge that the knower is not aware of. It can be elicited using special observation or interview techniques.

Finally, from point of view of the knowledge holder, the knowledge can be split into individual and collective knowledge.

The *individual knowledge* refers to the knowledge held by one person and it is not context dependent.

The *collective knowledge* is knowledge that is significant in a specific environment (e.g. company, enterprise, etc...). It is composed not only of individual knowledge of the members of the society but also of common knowledge shared by everyone.

1.3 What is the Enterprise Knowledge Management?

Managing knowledge is a hard task, because it is human-based, dynamic and often intangible. In addition, the organizations, as living entities composed of persons whom execute a set of activities, are based on the individual knowledge of their members to create economic value. Capturing and combining these individual memories to form a collective Organizational Knowledge Base is fundamental to survive in a knowledge-based economy . For this reason, organizations needs to introduce processes and technologies that aim to convert individual knowledge into organizational knowledge[2].

An *Organizational Knowledge Base* is founded on two essential components: the individual knowledge of the members of the organization and the framework that connects them. Consequently, one of the major issues of Enterprise Knowledge Management[3] is to promote an organizational culture that supports effective knowledge exchange and reuse. Additionally, because all members of the organization contribute to the Organizational Knowledge Base, it contains knowledge comes from a wide range of different projects, activities and business processes. Consequently, it should be organized in individual knowledge domain.

Enterprise Knowledge Management entails formally managing knowledge resources in order to facilitate access and reuse of knowledge, typically by using advanced information technology and taking into account that knowledge resources can generally include manuals, customer information, expertise, knowledge derived from work processes and so on. For these reason a wide range of technologies can be used to implement KM such as e-mail; databases and data warehouses; group support systems; browsers and search engines; expert and knowledge-based systems and intelligent agents and many others.

1.3.1 The SECI Model

According to Nonaka et.al [4][5][6][7], the evolution and creation process of organizational knowledge is a spiral that crosses two levels: individual and collective level. In each level, the knowledge takes two dimensions: tacit and explicit dimension.

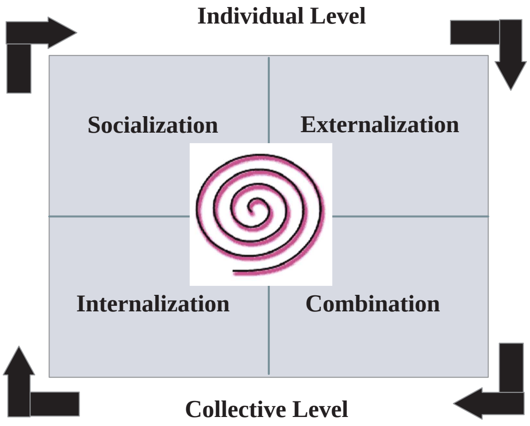


Fig. 1.3: The Seci Model.

The tacit dimension of knowledge, as previously seen, is characterized by the fact that it is strictly linked to the people and therefore hard to formalize and communicate. The tacit knowledge is constituted by experiences, personal skills and know-how and it comprised of both cognitive and technical elements. The cognitive element refers to an individual mental models consisting of mental maps, beliefs, paradigms and viewpoints. The technical component consists of concrete know-how and skills that apply to a specific context.

The explicit dimension of knowledge, on the other hand, is the knowledge that is simply transferable, as previously seen. While in the individual level the knowledge is created by an individual, the collective knowledge is created by the collective actions of a group. Human beings acquire knowledge by actively creating and organizing their own experiences. So, the Knowledge Management requires continuous knowledge conversion processes that permit to capitalize information, experiences and expertise.

Nonaka and Takeuchi [6] defined a dynamic model, the SECI Model, based on the principle that human knowledge is created and enhanced through social interaction so that the tacit knowledge can be transformed in explicit knowledge through four modes of knowledge conversion, as shown in Fig. 1.3:

- *Socialization* (from tacit to tacit) is the process of learning by sharing experiences that creates tacit knowledge as shared mental models and professional skills.

- *Externalization* (from tacit to explicit) is the process of conversion of tacit into explicit knowledge.

- *Combination* (from explicit to explicit) is the process of enriching the available explicit knowledge to produce new bodies of knowledge.
- *Internalization*(from explicit to tacit) is the process of individual learning by repeatedly executing an activity applying some type of explicit knowledge.

The KM is thence any mechanism that can support systematically all these modes of knowledge conversion.

1.4 Knowledge Management System

Knowledge Management consists of techniques that use Information Technology tools for the management of information, and its goal is to improve the efficiency of work teams [8]; it studies methods for making knowledge explicit, and sharing professional expertise and informative resources. There are different approaches to classification of KM issues. Alavi and Leidner [9] group the problems of Knowledge Management, namely storage, creation, transfer and retrieval issues into four classes. Verwijs et al. in [10][11] analyzed different knowledge approaches in business processes, and categorized them as follows:

- **Knowledge storage approach**: knowledge is seen as a product or resource, as something that can be made explicit through codification. Knowledge storage involves issues for obtaining and capturing knowledge from organizational members and/or external sources;
- **Knowledge processes approach**: knowledge as more of a production factor than mere information [12]. Experiences, skills and attitudes are also part of this knowledge. The focus of this approach is on the transfer of knowledge through human interaction. There are many questions about knowledge transfer, one of the most important concerns knowledge flows between provider and knowledge seeker. From the provider's point of view, flow is a selective pull process, but from a seeker's perspective, flow is a selective push process. Thus it is necessary to strike a balance between pull and push processes [13];
- **Learning processes approach**: it involves the relationships between individual and organizational learning. Knowledge management can be seen as assisting organizational learning processes through which new knowledge is created;

- **Intellectual capital approach**: it measures the value of knowledge that resides in an organization [14], this approach focuses on to make explicit the knowledge of an organization, obtaining the transfer of tacit or implicit knowledge to explicit and accessible knowledge formats. This methodology aims to optimize the infrastructure of an organization, and includes training, customers' relationships, ICT, work organization and so on.

A generic Knowledge Management System, supporting the creation and storage of knowledge, gives the opportunity to make data, information and knowledge from different sources readily available. It contains data and documents (explicit knowledge), and can also store tacit knowledge, which is more difficult to express, and includes people's experiences, know-how and expertise. The issue of how to better capitalize and disseminate knowledge is one of the actual priorities in KM. To realize such goals, a KMS can make use of different technologies such as:

- **Document based technologies** for the creation, administration and sharing of different documents (such as doc, pdf, html and so on), managing the explicit knowledge of an organization.
- **Ontology/Taxonomy based technologies** which use ontologies and classification for knowledge representation. Knowledge concepts are frequently arranged in hierarchical structures, typically related by relationships. Such methodologies act on both explicit and tacit knowledge.
- **Artificial Intelligence based technologies** which use particular inference engines to solve peculiar domain problems. The frameworks based on these technologies generally manipulate tacit knowledge.

1.4.1 Knowledge Based System

The Knowledge Base represents the knowledge container, whose relations and concepts are described using an ontological structure of instances suitable for application purposes of a specific domain. A Knowledge-Based System is able to represent specific domain knowledge and to apply it to solve problems through inference processes. The main components of a Knowledge-Based System [15] are the following (Fig. 1.4):

- *The Knowledge Base* is the passive component of a Knowledge-Based System. It plays a role similar to a database in a traditional informative system.

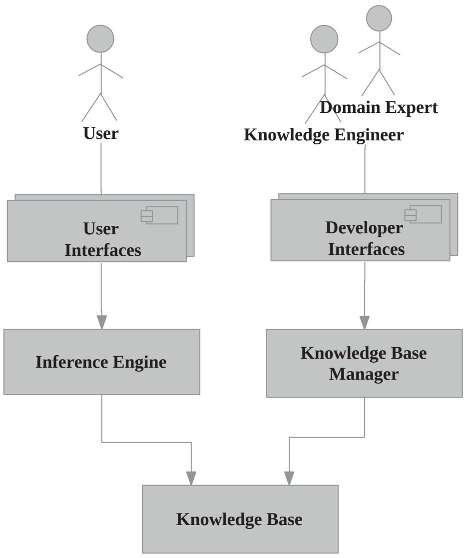


Fig. 1.4: Knowledge Based System Architecture.

- *The Inference Engine* is the core of the system. It uses the Knowledge Base content to derive new knowledge using reasoning techniques.
- *The Knowledge Base Manager* manages coherence and consistency of the information stored in the Knowledge Base.

1.4.2 Ontology for Knowledge Representation

Knowledge Management Systems (KMS) can represent knowledge in both human and machine-readable forms. Human-readable knowledge is typically accessed using browsers or intelligent search engine. Human-readable knowledge is represented using a wide range of approaches in Knowledge Management Systems [3]. But in some case, as the development of an expert system for decision support (that are frequently an integral part of KMSs), knowledge needs to be accessible in machine-readable forms. Therefore, one of the major questions of knowledge management is to obtain a method to represent knowledge in both human and machine-readable forms. To solve this problem, Ontologies [16] are generally used as knowledge containers for KMSs.

An ontology is defined by Gruber [17][18] as *an explicit specification of a conceptualization. ... For knowledge-based systems, what "exists" is exactly that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called*

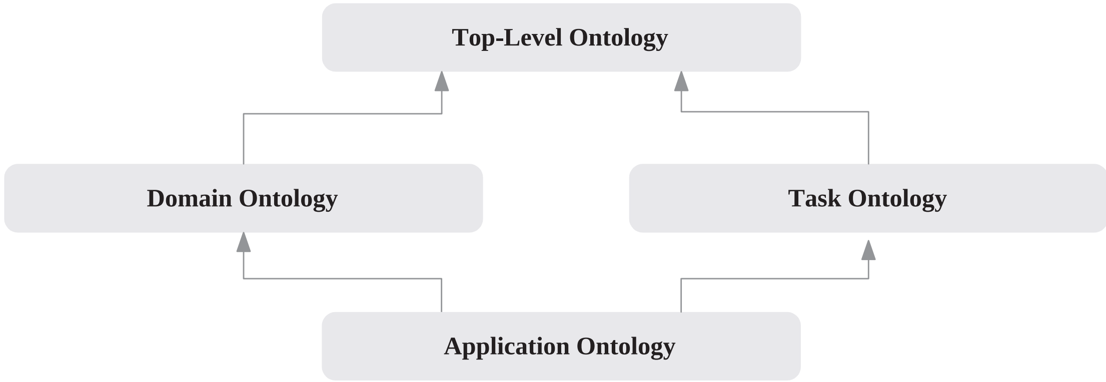


Fig. 1.5: Kind of ontologies.

the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge.

Describing an ontology for a knowledge base system means, thus, modeling a particular domain through the definition of a set of representational terms associated to the entities in the universe of discourse (e.g., classes, relations, etc...) using human-readable text.

According to the level of generality adopted, it is possible to develop different kinds of ontology [19][20][16](Fig. 1.5):

- *Top-level ontologies* describe very general concepts like space, time, matter, object, event, action, etc., which are independent of a particular problem or domain.
- *Domain ontologies and task ontologies* describe, respectively, the vocabulary related to a generic domain (like medicine, or automobiles) or a generic task or activity (like diagnosing or selling), by specializing the terms introduced in the top-level ontology.
- *Application ontologies* describe concepts depending both on a particular domain and task, which are often specializations of both the related ontologies.

In an enterprise KMS, ontology specifications can refer to taxonomies of the tasks or projects that define the knowledge for that system. So ontologies, defining a shared vocabulary, improve communication, search, storage, and representation. Moreover, in the field of enterprise knowledge management, ontologies are useful to the development of corporate knowledge repositories. The challenge is the integration of different kind of knowledge sources, interrelating people, organization and technology.

1.5 Overview of State of the Art

The challenge of Knowledge Management System developers is to create a system of tools to collect, organize, and share common data, documents and individuals expertise. Several researches has examined different aspects of the problem, studying novel knowledge representation models and involving modern Artificial Intelligence techniques to extract new knowledge from that already acquired.

In the last few years, the most important ICT companies have manifested an increasing interest for KM tools, especially for those based on web corporate portals.

A case of a web and ontology-based Knowledge Management System is WAICENT (World Agriculture Information Center) adopted by the United Nations Food and Agriculture Organization. This platform makes FAO's knowledge about food security and agricultural development widely available to users. WAICENT integrates a Decision Support Systems to improve food security through information use and a Document Management System (DMS) to allow people around the world to read and use FAO's documentation [21].

L. Razmerita et al. in [22] proposed a Knowledge Management System based on an ontological model for the user profile. This model takes into account some characteristics, such as users' preferences, competencies, habits, and it adopts semantic web techniques.

Many other KMSs in literature are designed and customized to satisfy the needs of specific firms. Liping Sui in [23] studied the benefits of a Decision Support System within the business management. While D.J. Harvey and R.Holdsworth in [24] observed the advantage of a Knowledge Management System in the aerospace industry. In addition an interesting study conducted by Chun, Sohn and Granados in [25], shows the use of a KMS in an industrial engineering company. After careful observation of the main demands of companies, they have identified what are the features of a KMS so that it can be considered a good investment for an enterprise.

Besides, another KM tool developed is OntoWEDSS [26][27]. OntoWEDSS is essentially a decision support system for wastewaters management with the aim to improve the diagnosis of faulty states of a treatment plant.

Another kinds of KMSs widely studied are Electronic Document Management Systems. These kinds of system deals with the burning issue of the growing volume of documentation produced during working activities. They have evolved from simple systems, which automatically register and archive specific documents, to systems which provide more advanced features in response to the huge amount of documents that organizations must process. Particular attention is paid to the study of tools able to treat information produced by heterogeneous sources. Turk and Bjork [28] categorize a range of basic document management functions which this kind of systems must to fulfill.

Niyogi et al. [29] have developed a knowledge based system for structural analysis of documents that uses a rule-based model. This system segments the digitalized newspaper page using a bottom-up technique. The knowledge rules which comprise the knowledge base define the general characteristics expected of the usual components of a newspaper and the usual relationships between such components in the image.

Nagy et al. [30] used a publication-specific page grammar to describe all legal page formats allowed for a given publication and to segment the document and simultaneously label some layout components with logical classes, defining a hierarchical document model, called a geometric tree, which contains knowledge about hundreds of different business letters.
Another example of Document Based Knowledge Management System is the ANNOTATE [31] which provides mechanisms to support Knowledge Management in federated organizations focusing on documents as repositories of relevant information for knowledge creation and use.

In [32] the authors have introduced a KM process model and its improvement cycle at a high-level applying to reuse processes.

The *Kromos* KMS proposed in this thesis is a framework that not only shows several features that are partially or not addressed by the previous cited work, but also integrates different mechanisms to solve a wide range of problem that a company can encounter in a unique friendly system.

Chapter 2

Expert Systems for Decision Support

This chapter gives a brief overview of the theoretical foundations of the expert systems for decision support. In the following sections will be reviewed the main features of an expert system and its application for decision support. Particular attention is paid to the rule-based expert system and to reasoning with uncertainty. It will be reviewed, specially, the principles underlying the application of the Bayesian probability theory and Fuzzy logic to treatment of uncertainty in decision support systems. Moreover, it will be exploit the potentiality of the probabilistic tools (such as Bayesian belief networks and influence diagrams), and fuzzy logic tools in order to built decision support systems.

2.1 Introduction

Making decisions concerning complex systems (e.g., the management of organizational activities, industrial processes or the control of robotic device in complex environment etc...) often is a task that exceeds human cognitive capabilities. In fact, often is more easy understood how the individual interactions among system's variables occur than predict how the system will react to an external manipulation. This is because many variables are involved in complex interdependencies and predicting the total outcome may be very difficult. The human intuitive judgment and decision making become far from optimal to grow of complexity of the decision process. Because in many situations the quality of decisions is important, overcoming the deficiencies of human judgment is an important issue in the scientific community. For these reasons, various methods have been developed for making rational choices. These methods, often, are enhanced by a variety of techniques originating from information science, cognitive psychology and artificial intelligence. More recently they have been implemented in computer programs, in stand-alone tools and also in integrated computing environments for complex decision making. Such environments are known with the common name of Decision Support Systems (DSSs).

Decision support systems are gaining an increased popularity in various domains, including business, engineering, the military and medicine. They are especially valuable in situations in which the amount of available information is prohibitive for the intuition of an unaided human decision maker and in which uncertainty factors occur. DSSs can aid human cognitive deficiencies by integrating various sources of information, providing intelligent access to relevant knowledge, and aiding the process of structuring decisions. They can also support choices among well-defined alternatives and employ artificial intelligence methods to address heuristically problems that are intractable by formal techniques. Application of decision-making tools in business environments increases productivity, efficiency, and effectiveness. In particular the use of a DSS gives to enterprises a comparative advantage over their competitors, allowing them to make optimal choices for technological processes, planning business operations, logistics, or investments.

A DSS might be used in a great many different ways, this wide range of applications creates substantial variation in its features and capabilities. In the following sections will be illustrated the common features of DSSs and will be introduced the reasoning under uncertainty.

2.2 Expert Systems

The term *Expert Systems* was coined by the academic community, engaged in research into Artificial Intelligence(AI), to describe the use of techniques of AI [33] in practically useful systems. Expert system are intelligent information system that emulate a human expert in a specific application domain [34].
The most successful systems typically provide an expert problem-solving or advice-giving capability and require considerable knowledge and experience in the problem domain.
Some of the most important issues of expert systems are:

- Knowledge representation and Knowledge Base organization;
- General problem solving techniques;
- Human interfacing problems (natural language, spoken language, and other techniques).

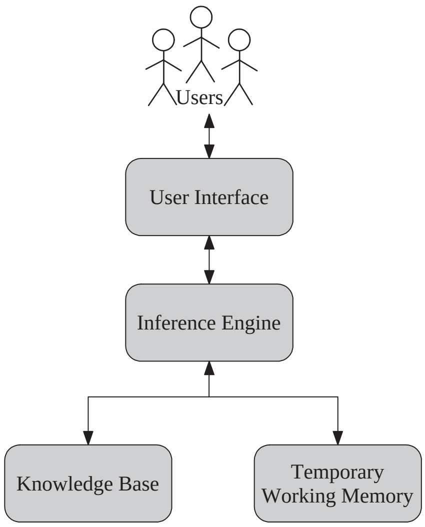


Fig. 2.1: Expert System Architecture.

Expert systems are typically composed of three modules:

1. Knowledge-base.
2. Inference Engine.
3. User Interface.

The Fig. 2.1 shows the components and their interconnections of an expert system. While the role of the user interface is obvious, the definition of Knowledge Base and Inference Engine may require some clarification. The Knowledge Base contains the knowledge about a particular domain; it provides expert-level information necessary to solve problems in a specific domain. This information can be codified in several representation schemes, commonly a set of rules or frames or semantic net or belief network, etc. The inference engine solves problems stated by the user by using the Knowledge Base, and it generates user friendly explanations of the solutions. In addition to its continuous interaction with the Knowledge Base, the inference engine also records the information known about the current problem in a working memory.

The decision making process, instead, can be treated as the selection of a particular alternative from a given set of alternatives which satisfy some given goals. Thus the main issue to be solved is to evaluate the alternatives calculating their utilities.

An expert system for decision support must be based on an appropriate Knowledge Base in order to provide an efficient solution. There are some

definitions of decision support systems, in this thesis it is considered the following: *Decision support systems are interactive, computer-based systems that aid users in judgment and choice activities. They provide data storage and retrieval but enhance the traditional information access and retrieval functions with support for model building and model-based reasoning. They support framing, modeling, and problem solving* [35].
Typical application areas of DSSs are management and planning in business, health care, the military, and any area in which management will encounter complex decision situations. Finally, there are some kinds of expert system based on several types of knowledge representation for example:

- Rule-based Systems in which knowledge is represented by series of rules;
- Frame-based Systems in which knowledge is represented by frames;
- Hybrid Systems in which several approaches are combined, usually rules and frames;
- Custom-made Systems that meet specific need;
- Probabilistic Systems in which the knowledge is affected by uncertainty.

The following sections summarize the main features of three kinds of expert systems for decision support; the first is a deterministic rule-based expert system while the others are expert systems for uncertainty knowledge management.

2.3 Jess - Java Expert System Shell

In the previous section an expert system was defined as a system that can reason about the world and take appropriate actions based on some kind of knowledge about the world. When the knowledge is specified using facts and rules and a reasoner engine operates on them, then an expert system is also called a "rule-based system". In each system a rule usually takes the form "IF <if-part> THEN <then-part>". The if-part of a rule is commonly referred to as its left-hand side, premise or antecedent. Premises contain a collection of conditions that must be satisfied before the rule may be used. The then-part instead, also called right-hand side or consequence, contains a set of actions to be performed when the rule is applied. In particular, Jess (Java Expert System Shell) [36] [37] [38] is an interpreter for the Jess Language, a rule-based language for specifying expert systems. Architecturally, Jess is a production system executing a rule-based program.

The Jess engine can be invoked as an interactive interpreter, where Jess language strings can be typed into a shell and invoked in real-time, or in batch mode, where one or multiple files of Jess code can be executed at once. The Jess engine is implemented in Java, and as well as the shell or interpreter mode, it can also be invoked from Java code at runtime. Jess code is able to call other Java code, or be executed in a Java object.

2.4 Reasoning with Uncertainty

An important issue addressed by the expert system is the uncertainty management [1]. There are several sources of ‟uncertainty ″in particular [39]:

- Information can be unreliable due either to ill-defined domain concepts or to inaccurate data.
- Descriptive languages lack precision due to numerous ambiguities in natural language.
- Inference is sometimes drawn with incomplete information.
- Experts disagree in fact combining the views of multiple experts into a consensus knowledge base is difficult and confused.
- Unpredictable event can occur.

Therefore, an automated expert system should be able to handle these issues, and since uncertainty come from a variety of sources there are several methods to handle uncertain information, in particular is considered Bayesian and Fuzzy method to treat the uncertainty.

2.4.1 Probabilistic Reasoning

Many aspects of our day-to-day life are based on the concept of probability. Often we talk about probability without understand that we making some kind of prediction based on uncertainty knowledge that we have of the world. Conversations about the weather forecasting or who will win the Football League are very common. The concept of probability may be traced back thousands of years to the introduction of the words like maybe, chance, probable and so on, into spoken language. The mathematical theory instead was formulated around 1660 [39].

[1] Part of the theoretical discussion about the uncertainty reasoning is taken from the work of Keung-Chi et.al [39], because in my opinion they give a good overview.

An event's probability is commonly defined as the "proportion of cases in which the given event occurs ", but others interpretation are possible [40]. The three dominant interpretations [41] are:

- **Frequency**: In this interpretation the probability measures the ratio between the occurrences of observations of a specific event in a given experiment, repeated a large number of times under similar conditions, and the number of experiments executed. In other words it measures the percentage of an event's occurrence. This point of view is also called objectivist, because it interprets probability as an objective property of the world.

- **Subjectivist**: In this interpretation the probability measures the confidence that an individual has about a particular proposition's truth. In other words the probability of an event is a measure of personal degree of belief in that event given the person's current state of knowledge. Moreover this perspective does not deny the possibility that two reasonable individuals faced with the same evidence may have different degrees of belief in the same proposition's truth. The Bayesian theory is often used as synonym subjective probability theory.

- **Logical**: In this interpretation the probability measures the extent to which one set of propositions confirms the truth of another. This interpretation is viewed as an extension of logic.

These different probability theories use different inference approach. However, only two main school of probability calculus exist: Pascalian (or conventional) and Baconian (or inductive). Pascalian calculus uses Bayes' rule for belief revision; Baconian calculus uses the rules of logic to prove or disprove hypothesis. Thus, researchers of the field of the objectivistic and subjectivist probability follow Pascalian calculus, instead the supporters of the logical interpretation of probability use Baconian calculus.

For our intent, because we are analyzing expert systems that treat human knowledge, the only appropriate interpretation of probability is as subjective belief. As result, most probabilistic expert system was built by Bayesian researchers.

2.4.1.1 Probability Theory

Let E be an event in the world. The collection of all possible elementary events, Ω, is called the sample space or the event space. The probability of an event A is denoted by p(E), and every probability function p must satisfy three axioms:

1. The probability of any elementary event E in non-negative: $\forall E \in W$: $p(E) \geq 0$.
2. The probability of the entire sample space is one: $p(W) = 1$.
3. Let k events $E_1, E_2, ..., E_k$ mutually exclusive, then the probability that at least one of these events will occur is the sum of the individual probabilities: $p(E_1 \cup E_2 \cup ... \cup E_k) = sum_{i=1}^{k} p(E_i)$ when $E_1 \cap E_2 \cap ... \cap E_k = \varnothing$.

Moreover, the probability of E's complement $(\neg E)$, which contains all of events in Ωexcepts E, is the difference from the probability of the entire sample space and the probability of event E, because E and $(\neg E)$ are event mutually exclusive: $p(\neg E) = 1 - p(E)$.

The Fig. 2.2 shows these properties.

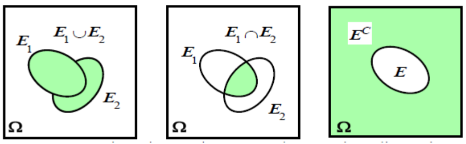


Fig. 2.2: Union, Intersection and Complementation of events.

Additionally, let A, B $\in W$, the probability tat A will occur given that B occurs, written $p(A \mid B)$, is called the conditional probability of A given B. The probability that both A and B will occur, $p(A \cap B)$, is called the joint probability of a and B. By definition, the conditional probability $p(A \mid B)$ equals to the ratio of the joint probability $p(A \cap B)$ to the probability of B:

$$p(A \mid B) = \frac{p(A \cap B)}{p(B)} \tag{2.1}$$

Similarly, the conditional probability of B given A, $p(B \mid A)$, equals

$$p(B \mid A) = \frac{p(A \cap B)}{p(A)} \tag{2.2}$$

From previous equations 2.1 and 2.2 is obtained the Bayes'rule:

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)} \tag{2.3}$$

If we make the assumptions that A and B are independent (in other words one event's occurrence does not affect the other's occurrence), then by definition $p(A \mid B) = p(A)$ and $p(B \mid A) = p(B)$. Moreover if A and B are disjoint events then $p(A \cup B) = p(A) + p(B)$ and $p(A \cap B) = p(A)p(B)$. Additionally, from set theory it is possible write B as $(B \cap A) \cup (B \cap \neg A)$. Since this union is clearly disjoint, then

$$\begin{aligned} p(B) &= p((B \cap A) \cup (B \cap \neg A)) \\ &= p(B \cap A) + p(B \cap \neg A) \end{aligned}$$

$$= p(B \mid A)p(A) + p(B \mid \neg A)p(\neg A) \tag{2.4}$$

Combining equation 2.3 and 2.4 yields

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B \mid A)p(A) + p(B \mid \neg A)p(\neg A)} \tag{2.5}$$

Equation 2.5 lays the groundwork for using probability theory to manage uncertainty, in fact it provides a way of obtaining the conditional probability of A given B from conditional probability of B given A. This relationship allows expert systems to make inference about event of the world. In fact supposing that an expert system is based on these type of rules: *IF H is true THEN E will be observed with probability* p[2]. Thus, if H is observed, by means of this rule an expert system can infer that the probability that event E occurred is p. Instead what happen if the status of event H is unknown, but event e is observed? The previous equation allows to compute the probability that H is true.

The general probability theory can be instantiated into for the probabilistic calculation in expert systems. In fact replacing the term A and B with H and E in equation 2.5 we obtain the following equation that constitute the pillar of the probabilistic expert systems.

$$p(H \mid E) = \frac{p(E \mid H)p(H)}{p(E \mid H)p(H) + p(E \mid \neg H)p(\neg H)} \tag{2.6}$$

In this interpretation H usually represents a hypothesis and E denotes a piece of evidence. Moreover the equation 2.6 suggests to define the prior probability of hypothesis H, p(H), as the probability assigned to H prior to the observation of any evidence. In expert systems, a human expert gives the probabilities associated to the random variables and stores them in a knowledge base. These probabilities include the prior probabilities for all possible hypothesis (p(H)) and the conditional probabilities for observing a piece of evidence given a hypothesis ($p(E \mid H)$). Thus when an individual

[2]This example is taken from [39]

uses this kind of expert system, he gives some information about observations (it is commonly said *He sets some evidences*) and the system computes $p(H_i \mid E_j...E_k)$ for all hypothesis $(H_1...H_m)$ in light of the evidences $(E_j...E_k)$ observed. The probability $p(H_i \mid E_j...E_k)$ is called the *posterior* probability of hypothesis H_i upon observing $E_j, ..., E_k$.
Equation 2.6 refers to an evidence that influenced only one hypothesis. We can generalize it taking into account both multiple hypothesis $(H_1, H_2, ..., H_m)$ and multiple evidences $(E_1, E_2, ..., E_n)$.
Single evidence, multiple (mutually exclusive and exhaustive) hypothesis follow:

$$p(H_i \mid E) = \frac{p(E \mid H_i)p(H_i)}{\sum_{k=1}^{m} p(E \mid H_k)p(H_k)} \tag{2.7}$$

Multiple evidence, multiple(mutually exclusive and exhaustive) hypothesis follow:

$$p(H_i \mid E_1E_2...E_n) = \frac{p(E_1E_2...E_n \mid H_i)p(H_i)}{\sum_{k=1}^{m} p(E_1E_2...E_n \mid H_k)p(H_k)} \tag{2.8}$$

Because this equation requires us to know the conditional probabilities of all possible combination of evidence for all hypothesis, often we assume conditional independence among pieces of evidence given a hypothesis. This reduces equation 2.9 to

$$p(H_i \mid E_1E_2...E_n) = \frac{p(E_1 \mid H_i)p(E_2 \mid H_i)...p(E_n \mid H_i)p(H_i)}{\sum_{k=1}^{m} p(E_1 \mid H_k)p(E_2 \mid H_k)...p(E_n \mid H_k)p(H_k)} \tag{2.9}$$

2.4.1.2 Bayesian Network

Bayesian belief networks (BBNs)[3] introduced by Judea Pearl [42] are a graphical formalism for representing joint probability distributions.
A Bayesian Network (BN) shows probabilistic relationships among variables of given problem conditional on uncertainty constraints. In recent years, BNs have been successfully used in many fields.
A BN is a directed acyclic graph with following properties:

- A set of random variables are network nodes.
- A set of oriented arcs connect couple of nodes and represent the cause/effect relationships.

[3] Also called simply *belief networks*, *belief nets*, or *causal networks*.

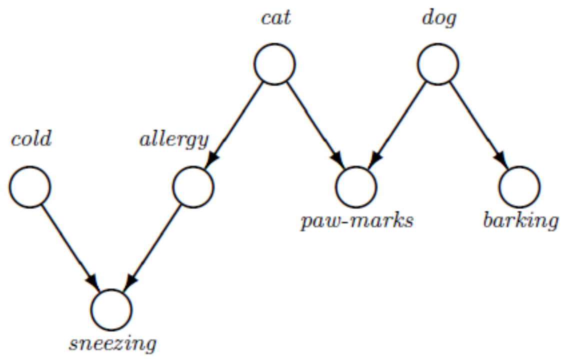


Fig. 2.3: An example of a Bayesian Belief Network.

- There is a probability table for each node, specifying how the probability of each state of the variable depends on the states of its parents.
- Each node without parents has a prior probabilities table of each state.

The relationship between random variables follows clearly Bayes' rule.

Bayesian belief networks provide a convenient and coherent way to represent uncertainty in uncertain models and are increasingly used for representing probabilistic knowledge. An example of a domain encoded in a BBN is given in Fig. 2.3. This BBN describes knowledge about possible causes of sneezing of an individual visiting an unknown house. The individual suffers from frequent colds and is allergic to cats, both of which are possible causes of sneezing. The network models also other relevant facts, such as prints of paw marks on the floor that can provide evidence for presence of a cat. For simplicity, the example assumes that all nodes are binary, that is, have two outcomes (e.g., cat-present and cat-absent).

The example network in Figure 2.3 is singly connected, which means that there is only one path between any two nodes in the network. Singly connected networks are also called polytrees. Networks in which there are nodes connected by multiple paths are called multiply connected. If there were a direct arc between the nodes cat and dog, the network would be multiply connected.

A graph tells us much about the structure of a probabilistic domain but not much about its numerical properties. These are encoded in conditional probability distribution matrices that are associated with the nodes. It is worth noting that there will always be nodes in the network with no predecessors. These nodes are characterized by their prior probability distribution.

Any probability in the joint probability distribution can be determined from these explicitly represented prior and conditional probabilities.

Structural properties of the belief networks along with the matrices of conditional probabilities associated with their nodes allow for probabilistic reasoning within the model. Probabilistic reasoning within a BBN is induced by observing evidence. A node that has been observed is called an evidence node. Observed nodes become instantiated, which means that their outcome is known with certainty. The impact of the evidence can be propagated through the network, modifying the probability distribution of other nodes that are probabilistically related to the evidence. Suppose, that the individual in our example of Figure 2.3 starts sneezing. Node sneezing becomes instantiated. This tells us something about the likelihood of cold and allergy, notably both become more likely. If we later observe paw marks on the floor, node paw-marks becomes instantiated, resulting in an increase of the probability of a cat in the house (also a dog, but this is of less interest for the possible causes of sneezing), and indirectly an increase in the probability of allergy. It is interesting to note that if the directed arcs of a belief network are interpreted causally, the reasoning in the network can be interpreted as diagnostic and causal inference. Diagnostic inference involves reasoning from observable manifestations to hypotheses about what may be causing them, for example, reasoning from sneezing to allergy. Predictive or causal inference involves reasoning from causes to possible manifestations; in our example, it might be predicting allergy after observing a cat.

2.4.1.3 Decision Network

The Decision Network is an extension of Bayesian network by adding a decision nodes represented by rectangles and a utility nodes represented by diamonds (see fig.2.4). The utility nodes are the variables to optimize and the decision nodes are the variables for decision. The decision node defines a finite set of alternatives corresponding to choices to take in order to achieve the desired aim.

Let C=$c_1, ..., c_n$ be a set of mutually exclusive choices. The utility of a choice depends on the state of some variables. Let N be the associated random variables. The objective is reached optimizing the expected utility function (EU) that esteems the preferences among the states of the world. The expected utility associated to the *i-th* choice c_i is given by:

$$EU(c_i) = \sum_{N} U(c_i, N)P(N|c_i) \tag{2.10}$$

where $U(c_i, N)$ is the utility value for the *i-th* choice associated to the

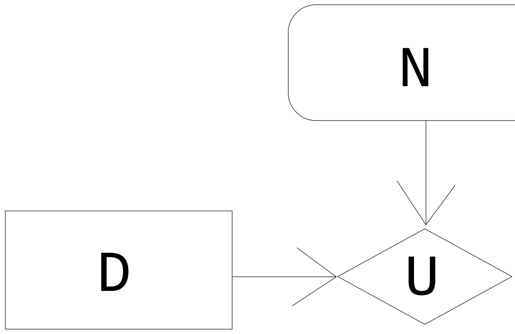


Fig. 2.4: An example of Decision Network.

state of the stochastic variables and $P(N \mid c_i)$ represents the probability of N conditioned by choice.

2.4.2 Fuzzy Reasoning

Fuzzy theory was introduced by Dr. Lofty Zadeh of UC/Berkeley in the 1960's as a means to model the uncertainty of natural language [43]. The difficulties in representing inexact or vague information using classical Boolean logic motivated his research.

In an expert system, the knowledge base contains a great amount of human knowledge, most of which is imprecise and qualitative. When an individual express knowledge about a given problem, he uses linguistic terms such as for example "young ", "very young"or "tall "to describe a state of the world. Moreover apart from imprecision in human knowledge, facts about the world are rarely known with certainty. Thus, when this type of knowledge is encoded into classical Boolean expert system, the *fuzziness* or imprecision is usually lost, and an imprecise information is represented with specific values. In other words Zadeh devised fuzzy theory to express these vague terms. Fuzzy logic replaces the binary logic with multivalued logic.

2.4.2.1 Fuzzy Logic

Fuzzy logic [43] [44] is a super set of conventional (or Boolean) logic and contains similarities and differences with Boolean logic. Fuzzy logic is similar to Boolean logic, in that Boolean logic results are returned by fuzzy logic operations when all fuzzy memberships are restricted to 0 and 1. Fuzzy logic differs from Boolean logic in that it is permissive of natural language queries and is more like human thinking; it is based on degrees of truth.

2.4.2.2 Fuzzy Sets

In classical set theory, a subset U of a set S can be defined as a mapping from the elements of S to the elements of the set 0, 1,

$$U : S - - > 0, 1$$

This mapping may be represented as a set of ordered pairs, with exactly one ordered pair present for each element of S. The first element of the ordered pair is an element of the set S, and the second element is an element of the set 0, 1. The value zero is used to represent non-membership, and the value one is used to represent membership. The truth or falsity of the statement

$$x \textit{ is in } U$$

is determined by finding the ordered pair whose first element is x. The statement is true if the second element of the ordered pair is 1, and the statement is false if it is 0. Similarly, a fuzzy subset F of a set S can be defined as a set of ordered pairs, each with the first element from S, and the second element from the interval [0,1], with exactly one ordered pair present for each element of S. This defines a mapping between elements of the set S and values in the interval [0,1]. The value zero is used to represent complete non-membership, the value one is used to represent complete membership, and values in between are used to represent intermediate *degree of membership*. The set S is referred to as the *universe of discourse* for the fuzzy subset F. Frequently, the mapping is described as a function, the *membership function* of F. The degree to which the statement

$$x \textit{ is in } F$$

is true is determined by finding the ordered pair whose first element is x. The *degree of truth*of the statement is the second element of the ordered pair. In practice, the terms "membership function"and fuzzy subset get used interchangeably. For example suppose that we talk about people and their *tallness*. In this case the set S (the universe of discourse) is the set of people. Let's define a fuzzy subset TALL, which will answer the question *to what degree is person x tall?* Zadeh describes *tall* as a *Linguistic variable*, which represents our cognitive category of "tallness". To each person in the universe of discourse, we have to assign a degree of membership in the fuzzy subset TALL. The easiest way to do this is with a membership function based on the person's height.

Formally, let X be a space of points, with a generic element of X denoted by x. Thus X = x.

A fuzzy set A in X is characterized by a membership function $f_A(x)$ which associates with each point in X a real number in the interval [0,1], with the values of $f_A(x)$ at x representing the "grade of membership" of x in A. Thus, the nearer the value of $f_A(x)$ to unity, the higher the grade of membership of x in A [43].

2.4.2.3 Fuzzy Expert System

A fuzzy expert system [45] is an expert system that uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data. The rules in a fuzzy expert system are usually of a form similar to the following:

IF x is low AND y is high THEN z = medium

where x and y are input variables (names for known data values), z is an output variable (a name for a data value to be computed), low is a membership function (fuzzy subset) defined on x, high is a membership function defined on y, and medium is a membership function defined on z. The antecedent (the rule's premise) describes to what degree the rule applies, while the conclusion (the rule's consequent) assigns a membership function to each of one or more output variables. In a fuzzy expert system, this set of rules is its knowledge base.

The general inference process proceeds in four steps.

1. *Fuzzification*: in this phase the membership functions defined on the input variables are applied to their actual values, to determine the degree of truth for each rule premise.

2. *Inference*: the truth value for the premise of each rule is computed, and applied to the conclusion part of each rule. This results in one fuzzy subset to be assigned to each output variable for each rule. Usually only MIN or PRODUCT is used as inference rule. In MIN inference, the output membership function is clipped off at a height corresponding to the rule premise's computed degree of truth (fuzzy logic AND). In PRODUCT inference, the output membership function is scaled by the rule premise's computed degree of truth.

3. *Composition*: during this phase, all of the fuzzy subsets assigned to each output variable are combined together to form a single fuzzy subset for each output variable. Again, usually MAX or SUM are used. In MAX composition, the combined output fuzzy subset is constructed by

taking the pointwise maximum over all of the fuzzy subsets assigned to variable by the inference rule (fuzzy logic OR). In SUM composition, the combined output fuzzy subset is constructed by taking the pointwise sum over all of the fuzzy subsets assigned to the output variable by the inference rule.

4. *Defuzzification*: finally, this process is used when it is useful to convert the fuzzy output set to a crisp number. There are more defuzzification methods. Two of the more common techniques are the CENTROID and MAXIMUM methods. In the CENTROID method, the crisp value of the output variable is computed by finding the variable value of the center of gravity of the membership function for the fuzzy value. In the MAXIMUM method, one of the variable values at which the fuzzy subset has its maximum truth value is chosen as the crisp value for the output variable.

2.4.3 Probability and Fuzzyness

One of the most controversial issues in uncertainty modeling and information sciences is the relationship between probability theory and fuzzy sets [46]. Fuzzy logic may appear similar to probability and statistics as well. Although, fuzzy logic is different than probability even though the results appear similar.

The **Probability Theory** is essentially based on a bivalent logic. In fact, when we talk about probabilities we simply assign to a given event a number between zero and one. This number represents the eventuality that the given event occurs, but the event's essence is bivalent: or is happening or not happening, there are not other possibilities. The probabilities involved in any situation where we are faced with uncertainty due to lack data, information, etc. The events do not have a deterministic nature but they are governed by the randomness that prevents their prior determination.

In **Fuzzy Logic** the world is characterized by complete vagueness and uncertainty. More information helps us to determine the fuzziness whereby the sets overlap, in other words the fuzzy logic eliminates the boundaries that mark where a thing ceases to be *that* thing. The probability instead becomes less significant when the *known data* become more numerous. In fuzzy logic objects belong, and at the same time, do not belong to a set. For each element is attributed a value, always between zero and one, which expresses its membership of a particular fuzzy set. This value is not a probability, it does not refer to the event occurs or not, but it is the measure of a deterministic fact, but "vague" to the same time, that it is not responding to a bivalent nature but polyvalent, with infinite degrees of membership between 0 and 1. The probability statement, "There is a 70% chance that Bill is tall"supposes that Bill is either tall or he is not. The fuzzy logic statement, "Bill's degree of membership in the set of tall people is .80"supposes that Bill is rather tall. The fuzzy logic answer determines not only the set which Bill belongs, but also to what degree he is a member. There are no probability statements that pertain to fuzzy logic. Fuzzy logic deals with the degree of membership.

Chapter 3

Kromos: A Knowledge Management System for ICT societies

In this chapter an overview to the features of Kromos Platform, a KMS developed for an ICT company of Sicilian government, and its architecture is given. In particular the main components of the system that are the Document Management component and the Decision Support component are explained. The second component will be fully illustrated in the next chapters, because it constitutes the core of my Ph.D work.

3.1 Introduction

The work carried out during my Ph.D was applied to the realization of *Kromos*, a Knowledge Management System for business environments. The objective of Kromos is to collect, share and reuse different sort of knowledge (such as documents, experiences, common knowledge and so on). Kromos can be considered as an ontology-based Knowledge Management System founded on two different ontological models: a domain and project ontology. The former holds concepts and relations about the domain of interest (a business environment) such as processes, activities, offices, etc. The latter, on the other hand, represents notions about business products of the organization, that are, in our case of study, ICT projects.

The main components of Kromos are two subsystems: a Document Management Subsystem and a Decision Support Subsystem. The first one analyses documents extracting information about their structure and content. It includes an Information Retrieval Engine that allows for the insertion, indexing and retrieval of documents. The second one, instead, provides functionalities to assist enterprise managers during their activities emulating human logic in decision making.

The design and development of the framework was a product of collaboration between the Computer Engineering Department (DINFO) of Palermo University and the Sicilian local Government ICT society, Sicilia e-Innovazione. This agreement gave us the chance to verify the ontological model and to test our prototype of general-purpose Knowledge Management System in a real business environment.

3.2 Requirements Analysis in Business Environments

Using knowledge in a business environment is not necessarily about thinking new products or new ways of selling them. The challenge is to manage these intangible assets in a coherent and productive way. In fact, useful and important knowledge already exists into organization, it can be found in:

- the experience of the members of the organization;
- the designs and processes for goods and services;
- files or documents;
- plans for future activities, such as ideas for new products or services

The main function of Knowledge Management is to make knowledge accessible to all those who can benefit from its application. To do this, it is necessary have an organizational framework which disseminates the knowledge to individual workers through a friendly access to the organization's knowledge repository. Furthermore, if this framework would offer the possibility to assist employees to decision making process using the organizational know-how and market forecast, the organization would become more competitive.

In the case of study at issue, I have analyzed the situation of some government institutions and particularly of an ICT society, which main task was to develop ICT projects in order to automate processes of the government offices. In this context it was noted that:

- the lack of an intelligent system, to manage the great amount of knowledge that they produce (such as documents, technical reports, letters, etc...) causes a great waste of time to find a particular file;

- the lack of an automatic information retrieval forces the employees to read a huge amount of documents to find the information they needed;
- the absence of communication and knowledge sharing among developers causes a restricted view of corporate know-how;
- a program manager operates without any kind of support in decision making processes.

For these reasons, agree with the program manager of the company, the adoption of a system that effectively supports the workers of government offices or the operators of public administration in business activities, and the use of smart engines, which can support workers in decision making, will improve organizational performance. A KMS can provide:

- *A* knowledge collection, that is the capacity to obtain data and information from different operators and, for instance, from different kind of sources;
- *A* document manager to an efficient storage and retrieval of documents;
- *A* decision making support, that is the capability of suggest actions under particular circumstances such as uncertainty conditions.

In addition, a KMS can use several techniques to achieve such goals. For example, intelligent agents or reasoners could be applied to analyze the collected data and to infer new knowledge, data mining and knowledge discovery techniques could also be employed to look for data stored inside the organization's knowledge repositories, and so on.
For the needs of the case in point, it was realized:

1. an ontology-based model for knowledge representation;
2. a system to extract information from files analyzing their structure and content providing also an automatic indexing system and query engines to collect and retrieve documents;
3. a decision support system composed by two kind of expert system to address two different problems. The first is the improvement of knowledge reuse and the second is making decisions under uncertainty factors.

3.3 Kromos Framework

The Knowledge Management System developed is called *Kromos*(the logo of the system is shown in Fig. 3.1. Kromos[47][48][49] is a knowledge management system which manages organizational knowledge with the purpose of increasing the productivity of the ICT company operators and improving the knowledge access of government institution to public administration operators. This framework is the product of collaboration between the Computer Engineering Department of Palermo University and the Sicilian local Government ICT company Sicilia-e-Innovazione [50]. It proposes an ontology-based knowledge management system capable of modeling the structure of the public administration offices and the ICT company's projects. In addition, it is able to discover which processes can be automatized and which projects can be reused or developed ad hoc and to suggest which actions to take in uncertain circumstances.

The system consists of a Document Management Engine, which deals with collecting, extraction and searching of knowledge, and an Expert System for Decision Support to elaborate specific information in order to help users during their working activities, emulating human logic and reasoning. The main goal of this work is the development of a complete system for managing data and information, that exploits a new approach with a separation between knowledge representation and knowledge management, so that change in the infrastructure of concepts in the knowledge base, done by the expert of the domain, does not influence the inference mechanisms and logical reasoning processes.

Fig. 3.1: Logo of the Kromos Knowledge Management System.

3.3.1 Case of Study

Sicilia e-Innovazione is the research and development company of the Sicilian Local Government, which operates in the ICT field, representing one of the most interesting settings in order to acquire experiences for the development of a Knowledge Management System. The company was created in June 2006 and since it carried out a lot of projects, many of them to automate public administration processes.

The company was founded to realize, first of all, an integrated unique data transmission platform for the entire Government structure, called *Piattaforma Telematica Integrata(PTI)*, that can connect every office in the Sicilian territory.
In the first years, hundreds of other projects have been created in order to modernize and make more efficient the Sicilian Public Administration. These projects involve at least the 70%of services of the Sicilian public administration. The real estimation of the number of services to be automated cannot be actually done. Each project is described and documented by several technical reports and documents, representing a source of precious knowledge. Each document contains an explanation of the technology used with the description of the system, financing sources, responsibility information, technical and developing choices, and the processes to be automated. The number of pages that compose a generic technical report fluctuates from some dozens to approximately 1000 pages for each project; 70 GB of memory is actually used to store all these technical reports. These documents are generally a complex set of pdf, doc, code and generic writings used and produced during projecting and development. Technical writing translates complex technical concepts and instructions into simple language to enable users to perform a specific task in a specific way.

All this knowledge is precious and could be used in several different ways, for instance, to estimate costs and necessary resources for new projects, to find similarities in different contexts, to increase the code reuse. When Kromos project started, the company was organized as a collection of atomic groups, each working independently by the others. Different teams replied the same development activities, so reuse was minimal, there was no effort to organize and share experiences and knowledge acquired by previous projects, and there was no use of any technical tool (such as search engines) to search and compare different projects. This situation produced knowledge islands not shared among several teams, a limited ability to learn from existing knowledge and no unified vision. This is due to several reasons such as the novelty of the company mission, the heterogeneity of data sources, the difficulty of adopting traditional tools, and the involvement of different

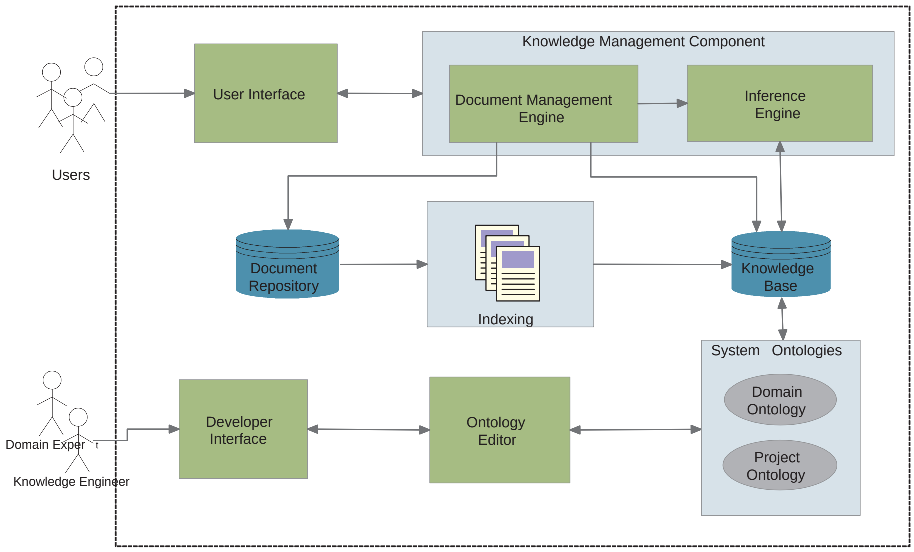


Fig. 3.2: Architecture of Kromos framework.

users. Moreover, previously the start of the project, the problem of data and knowledge management had received small attention. In that period, only few documents on business processes and organization of government offices were produced, while a huge amount of technical documents were generated and stored without any management organization.

All these reasons make this company an ideal scenario where experimenting a process of knowledge engineering, in order to promote the adoption of a Knowledge Management System as a tool for sharing the already acquired knowledge, for the production of the new one and to support any kind of activities.

3.3.2 Kromos Architecture

Kromos is an ontology-based system of knowledge management with the aim of optimizing business processes for creating and managing ICT projects for government offices. The system was designed with the following features (Fig. 3.2):

- specific domain knowledge to build a knowledge base;
- reasoning ability performed by a complex expert system;
- advanced techniques of document and information retrieval.

Kromos uses a knowledge base, to store ontological instances of domain and projects concepts, and a repository of documents to store project documents, used by the Document Management subsystem as sources of data and information to be organized by the use of the Knowledge Base. Knowledge is created through codification and exploitation; codification refers to the collection of tacit knowledge into the Knowledge Base. This make this intangible knowledge comprehensible and accessible to users using a friendly interface for collaborating insertion of data about software projects and offices structure; exploitation in Kromos refers to the collection of data from document, using emerging technologies to provide enhanced document analysis and understanding capabilities for the automatic extraction of data in a Document Management subsystem.

Kromos is, additionally, provided of a reasoning system that allows for complex ontological querying and a decision making support to enterprise planning processes and knowledge reuse. In the next sections of this chapter is given an overview of the Kromos system and its principal elements, while more details about the reasoning mechanism will be discussed in Chapters 4 and 5.

3.4 Knowledge Representation and Storage

3.4.1 Kromos Knowledge Base

Knowledge-based systems [51] are computer programs that achieve expert competence to solve tasks about a specific domain. Their main objective is, thus, to emulate human logic and reasoning (e.g., [52]). KBS are based on a coded human knowledge represented through a mathematical model. The general architecture of a Knowledge Base System, as we have seen in Chapter 1, is composed by following main components: a knowledge container (the knowledge base), an inference engine and a user interface. The knowledge base and the inference engine are the core of the Knowledge Base System; the former is an active database with some kind of formal knowledge; the inference engine is a mechanism that infers new knowledge from knowledge base.

The main issue in order to build a knowledge base for a particular domain is to find a formal model that allows a description of the concepts that characterize the environment. An efficient formalism to knowledge representation is represented by the ontologies. In the Knowledge Base of Kromos, the relations and concepts are described using an ontological structure of instances in order to collect and manage knowledge. Ontologies of the proposed system are performed through Frames [53] and built using Protégé [54], a free and open source platform developed by Stanford University, that supports frame-based ontologies according to the Open Knowledge Base Connectivity Protocol (OKBC) [55]. In a frame-based model, ontology is composed by:

- a set of classes, hierarchically organized to describe the domain concepts;
- a set of slots for the classes, which describes properties and relations between concepts;
- a set of class instances, examples of concept with their specific values and properties.

The use of such ontological model transforms abstract concepts into logical descriptions.

3.4.2 Kromos Ontologies

Kromos knowledge representation model is based on two kind of ontology: the former is the description of the knowledge relevant to the practice of software projecting and the latter is the definition of the government offices structure. It was used Protégé environment to create a graphical representation of the concepts of the domain, and to describe the constraints on relations and concepts. The ontology construction was started by looking for the structure of the Sicilian Government Offices and the information that have to be collected and used during the development of a software project, answering the question about what concepts and knowledge have to be captured to be used. It resulted in a set of concepts divided in two different ontological sub-models, the former, called *Domain Ontology* that represents the domain of action of Kromos Knowledge Management System, the latter, called *Project Ontology*, that designs concepts about the projects developed in the ICT companies. The two sub-models will be described in more details in next two subsections.

3.4.2.1 The Domain Ontology

The Domain Ontology is one of two sub-ontologies corresponding to the Government offices structure and activities. Knowledge about the organization chart is difficult to collect in a dynamic business environment such the one we observed.

The competency questions for the Domain sub-ontology are: what are the main type of offices and structures of the Sicilian Government organization chart? How do they relate to each other? Which information define an office? What are their main activities and responsibilities? Which functionality of these offices are applied using software systems?

To answer these questions is necessary the decomposition of the organizational chart in a set of concepts and relations like depicted in Fig. 4.1.

Modeling knowledge about the government offices world required some assumptions about its structure and activities, as well as about the nature of the "observer" expected to use, understand, and rely on the model; in order to keep the ontology easy to understand, only a few concepts from the government offices' domain are collected.

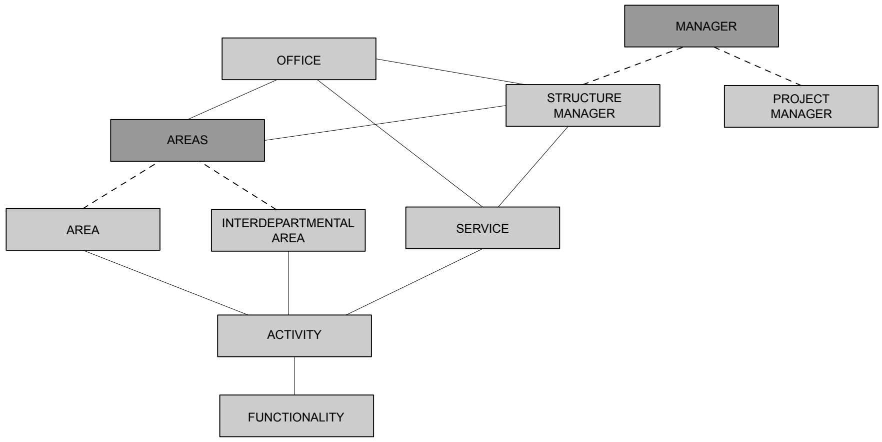


Fig. 3.3: Domain ontology of Kromos platform.

3.4.2.2 The Project Ontology

In the Kromos Project Ontology definition we are interested to organize concepts from the project planning and development environment, to be used during reasoning phase for decision support, or to answer for information about past working activities in ICT companies. Possible competency questions are: what are the features and characteristics that define in a complete and consistent way the project planning and development phases? Who will submit them? What are the possible sources of that kind of information? What are the activities performed during development? What the responsible? What does a Project Manager or a Program Manager need to perform his working activity? What do they produce when they end the development? A project originates from a request submitted by a client or a group of clients. The requests from clients are classified either describing the problem detected by the user, and describing the new requirements to be satisfied. A list of information have to be used to identify the project, like the project identification code used, the name adopted by the working team to identify it, the responsible that will supervise; others are necessary to describe the resources used, like financial, software, hardware and human resources; others refer to the office that will use the product; and so on.

The resulted Kromos Project Ontology is useful to describe ICT company projects; it maps the structure of the project components containing semi-structured explicit knowledge (Fig. 4.2).

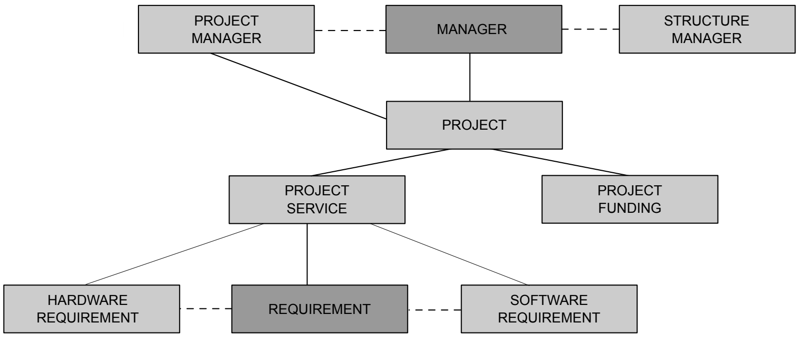


Fig. 3.4: Project ontology of Kromos platform.

3.5 Document Management Subsystem

The Document Management Subsystem[1] provided by Kromos is a module capable of pre-processing documents, analyzing and extracting the structural information about logical units layout, and retrieving data from the blocks enriching the Kromos Knowledge Base. Moreover, it provides systems for the indexing of texts and search engines, to insert and search documents in the file system.

In ICT companies the volume of documents produced during working activities grows rapidly, collecting them in traditional forms becomes almost impractical and also searching them without automatic search engines is a great waste of time. Documents contain most of the information about projects, functionalities, people involved and so on. In addition to the techniques and systems for Document Analysis, Understanding and Extraction, Kromos integrates an Information Retrieval Engine adapted for the insertion,

[1]Implemented in collaboration with Eng.Antonio Oliveri, contacts at: oliveri@dinfo.unipa.it.

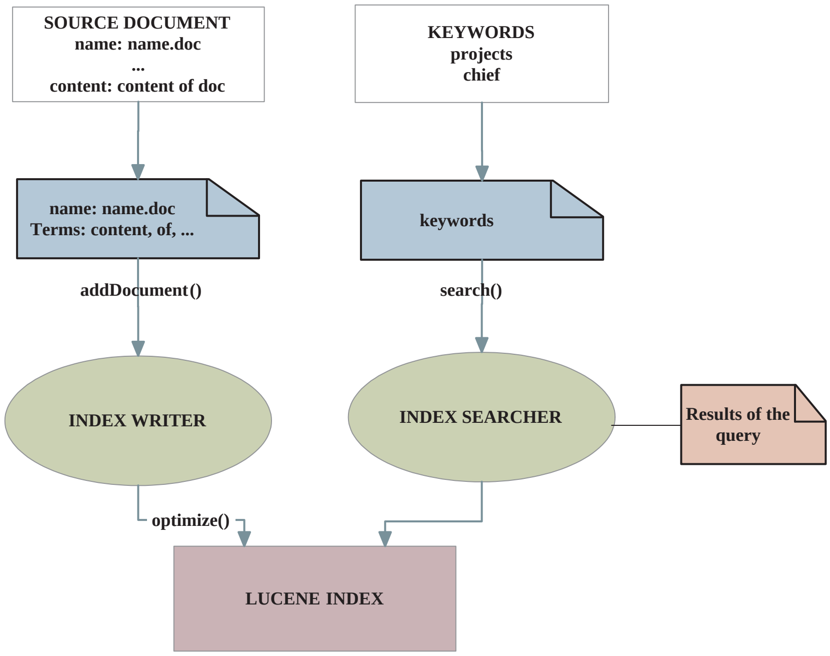


Fig. 3.5: Document management subsystem.

indexing and retrieval of documents in different formats, based on Apache Lucene.

Lucene [56] is an open-source, high performance, scalable, full-featured text search engine and information retrieval library written in Java and suitable for any application requiring a full text search, through which any piece of data convertible to a textual format can be indexed and made searchable.

The main functionality of the indexing and querying engines of Kromos Document Management Subsystem are: pre-processing of documents and their content to obtain a text representation without any lexical or semantic redundancy; document indexing to store information about files in an ordered structure to be used for the search phase; searching for documents using keywords, calculating the degree of satisfaction of the requirements expressed in the query (see Fig. 3.5).

3.6 Decision Support Subsystem

Kromos Knowledge Management System is able to integrate both decision support and collaborative processing functions, not only connecting decision-makers to all the available digital information or assisting top managers to choose the right products for their enterprise, but also helping general workers to accomplish their business goals. The main goal of an Expert System is finding solutions that usually need the intervention of specifically skilled people. The goal is to incorporate implicit knowledge about the specific field in a computational model.

The Expert System prototype for the Kromos platform is a fusion of a two kind expert system, the first one is a rule-based system while the second one is reasoner that makes decision in uncertainty condition. The rule-based system is developed to improve the knowledge reuse, and it have been developed in Jess (Java Expert System Shell) [36], that can be used for reasoning in different knowledge base contents, adapting rules to different kinds of domains. Instead the second is realized to support the program manager into planning process of projects, because this process involves several factor of uncertainty.

In next subsections we will present different use cases of decisional processes in which our Expert System can offer support. In next chapters we explain fully their features.

3.6.1 Use of Decision Support System in Project Planning

The first use case in which our Expert System can offer support is the Project Planning Process. During planning of company projects there are many different constraints to be considered in order to improve enterprise yields and avoid wasting resources. Planning is a process for the definition of a future goal, the activities to exploit in order to reach that objective, and all the resources to be used to complete these activities. The planning process has to identify business components directly connected to the real progress of business activities, measuring their impacts and connected benefits, and to analyze the investment policy.

3.6.2 Evaluation of Project Functionality Reuse

The second use case is the decisional process of project reuse, which is the analysis of government requirements and already developed projects to find reusable components. Sicilia e-Innovazione, the organization we are analyzing, produces a great amount of ICT products to automatize district processes. A process is a set of interrelated activities, grouped in phases. Therefore each project is composed by a set of components, each supporting a single phase of the entire process. Different district processes could have certain phases in common, so that the organization could choose to reuse some components taken from other projects during automation activities, in order to reduce developmental costs.

3.7 Kromos Platform Interface

In the institutional world, the Knowledge Portal's purpose is to display and supply business-specific information, in a certain context, helping users of corporate Knowledge Management Systems to find the information they need to face their competitors. The corporate Knowledge Management Portal is considered an evolution from Intranets, incorporating, to this technology, new tools that enable identification, capture, storage, retrieval and distribution of great amounts of information from multiple internal and external sources, useful for enterprise individuals and teams.

The aim of knowledge portals is to make knowledge accessible to users and to allow its exchange. The proposed system implements the typical client-server paradigm using JSP and Java servlets [57]. An essential feature of a knowledge management portal is the addition of new information and/or the updating of old information in an easy way. Thus, information can come from many different sources. With a graphical interface the users can select different application areas. Nevertheless, presentations of, and queries for, information contents must be allowed in many ways that need to be independent from the way that information was provided originally. The Knowledge Management System must remain adaptable to the information sources contributed by its providers.

Kromos system can be considered as divided in three different macro-areas: Public Administration, Process Management and Project Managements. The first allows the user to manage information about government offices and interacts with domain ontology. The second enables the user to organize new processes in activities that can be potentially automated. The first two areas involve more ontology domain formalization and building processes; the third allows the Project Managers to access the system functionalities. Using that area the user can create complex queries for the Expert System.
Next figures show some screenshots of the different views of Kromos platform.

3.8 Simulation Results and Discussion

The experience carried out during the process of development in the bosom of Sicilia e-Innovazione gave us the chance to verify the ontological represen-

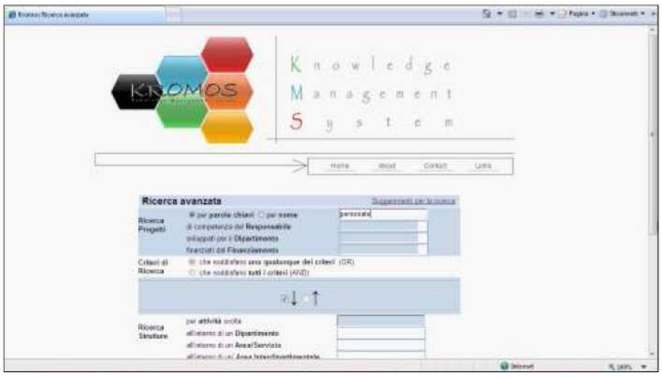


Fig. 3.6: Decision support subsystem interface.

tation of a specific domain, and to test our general purpose KMS system in a real applicative environment. Our working activity in the firm consisted of two phases: the first one, after the acquisition of specific knowledge through the observation of business activities and processes, allowed us to build an ontological model for the domain representation; the latter was centered on the setting of Kromos reasoner and document management services to the company's needs. In the meanwhile some new requirements emerged and that permitted us to refine our system with new features in order to improve Knowledge accessibility to different skilled users.

This experience provided all the information to comprehend the importance of a correct and efficient knowledge resources, people and organization management; to estimate the efficacy of adopted technical solutions, their advantages and disadvantages; and to evaluate needs satisfaction degree using these solutions. We can anticipate here that the adoption of Kromos for the data management by Project and Program Managers of Sicilia e-Innovazione evidenced the advantages of a correct knowledge management in every organization, emphasizing the importance of sharing and reusing information by differently skilled workers. During use of the system, more information was collected and ontology structure changed and grew, but the functionalities of the Expert System did not require any modification. The web based interface, hiding the complexity of the system's functionalities, gave the platform a characteristic of usability, so that the collection of data and documents and the sharing of information seemed to be incremented.

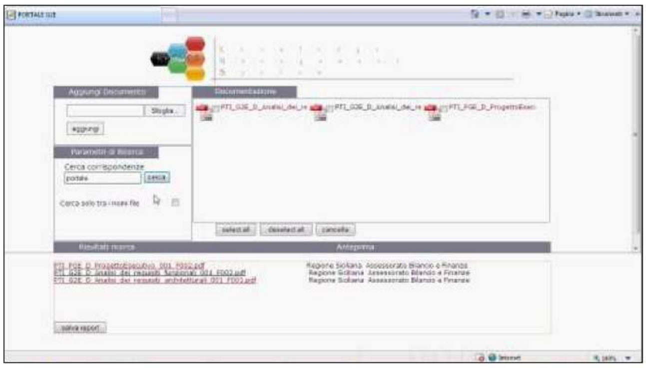


Fig. 3.7: Kromos Document Management Subsystem interface. The image shows the ”Portale G2E” project technical documentation. User can add more docs that are analyzed to extract information and indexed to be organized, and can also search for documents using the search engine. Results compare at the bottom of the screen.

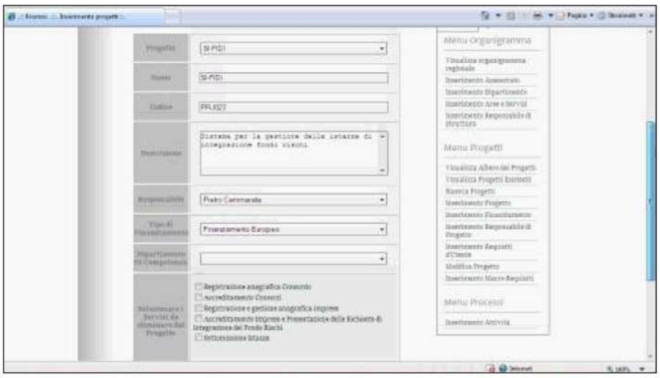
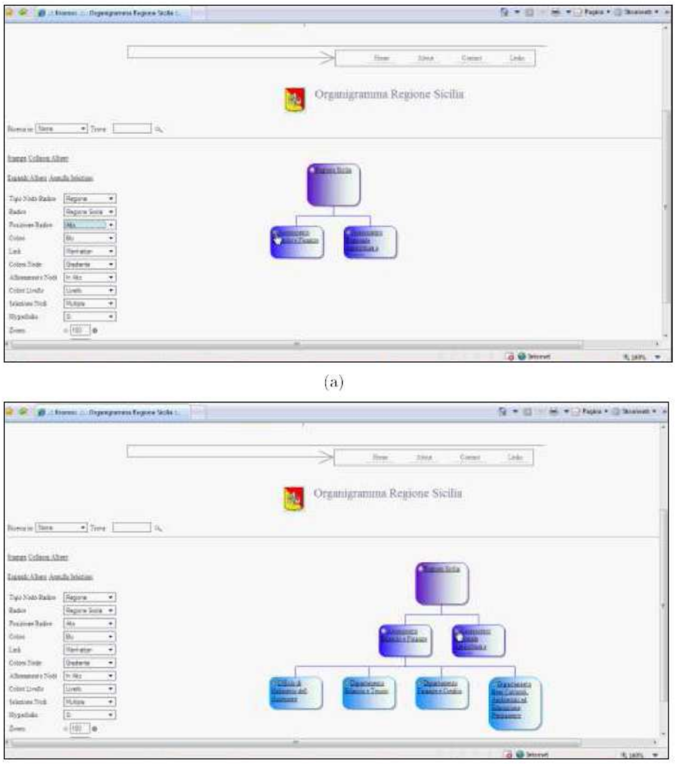


Fig. 3.8: This interface allow for insertion of new ontology instances.



(a)

(b)

Fig. 3.9: Navigable graphical view of the Government offices organization chart. Clicking on a node it is expanded to move to the second level of detail.

Chapter 4

An Ontology-based Expert System for Knowledge Reuse

This chapter presents an Ontology-based Expert System to enterprise knowledge reuse, which has been integrated in Kromos Platform. In order to deal with the issues of the enterprise knowledge management, the our approach separates knowledge content into project's knowledge, that describes relevant concepts of software's projecting, and into domain's knowledge, that represents the structure of government's offices. We give an ontological representation about the enterprise knowledge, and we use an inference engine to extract new knowledge. This system integrates Protégé, to build the Knowledge Base, and JESS to realize the rule-based expert system.

4.1 Introduction

Software reuse is not a technology problem nor it is a management problem, although both are important. Reuse is fundamentally a knowledge management problem.

Knowledge is essential in everyday work. Everyone learns by experience and can reuse this knowledge in similar context by adapting to a new situation. The general purpose of Knowledge Management (KM) is to make this kind of knowledge usable for more than one individual, for example an organization. Due to the steadily increasing speed at which new techniques are being developed along with the need of integrating knowledge into processes and products, knowledge is, at present, more and more widely considered as the most important asset of organization [58]. The Knowledge management and the issue about knowledge reuse, in particular, are more important in many organizations, especially for those of Information and Communication Technology (ICT). In fact, ICT enterprises are influenced by the continuing evolution of new technologies, by deal with frequently changing customer requirements and more complex software products, and by cope with the competition in highly demanding markets. These ICT organizations have begun to understand that if they want to be market competitive, they need to manage and use more effective and productive their organizational knowledge [1][7]. In particular, if they want to improve the process of software development, they must increase the reuse of software component, that constitute the greater part of ICT organizational knowledge.

The concept of KM and Reuse are similar to a great extent - while Reuse commonly stands for the reuse of software components, KM denotes the reuse of knowledge. The difference lies in the scope and approach of the concepts. Whereas KM has a wide scope, as almost everything can be regarded as being based on knowledge, reuse is more focused on components appearing in software development. While KM puts an emphasis on tacit knowledge, Reuse focuses on explicit knowledge as embedded in reusable components. In the following sections we show an expert system based approach to applying Knowledge Management to Software Reuse.

4.2 Problem and Tools

This section reviews the concepts of Software Reuse, the definitions of ontology in the era of knowledge management, and it provides a brief description of both JESS and Protégé for the development of this system.

4.2.1 Reuse

Reuse can be defined as further use or repeated use of an artefact (or asset). The idea of reuse in software development was first introduced by M. D. McIlroy in 1969 [59], when he proposed a catalogue of software components from which software could be assembled. It has been widely accepted that Software Reuse offers yet unrealized potential for enhancing software development productivity, and for improving the quality and reliability of the end products [60].

There are commonly two kind of perspective of Reuse: *For Reuse* and *With Reuse*. From perspective *For Reuse*, the reuse is viewed as development of reusable assets, while from perspective *With Reuse*, the reuse is the building of new products by reusing these assets.

The general steps for the procedure of *For Reuse* are (adapted from [61] [62]):

1. Collect requirements from potential reusers.
2. Analyze requirements and domain.
3. Select and prioritize requirements.
4. Develop and document the solution for selected requirements.
5. Evaluate the asset (testing).
6. Store asset with documentation.
7. Support reusers.

In *With Reuse*, instead, a new product is created by taking applicable assets from the asset base, tailoring them as necessary through pre-planned variation mechanisms, such as parameterization, adding any new assets that may be necessary, and assembling the collection. The general steps of the *With Reuse* procedure are (adapted from [61] [62]):

1. Find candidate assets for reuse from asset base.
2. Evaluate the candidate assets with respect to requirements.
3. Select assets to be reused.
4. Adapt/modify the assets, if needed.
5. Integrate assets into product.
6. Test.

In our application we see the problem from perspective of With Reuse; in particular we deal with the first step of the With Reuse procedure. In other words, we try to find, from knowledge Base, a possible candidate software component that satisfy some requirements. To do this we have implemented an expert system integrating tools like Protégé, as Knowledge Base builder, and JESS, as inference engine.

4.2.2 Ontology

The word ontology was taken from philosophy, where it means a systematic explanation of beings. In the last decade, the Knowledge Engineering community has adapted the word ontology to refer to a systematic analysis of knowledge of some domains of interest, so that it can be shared by others. The most often cited ontology definition is from Gruber [17]: *an ontology is a formal, explicit specification of a shared conceptualization.* The term *Conceptualization* refers to an abstract model of phenomena in the world by having identified the relevant concepts of those phenomena.*Explicit* means that the type of concepts used, and the constraints on their use are explicitly defined. *Formal* refers to the fact that the ontology should be machine readable. Finally, *Shared* reflects that ontology should capture consensual knowledge accepted by the communities.

While this is a very general definition, there are several other definitions less abstract. Noy et al. [63] provide a more specific definition: *an ontology is a formal explicit representation of concepts in a domain, properties of each concept describes characteristics and attributes of the concept known as slots and constrains on these slots.* Swartout et al. [64] relates ontology to knowledge base by saying that *an ontology is a hierarchically structured set of terms to describe a domain that can be used as a skeletal foundation for a knowledge base.* Finally for Chandrasekaran et al., an ontology represents a set of vocabulary and related content theory [65]. The vocabulary consists of terms that are used for capturing the conceptualization of the domain, and the content theory refers to the identification of specific classes of objects, their properties, and their relationships that exist in the domain.

In our application, the ontology that describes the domain of interest can be seen from different points of view. In compliance with Noy et al. we have defined our ontologies as set of class, slots and constraints according to the Protégé model. Moreover the use of ontologies allows us to define a set of common terms that improve the communication. Finally, the instances of our ontology constitute the Knowledge Base used by our expert system.

4.2.3 Java Expert System Shell

JESS (Java Expert System Shell) [36][37], as we have seen in the Chapter 2, is a Java based expert system shell, and it was developed by Sandia National Laboratories. Its former form is C Language Integrated Production System (CLIPS) [66] and was also developed by NASA earlier. With the introduction of Rete Algorithm [67], JESS and CLIPS are became very efficient in pattern matching. In addition to the functions that are available in CLIPS, JESS has added other functions that make it a very powerful expert system shell [68][37]. In addition, JESS is capable of conducting inferences with both Forward Chaining and Backward Chaining functions, which makes it being compatible with most expert system shells. In knowledge representation, it allows users to express knowledge content in both rules as well as frames.

4.2.4 Protégé

Protégé [63][54] was developed by Medical Informatics of Stanford University, and it was designed as a platform to reduce the difficulty in knowledge acquisition, which has been recognized as a major bottleneck in developing knowledge system. It is an ontology based development, which allows users to develop knowledge taxonomy and express relationships between categories with ease. One of its important features is the extensible architecture, which enables its integration with other applications; thus one can easily connect external semantic modules to Protégé. It also provides customization features to allow for building knowledge bases with maximum flexibility. The elements of its knowledge model consist of classes, slots, facets, and instances. A class describes a category of objects or concepts that are of the same properties. Instances are the actual entities of a class. Slots represent attributes of classes and instances, and facets express additional information about slots. In addition to these features, this platform provides a set of Open Source API, so that users can design their components in JAVA as plugins to customize the knowledge base. Moreover with the use of JessTab Plugin it is possible integrate JESS into Protégé.

4.3 Expert System Design

As mentioned in the Chapter 3, we have developed an expert system module that have been integrates in Kromos Platform. This prototype of expert system is built to assist project managers during their activities and to find solutions that can be improve the software component reuse. Especially, our ES can offer support for evaluation of previous projects' functionalities reuse. The problem addressed by our expert system is essentially make complex ontology queries to find software requirements that match the user demands.

Generally, an ICT organization produces a great amount of ICT products, which generally serve to automatize business processes. A *process*, in this context, can be seen as a set of interrelated activities, grouped in phases. Moreover, we consider an *ICT product* as a software project that is composed by a set of components, each supporting a singular phase of a given process. Administrative offices, enterprises and any kind of firms, that provide different services or products, generally perform different processes to their business affair. This processes even if belong to different business environment could have certain phases in common: for example in an administrative office, the handling process of incoming mail presents a phase of protocol, as in a supermarket the invoices from suppliers generally are protocoled during their sorting process.

For this reason a software house that must to develop projects for an administrative office and for a supermarket can reuse the same component to automate the protocol phase, reducing developmental costs. But when a software house develops several projects arise two problems: the former is that, for each project, there are several components and a huge amount of documents and the latter is that projects are generally assigned to different project managers which don't know what another project manager has developed.

In this section we propose an expert system which finds reusable components from a given user requirement or else suggests which business process have common phases. To do this, we have created a conceptual model to represent business knowledge and a rule based expert system to support decision making. In particular, our approach are modeled on a real case of study, that is a Sicilian ICT society called Sicilia−eInnovazione. As we have introduced in Chapter3, this society provides a software products to automatize processes of government's offices. In the following sections we illustrate the phases of expert system design.

4.3.1 Ontology Model

Modeling knowledge about the government's offices required some information about its structure, relationship and activities, similarly, for the design of the conceptual schema of the software projects. The initial phase of collection of the needed information, concerned the examined business environment, led us to define our system ontology as a set of two correlated ontologies, a *domain ontology* and a *projects ontology*. In order to keep the ontology easy to understand, only a few concepts from the government offices' domain and from computer engineering projects are collected. This results in a description of Projects with a group of details, attributes and relations, and in a description of Processes and Structure of government's offices.

The **Domain Ontology** is a formal representation of the government's offices structure and activities. It is used to characterize the environment in which the system works, and is organized as a set of concepts and relations;

it reproduces the logical architecture of government offices arranged in levels, each depending on the previous in a hierarchical organization (Fig. 4.1).

The **Project Ontology** is useful to describe ICT company projects. It maps the structure of the project components containing semi-structured explicit knowledge. During the execution of queries, each component of this ontology is used to get all the elements of the domain in agreement with the query (Fig. 4.2).

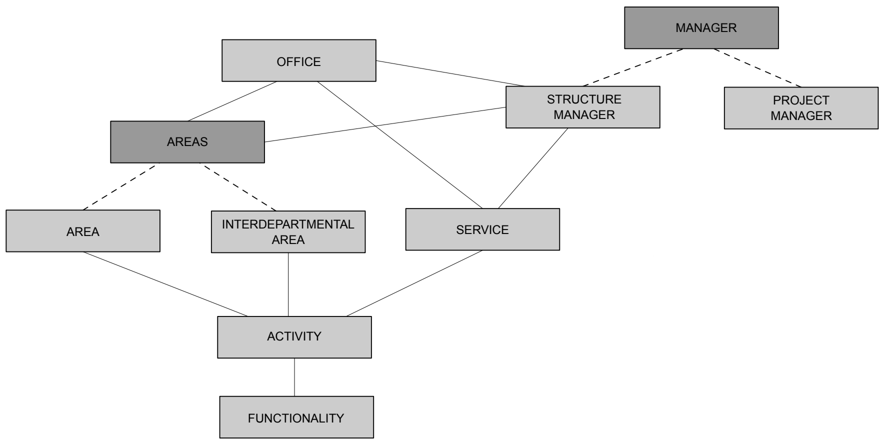


Fig. 4.1: Domain ontology of the Kromos platform.

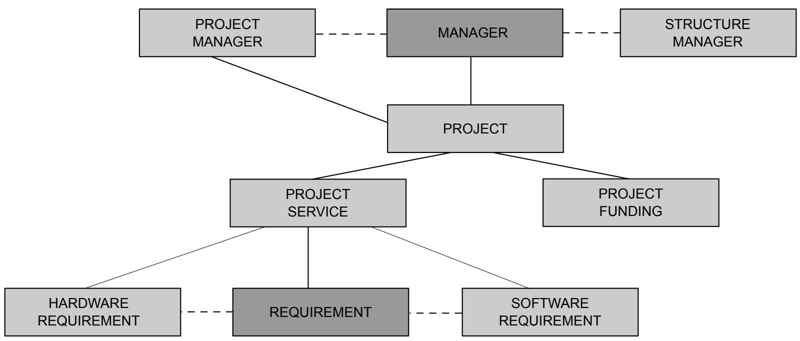


Fig. 4.2: Project ontology of the Kromos platform.

4.3.2 System Model

The ES prototype implemented for Kromos framework uses a rule-based system developed in Jess. The knowledge is coded into a set of rules, each representing a small piece of the expert's skills, incorporating implicit knowledge about the specific field in a computational model. Each rule is an *if-then* statement. Rather than a procedural paradigm, where a single program has a loop that is activated only once, the declarative paradigm used by Jess matches a rule with a single fact specified as its input and processes that fact as its output. When the program is run, the rules engine will activate one rule for each matching fact. The ES exploits two different kinds of knowledge: declarative facts, captured by the ontological model, and procedural facts, expressed using rules defined by an expert.

The code showed below is a portion of rule set used by our ES. In this scenario the expert system uses the declarative part of the knowledge that expresses concepts of OFFICE and PROJECT and the active part of procedural knowledge emulating the behavior of a human expert.

```
 (DEFCLASS office
//represent a general company offices
(multislot linked_projects
(comment "projects of the department"
(type INSTANCE))
(single-slot name
(type STRING)
(cardinality 0 1))
(single-slot council_of_competence
(type INSTANCE)
(allowed-classes COUNCIL)
(cardinality 0 1))
(multi-slot comprehend_services
(type INSTANCE)
(allowed-classes SERVICE)
...

(defrule office_processes ?instance
<-(object (is-a OFFICE)
(name ?n&:(call ?*ric* equals
(slot-get"+ off+"name)
(lowcase ?n))))=>(bind $?area
(slot-get (instance-name ?instance)
comprehend_areas
(foreach ?j $?area
(if  (call ?*ric* different
(slot-get (instance-name ?j) name) empty)
then...
```

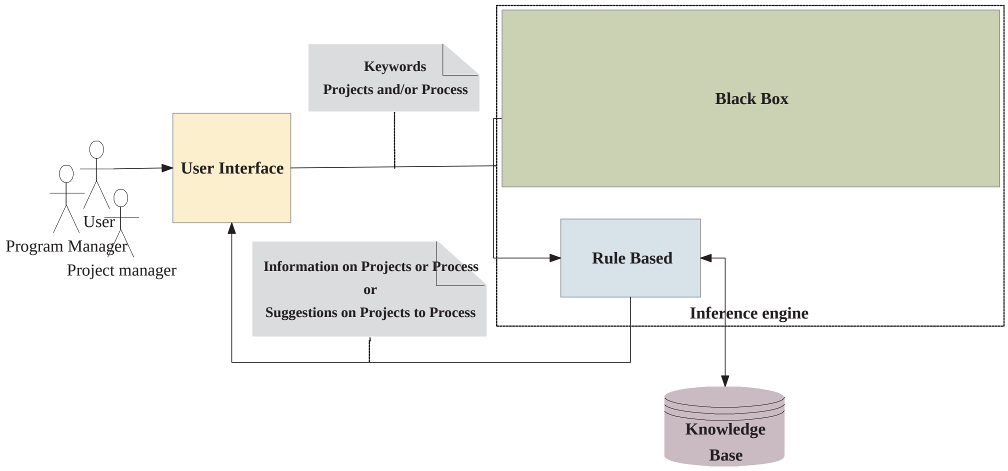


Fig. 4.3: Architecture of Rule Based Expert System of the Kromos Platform.

4.4 Expert System Architecture

The architecture of proposed expert system is shown in Fig. 4.3[1]. This architecture is made up of modules of knowledge base, inference engine and a user interface. The knowledge base module is the core of the system; it consists of the instances of the previous defined domain and project ontology. The user interface, instead, accepts a user query with some keywords and invokes the inference engine to activate the decision rules. Then, the decision rules will access to the relevant data and information residing in the Knowledge Base. The conclusion of the inference process will be presented through the user interface. Additionally, from user interface it is also possible modify, update and insert contents into knowledge base.

The entire ES module is built in JAVA environment, which is essential in achieving integration between several tools used during development. In fact, to develop the knowledge base, we made use of the well known ontology-based knowledge platform Protégé to build the ontology that formalize the domain and project knowledge, and we utilize the JESS rule to build the operational knowledge that is the decision rules. Protégé is JAVA based and is designed with an open architecture, which makes it possible to communicate with other JAVA based systems through Plugins. For the case with JESS, the Plug-in JessTab [68] was specifically designed to serve as a

[1]In the Fig. 4.3 is showed a black box because this rule-based expert system is a part of a more complex expert system (called Kromos Expert System in Chapter 3). For this reason, we have preferred to hide some part of the architecture that will explain in the Chapter 5

bridge to allow Protégé to communicate with JESS. API of Protégé is able to work with functions of JessTab, which enables JESS inference engine to function as the inference engine of the combined expert system. During execution, functions of JessTab can manage Protégé knowledge base, and allows rules in JESS to access contents of the ontology in Protégé. This ability enhances the level of reasoning complexity of Protégé beyond its original capability. The following code portion presents commands for actual implementation, which brings both Protégé and JESS into the JAVA environment. The load-project command allows to load the current Protégé project while the getProtegeKB() command makes accessible the knowledge Base. Finally, the "mapclass"command maps the contents, including instances, attributes, and corresponding relational structure, of Protégé into a template that is accessible by the JESS engine.

```
import edu.stanford.smi.protege.model.*;
import se.liu.ida.JessTab.*;
import jess.*;
...

// The knowledge base
KnowledgeBase kb;

// The engine
 Rete engine=new Rete();

// protégé project path
String PROJECT_FILE_PATH = " /Progetto_protege/SeI.pprj";

// JessTab Invocation

engine.addUserpackage(new JessTabFunctions());

// loading protégé project into rete engine
engine.eval("(load-project " + PROJECT_FILE_PATH + ")");

// loadin knowledge base
kb = JessTab.getProtegeKB();

// executing mapclass command for each protégé classes
engine.eval("(mapclass PROGETTO)");

...
```

Chapter 5

Optimizing ICT Project Planning Process

Enterprise planning process is an important issue in business environments because it allows being market competitive. When a new project must be developed, the project managers have planning the activities required to reach own goals. In order to achieve these objectives, the managers must be taking into account some issues. In this chapter we give a particular attention to optimize Project Planning Process in Information and Communication Technology (ICT) organization. Especially we take into account some factors that can influence this process, defining a decision support system that performs two kind of reasoning: fuzzy reasoning and a probabilistic reasoning.

5.1 Introduction

In today's dynamic environment, enterprises need to incorporate strategies in order to planning their future work, in particular those belong to Information and Communication Technology area. Generally, an ICT organization has more projects committed and therefore available for future planning. In order to decide which of the proposed projects should be selected, a number of factors must be considered. In fact, each project includes environmental advantages and disadvantages, tangible and intangible benefits, costs, availability of human and hardware resource and many others. Several studies have been proposed to help organizations in selection of good projects. Most of these studies have focused on the fields information technology [69][70][71][72] and marketing [73][74] using several approaches which take into account various aspects of project selection. In this thesis, to address this problem, we used a decision support system founded on a mixed approach based on fuzzy logic and Bayesian network.

A project planning process is a typical decision making problem. In fact, in complex and dynamical environment, such as marketing context, it is difficult to take decisions because there are several factors to take into account. Because, as well as we have said before, this process is a delicate enterprise task since it will have effects on its future incomes. Making right decisions is fundamental to enterprise survival. Moreover, the greatest part of the factors that influence the project planning process are sources of uncertainty that must be opportunely valued in order to optimize the decisional task.

In particular in this work we considerer two types of uncertainty factors: the former due to non-deterministic events and the latter due to the subjectivities of the decision maker. In order to cope with these two issues we have developed an expert system for decision support by means of the fusion of two kind of reasoning: a fuzzy reasoning and a probabilistic reasoning. In the following section we explain in detail its features.

5.2 A Decision Support System for Optimization of ICT Project Planning Process

As we previously said, the project planning (also called selection) process can be influenced by two kinds of uncertainty factors: the former due to non-deterministic events and the latter due to the subjectivities of the decision maker. In order to address this problem we have developed a decision support system which integrates two kinds of expert systems: a fuzzy expert system and a probabilistic expert system based on Bayesian network. The Fig.5.1 shows its architecture. The main components of the system are:

- a user interface that has an obvious function;
- a rule based expert system (described in the Chapter 4) which in this context is used to obtain some features about a project to be examined;
- a fuzzy expert system that analyzes some of the previous features in order to obtain a project complexity value (it will be fully explained in the section 5.3);
- a probabilistic expert system that evaluates whether or not select the proposed project on the basis of forecasts and propagate evidence (it

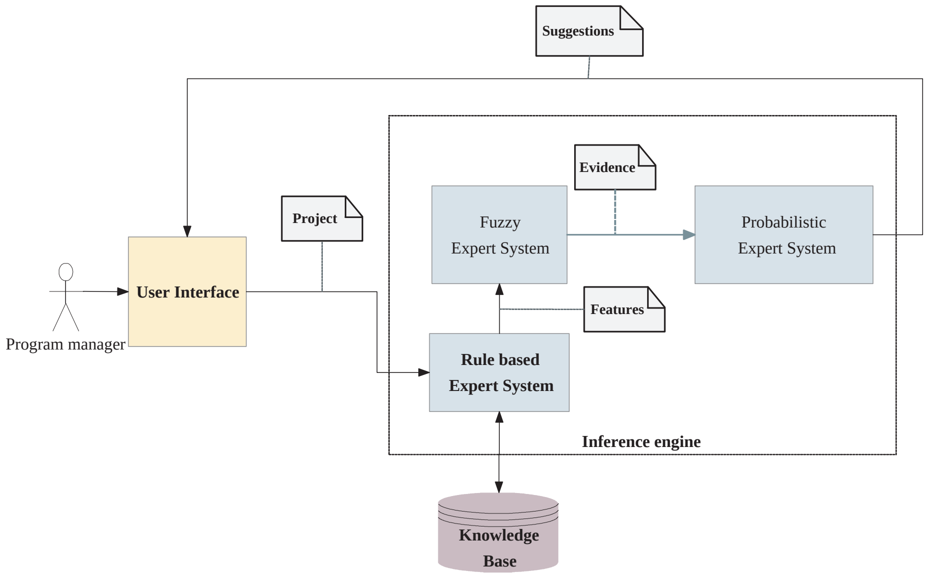


Fig. 5.1: Architecture of Decision Support system.

will be fully explained in the section 5.4).

5.3 Fuzzy Reasoning to Evaluate Project Complexity

In a project planning process must be taken into account some characteristics which provide a basis for determining the appropriate managerial actions. Complexity is one such critical project factors.

Bennett in [75] observes that frequently individuals describe their projects as simple or complex when they are discussing management issues. This indicates a practical acceptance that complexity makes a difference to the management of projects. Consequently, it is obvious that complex projects demand an exceptional level of management. Furthermore, the importance of complexity to the project management process is widely acknowledged, for example:

- It allows to determine planning, coordination and control requirements[76].
- It is an important criteria in the selection of an appropriate project organizational form [75].
- It influences the selection of project.

- It affects the project objectives of time, cost and quality. Broadly, a high value of project complexity increases times and costs [77].

For these reasons, making a correct valuation about *how complex is a given project* is fundamentally task. There is, however, a question to address: this task is "human sensitive". In fact, a decision maker generally has a personal opinion about a problem to solve rather than an objective judgment about the evaluation of the criteria that influence the given problem. So he is unable to provide exact numbers. Moreover, because a human opinion is based on individual capabilities and experiences, several opinions are possible about the complexity of a given project. To overcome these issues, fuzzy set theory has been considered to do the project complexity evaluation.

5.3.1 Fuzzy Expert System Design

The fuzzy expert system proposed in this work was built according to the following methodology. First of all, we have analyzed the factors that influence an ICT project complexity associating them some linguistic variables, and defined the appropriate membership functions. Then we have established the rules of the inference engine, and at last, we have integrated all these elements to develop the fuzzy expert system.

5.3.1.1 Linguistic Variables

The first step for developing a model for systematically measuring complexity is defining a set of software project characteristics that are the sources of complexity. An exhaustive list with sources of complexity would be enormously large, and at the end of no practical use since it would be proven difficult to measure everything. In [78] is presented a summary of the identified source of complexity along with indicative metrics that could be used. For our goal, we have considered some source of complexity described in the table 5.1. For each of these source of complexity defined in this table, we have associated them a corresponding linguistic variable.

Table 5.1: Definition of the sources of project complexity.

Complexity Source	Definition
Problem	The problem is well defined? It is easy to understand?
Scheduling	The customer has established an imperative deadline?
Topic	How much the project's topic belongs to the enterprise know-how?
Level of change required	Has the project a huge impact on existing client's framework?
Functionalities Reuse	It is possible reuse functionalities previously developed?

In particular, our fuzzy system works with five inputs(*Problem, Scheduling, Topic, Level of Change, Functionalities Reuse*)and one output (*Complexity*) linguistic variables. In the Fig.5.2 we show the linguistic variables given in input to our expert system and their fuzzy values. For each of them we have assigned a weight specifying their contribution to whole final complexity value.

Weight	Parameter	High	Moderate	Low
wp	Problem	*Easily understood*	*Moderately difficult to understood*	*Very difficult to define*
ws	Scheduling	*Flexible*	*Deadlines may be moderately extended*	*Deadline is fixed and cannot be changed*
wt	Topic	*Solution early achievable using existing technologies*	*Solution is unclear because technologies is new in organization*	*Solution is difficult to achieve. It require complex technologies*
wl	Level of Change	*Impact a single unit*	*Impact a moderately number of units*	*It have a huge impact of existing units*
wf	Functionalities Reuse	*Many*	*Medium*	*Low*

Fig. 5.2: Linguistic variables of Fuzzy Decision Making system.

5.3.1.2 Fuzzy Rules

Our expert system takes a decision, about complexity of a given project, through a schema based on five variables above mentioned. The rules are determined considering the impact that each parameter has on final decision by means of the weight associated. Following it is showed a portion of code of the expert system.

```
// Block definition

FUNCTION_BLOCK fuzzy_expert

// Define input variables
VAR_INPUT
   Scheduling : REAL;
   Level_of_Change : REAL;
   Functionalities_Reuse : REAL;
   Problem : REAL;
   Technologies : REAL;
END_VAR

// Define output variable
VAR_OUTPUT
    Complexity : REAL;
END_VAR

// Fuzzify input variable 'Scheduling'
FUZZIFY Scheduling
    TERM low := (0, 1) (4, 0) ;
    TERM moderate := (1, 0) (4,1) (6,1) (8,0);
    TERM high := (6, 0) (10, 1);
END_FUZZIFY

// Fuzzify input variable 'Level_of_Change'
FUZZIFY Level_of_Change
    TERM low := (0, 1) (2, 1) (4,0) ;
    TERM moderate := (2, 0) (4,1) (6,1) (8,0);
    TERM high := (6, 0) (10, 1);
END_FUZZIFY

// Fuzzify input variable 'Functionalities_Reuse'
FUZZIFY Functionalities_Reuse
    TERM low := (0, 1) (2, 1) (4,0) ;
    TERM moderate := (2, 0) (4,1) (6,1) (8,0);
```

```
    TERM high := (6, 0) (10, 1);
END_FUZZIFY

// Fuzzify input variable 'Problem'
FUZZIFY Problem
    TERM low := (0, 1) (20, 1) (40,0) ;
    TERM moderate := (20, 0) (40,1) (60,1) (80,0);
    TERM high := (60, 0) (100, 1);
END_FUZZIFY

// Fuzzify input variable 'Technologies'
FUZZIFY Technologies
    TERM low := (0, 1) (20, 1) (40,0) ;
    TERM moderate := (20, 0) (40,1) (60,1) (80,0);
    TERM high := (60, 0) (100, 1);
END_FUZZIFY

// Defuzzify output variable 'tip'
DEFUZZIFY Complexity
    TERM low := (0, 1) (20, 1) (40,0) ;
    TERM moderate := (20, 0) (40,1) (60,1) (80,0);
    TERM high := (60, 0) (100, 1);
    // Use 'Center Of Gravity' defuzzification method
    METHOD : COG;
    // Default value is 0 (if no rule activates defuzzifier)
    DEFAULT := 0;
END_DEFUZZIFY

RULEBLOCK No1
  // Use 'min' for 'and'
    AND : MIN;
    // Use 'min' activation method
    ACT : MIN;
    // Use 'max' accumulation method
    ACCU : MAX;
 RULE 1 : IF Scheduling IS high AND Level_of_Change IS high AND
          Functionalities_Reuse IS high AND Problem IS high AND
          Technologies IS high THEN Complexity IS high;
 RULE 2 : IF Scheduling IS high AND Level_of_Change IS high AND
          Functionalities_Reuse IS high AND Problem IS high AND
          Technologies IS moderate
          THEN Complexity IS high;
 RULE 3 : IF Scheduling IS high AND Level_of_Change IS high AND  ....
```

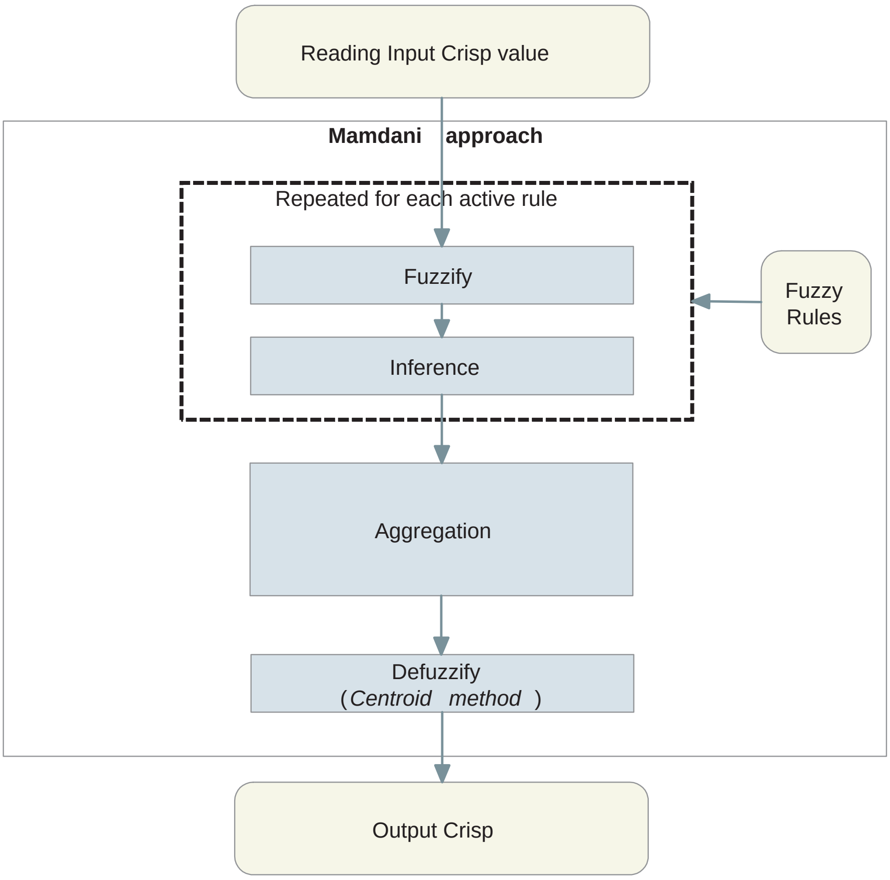


Fig. 5.3: Functional architecture of fuzzy expert system for project complexity evaluation.

5.3.1.3 Functional Architecture

The functional architecture of the expert system is illustrated in Fig. 5.3. Using the Mamdani approach, the system troughs four functional phases:

- *Fuzzification*: It measures the values of the input variables on their membership functions to determine the degree of truth for each rule premise.

- *Inference*: The evaluation of a rule is based on computing the truth value of its premise part and applying it to its conclusion part. This results in assigning one fuzzy subset to each output variable of the rule. In Min Inference the entire strength of the rule is considered as the minimum membership value of the input variables' membership

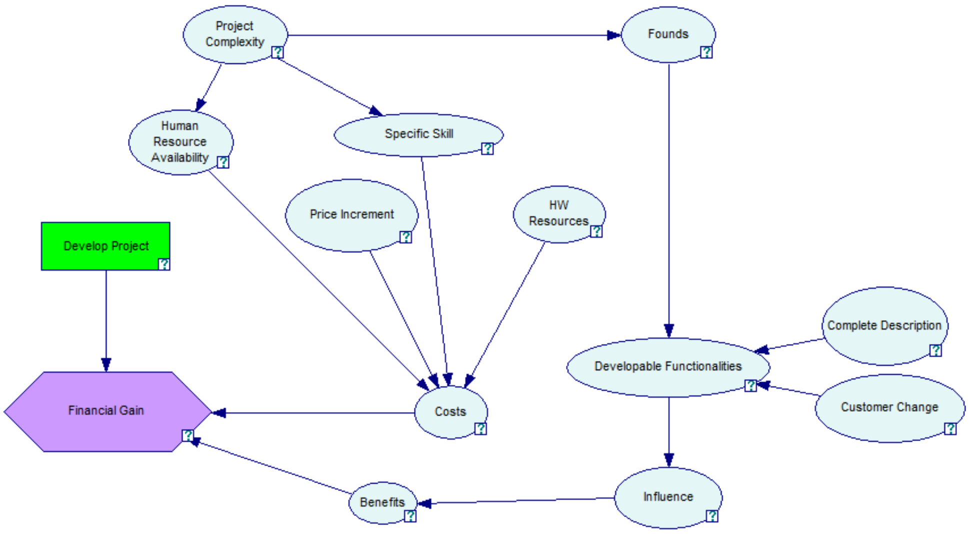


Fig. 5.4: Projects decision network.

values. A rule is said to fire, if the degree of truth of the premise part of the rule is not zero.

- *Aggregation*: It combines the conclusions inferred from the rules through a MAX aggregation operator.
- *Defuzzification*: It converts a fuzzy output into a crisp output. The method used in this work is the Centroid method.

5.4 Probabilistic Reasoning for Project Selection

In some complex and dynamical environment it is difficult to take decisions when it is dependent by uncertainty factors. The projects selection process is a delicate enterprise task because it will have effects on future incomes and because it is based on many different factors. The greatest part of such factors (like as time, costs, resources), being not deterministic, represent sources of uncertainty that must be opportunely esteemed in order to optimize the decisional trials of business planning. In the last years, the BN has become a popular representation for encoding uncertain expert knowledge in expert system [79]. In this work, we have proposed a Bayesian network to support the project selection process.

5.4.1 Bayesian Network Design

In this section, we present a model of Bayesian network in order to establish which project is more convenient in terms of costs/benefits. The development of the net has been divided in three phases: domain analysis, relationships discovery among the variables of interest and estimate of the probability table. The obtained decision network is shown in figure 5.4.

5.4.1.1 Domain Analysis

A set of stochastic variables that characterizes the domain of interest and their possible states are defined:

- *Project Complexity(PC)*= How the project is intricate. States Values: High, Medium and Low.
- *Founds (F)*= Available financing in Million of Euro (MEUR): States Values: Shorter than 1 MEUR, Between 1 and 3 MEUR and More than 3 MEUR.
- *Human Resources (HR)* = Availability of employees. States Values: Full Time and Part Time.
- *Specific Skills (SS)* = Availability of human resources with specific technical Know How. States Values: High, Medium and Low.
- *Develop Functionalities (DF)* = Project functionality that can be implemented. States Values: Many, Mean and Few.
- *Complete Description (CD)* = If the customer has provided an exhaustive description of the product. States Values: True and False.
- *Customer Changes (CC)* = If the customer can bring changes to the requisite of the product in work progress. States Values: True and False.
- *Influence (I)* = How much the project is important?. States Values: Many, Mean and Few.
- *Costs (C)* = Total costs for the development of the project. States Values: High, Medium and Low.
- *Price Increment (PI)* = Prices growing (raw material, renewal employment contract etc.). States Values: True and False.

- *HW Resources (HW)*= How much hardware is necessary to project development?. States Values: High, Medium and Low.
- *Benefits (B)* = Improvement of human process, of time and costs. States Values: High and Low.
- *Financial Gain (FG)*= It contains information about the decision maker's goals.
- *Develop Project(DP)*= It models decision maker's options.

5.4.1.2 Relationships Discovery

The Relationships discovery phase allows for the discovering of the causal relationships between the variables objects of our observation. In our modeled domain this set of dependence and independence conditions were discovered:

$$\begin{gathered} PC, PI, CD, CC, HW|\varnothing; \quad F, HR, SS|PC; \\ C|HW, PI, HR, SS; \quad DF|SS, CD, CC; \quad B|I; \quad I|DF; \end{gathered} \tag{5.1}$$

The symbol | represents the dependence of the right-side set of variables by the left-side set of variables.

5.4.1.3 Probability Tables Definition

The tables of conditional probability distributions for each node of the BN are generally the hardest task to execute. The main difficult in enterprise context, concerns the experience of the domain expert. Because every specialist could have been managed only some projects cases that could be a non-realistic sample set to estimate the probability tables. To avoid that, we consider different opinion about the same node coming from different domain experts, to reach a unique final opinion expressed as probability. To merge these experience data in a unique value of probability, the schema shown in the following figure was adopted:

The opinions about node j, produced by N experts can be combined using a weighted average:

$$f_j(o) = (\sum_{i=1}^{N} w_i o_i)/(\sum_{i=1}^{N} w_i) \tag{5.2}$$

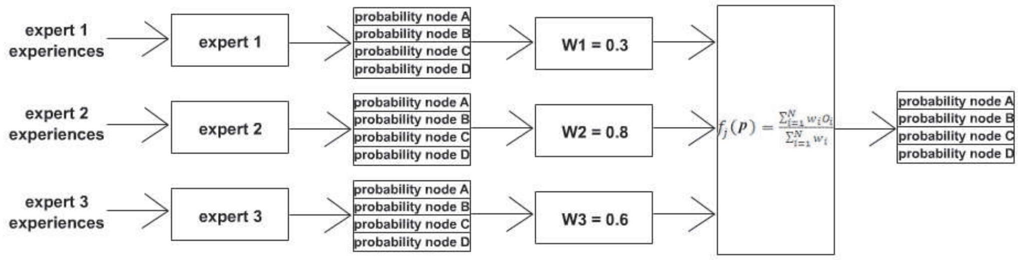


Fig. 5.5: Fusion schema of experts' opinions.

Where O_i indicates the percentage value of the opinion of the expert i, while w_i means the related weight. w_i is calculated as it follows:

$$\begin{aligned} w_i = G_i * A_i/(G_{max} * A_{max}) \\ with \quad G \in [1, G_{max}] \quad and \quad A \in [1, A_{max}] \end{aligned} \tag{5.3}$$

where G indicates the degree of experience of the expert in the company, while A indicates the number of years he worked for that company.

Chapter 6

Conclusion and Future Works

6.1 Summary and Conclusions

In this thesis I present *Kromos*, a work developed in collaboration with the Computer Engineering Department of Palermo University and the Sicilian local Government ICT society, Sicilia e-Innovazione.

Kromos grow out of the initial idea to create a useful tool for document management. After it evolved in a more complex framework able, not only to manage efficiently and intelligently documents, but also to become a helpful decision making tool. The main idea to reorganize the huge amount of project documents and to enhance the corporate know-how led me to investigate about some issues, such as:

- Which knowledge domain is useful to project managers in their activities?
- What are the sources of information in an ICT enterprise?
- How to represent the knowledge in a globally integrated form?
- How to use the knowledge efficiently?
- Which project is more convenient to take?
- It is possible to give an assessment to a project valuating some of its features?
- How to find software requirements for reusable components in oceans of projects?

In order to answer these questions and many others not mentioned here, I analyzed deeply the domain of interest and explored possible solutions. It was thought then to combine artificial intelligence technologies in an innovative way to make enterprise processes smarter. This was the genesis of Kromos.

Kromos is, in a nutshell, a Knowledge Management System for ICT business environment. It is mainly a tool for supporting several enterprise activities, offering specific technologies to encompass all needs and necessities analyzed.

In this thesis, at first I showed the whole framework with its main components. Then, I focused on Decision Support System component, addressing problems such as software reuse and projects selection. It was shown, therefore, how this KM model can be applied to reuse processes. In particular, it was illustrated how to find reusable software components by means of a rule based expert system, built on ontological description of business domain. Successively, I proposed a model for uncertainty reasoning, in order to cope not only to the unpredictability of some events that are intrinsically linked to the market environment, but also to overcome the vagueness and subjectivities of human judgments. This model is based on a fuzzy reasoning, which allows evaluating the complexity of an ICT projects unifying the contribution of several factors that complicate a project, and on a Bayesian network able to estimate the feasibility of a project on the basis of the evidence derived from fuzzy reasoning.

In conclusion, this dissertation wants to underline the following issues:

- A novel fusion of Artificial Intelligence techniques in order to the management of tacit and explicit knowledge in business environments has been proposed.

- A Knowledge Management framework was developed as a useful tool for different skilled workers. The system, composed by different subsystems and modules (such as the Document Management Subsystem for extraction of information and the Decision Support Expert System for knowledge reuse and project selection) can be used for a wide range of scenarios.

- The proposed KMS is modular, so it is adaptable to different domains by substitution of a limited set of rules.

- The prototype of Decision Support System presented is able to address issues about uncertainty reasoning.

- Some component of the framework was evaluated on a real ICT business environment, observing some factors like reduction of information retrieval time, diffusion of expertises and experiences.

6.2 Future Directions

Knowledge Management, as seen in the previous chapters, can be defined as a systematic approach for improving the handling of knowledge on all levels of an organization (individual, group and organizational level) in order to support the organization's business goals, such as innovation, quality, cost reduction and so on. KM is also a management discipline that combines methods for human resource management, strategic planning and organizational behavior.

Researches on KM are focused on the use of knowledge within organizations, thus aiming at the distribution of relevant information. This process is typically supported by centralized approaches. Knowledge about people, knowledge about processes, and domain knowledge are represented and maintained as information in global repositories, which are sources to meet knowledge worker's information needs. Such repositories may be structured by means of global ontologies and made accessible through knowledge portals, as Kromos.

A very interesting approach to distribute knowledge is represented by software agents. Agent based systems are able to support various aspects of Knowledge Management. They can be used both as personal information agents for knowledge retrieval and as agent based workflows for business process. The main features of autonomy and proactivity make an agent very useful to solve complex problems. These agent's features along with the progress of WWW, which has created a parallel world in which a huge amount of entities exist virtually in a universe of elaborate information [80], provide new research perspectives.

For these reasons, as well as many others [81][82][83][84][85], I consider very interesting the agent paradigm for the design of a self-organized virtual enterprise. In my opinion, this approach may have many advantages in a model where agents perform uncertainty reasoning, thus I'm directing my future works toward the formulation of such model.

Bibliography

[1] T. Davenport and L. Prusak, *Working knowledge: How organizations manage what they know.* Harvard Business Press, 2000.

[2] H. Tsoukas, "What is Organizational Knowledge?" *Journal of Management Studies*, vol. 38, no. 7, pp. 973–993, 2001.

[3] D. O'Leary, "Enterprise knowledge management," *Computer*, vol. 31, no. 3, pp. 54–61, 2002.

[4] I. Nonaka, "A dynamic theory of organizational knowledge creation," *Organization science*, vol. 5, no. 1, pp. 14–37, 1994.

[5] I. Nonaka and N. Konno, "The concept of ŚBaŠ: building a foundation for knowledge creation," *The knowledge management yearbook 1999-2000*, p. 37, 1999.

[6] H. Takeuchi and I. Nonaka, "Knowledge creation and dialectics," *Hitosubashi on Knowledge Management*, pp. 1–29, 2004.

[7] I. Nonaka and H. Takeuchi, "The knowledge-creating company," *New York*, vol. 1, p. 995, 1995.

[8] S. Staab, R. Studer, H. Schnurr, and Y. Sure, "Knowledge processes and ontologies," *Intelligent Systems, IEEE*, vol. 16, no. 1, pp. 26–34, 2005.

[9] M. Alavi and D. Leidner, "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues," *MIS quarterly*, vol. 25, no. 1, pp. 107–136, 2001.

[10] M. Alvarez, J. Arana, I. Pradales, and J. Santos, "Knowledge Management in the Extended Enterprise."

[11] C. Verwijs, J. Dijkstra, E. Heeren, and D. De Wit, *Kennismaken met kennismanagement.* Telematica Instituut, 1999.

[12] T. Davenport and J. Glaser, "Just-in-time delivery comes to knowledge management." *Harvard Business Review*, vol. 80, no. 7, p. 107, 2002.

[13] D. Holtshouse, "Knowledge research issues," *California Management Review*, vol. 40, pp. 277–280, 1998.

[14] T. Stewart, *The wealth of knowledge: Intellectual capital and the twenty-first century organization.* Random House, Inc., 2003.

[15] K. Hangos, R. Lakner, and M. Gerzson, *Intelligent control systems: an introduction with examples.* Kluwer Academic Pub, 2001.

[16] N. Guarino and P. Giaretta, "Ontologies and knowledge bases: Towards a terminological clarification," *Towards very large knowledge bases: knowledge building and knowledge sharing*, vol. 1, no. 9, p. 9, 1995.

[17] T. Gruber *et al.*, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, pp. 199–199, 1993.

[18] ——, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human Computer Studies*, vol. 43, no. 5, pp. 907–928, 1995.

[19] N. Guarino, "Understanding, building and using ontologies," *International Journal of Human Computer Studies*, vol. 46, pp. 293–310, 1997.

[20] N. Guarino and I. (Roma) Consiglio nazionale delle ricerc, "Formal ontology in information systems," 1998.

[21] D. O'Leary, "A multilingual knowledge management system: A case study of FAO and WAICENT," *Decision Support Systems*, vol. 45, no. 3, pp. 641–661, 2008.

[22] L. Razmerita, "Ontology-Based User Modeling," *Ontologies*, pp. 635–664, 2007.

[23] L. Sui, "Decision support systems based on knowledge management," in *Services Systems and Services Management, 2005. Proceedings of ICSSSM'05. 2005 International Conference on*, vol. 2. IEEE, 2005, pp. 1153–1156.

[24] D. Harvey and R. Holdsworth, “Knowledge management in the aerospace industry,” in *Professional Communication Conference, 2005. IPCC 2005. Proceedings. International*. IEEE, 2005, pp. 237–243.

[25] M. Chun, K. Sohn, and P. Granados, “Systems Theory and Knowledge Management Systems: The Case of Pratt-Whitney Rocketdyne,” in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*. IEEE, 2008, p. 336.

[26] L. Ceccaroni, “Ontowedss-an Ontology-based Environmental decision-support system for the Management of Wastewater Treatment Plants,” 2003.

[27] L. Ceccaroni, U. Cortés, and M. Sānchez-Marrč, “OntoWEDSS: augmenting environmental decision-support systems with ontologies,” *Environmental Modelling & Software*, vol. 19, no. 9, pp. 785–797, 2004.

[28] Z. Turk, B. Bjork, C. Johansson, and K. Svensson, “Document management systems as an essential step towards CIC,” 1994.

[29] D. Niyogi and S. Srihari, “An integrated approach to document decomposition and structural analysis,” *International Journal of Imaging Systems and Technology*, vol. 7, no. 4, pp. 330–342, 1996.

[30] G. Nagy, J. Kanai, M. Krishnamoorthy, M. Thomas, and M. Viswanathan, “Two complementary techniques for digitized document analysis,” in *Proceedings of the ACM conference on Document processing systems*. ACM, 2000, pp. 169–176.

[31] M. Ginsburg and A. Kambil, “Annotate: A web-based knowledge management support system for document collections,” in *System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*. IEEE, 2002, p. 10.

[32] T. Kucza, M. N
"attinen, and P. Parviainen, “Improving knowledge management in software reuse process,” *Product Focused Software Process Improvement*, pp. 141–152, 2001.

[33] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Prentice hall, 2009.

[34] D. Michie, S. for Artificial Intelligence, and S. of Behaviour, *Expert systems in the micro-electronic age*. Edinburgh University Press, 1979.

[35] M. Druzdzel and R. Flynn, “Decision Support Systems. Encyclopedia of Library and Information Science. A. Kent,” 1999.

[36] E. Friedman-Hill *et al.*, “Jess, the rule engine for the java platform,” *Sandia National Laboratories*, 2005.

[37] E. Friedman-Hill, *JESS in Action*. Manning, 2003.

[38] “Resource available at http://herzberg.ca.sandia.gov/jess/.”

[39] K. Ng and B. Abramson, “Uncertainty Management in Expert Systems,” *IEEE Expert*, vol. 5, no. 2, pp. 29–48, 1990.

[40] I. Good, “Kinds of Probability: Although there are at least five kinds of probability, we can get along with just one kind,” *Science*, vol. 129, no. 3347, p. 443, 1959.

[41] L. Savage, *The foundations of statistics*. Dover Pubns, 1972.

[42] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[43] L. Zadeh, “Fuzzy sets*,” *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[44] ——, “Fuzzy logic,” *Computer*, vol. 21, no. 4, pp. 83–93, 2002.

[45] H. Zimmermann, *Fuzzy sets, decision making, and expert systems*. Springer, 1987.

[46] D. Dubois and H. Prade, “Fuzzy sets and probability: misunderstandings, bridges and gaps,” in *Fuzzy Systems, 1993., Second IEEE International Conference on*. IEEE, 2002, pp. 1059–1068.

[47] P. Ribino, A. Oliveri, G. Re, and S. Gaglio, “A Knowledge Management System Using Bayesian Networks,” *AI* IA 2009: Emergent Perspectives in Artificial Intelligence*, pp. 446–455, 2009.

[48] A. Oliveri, P. Ribino, S. Gaglio, G. Lo Re, T. Portuesi, A. La Corte, and F. Trapani, “KROMOS: ontology based information management for ICT societies,” 2009.

[49] P. Ribino, A. Oliveri, G. Re, and S. Gaglio, “A Knowledge Management System Based on Ontologies,” in *New Trends in Information and Service Science, 2009. NISS’09. International Conference on*. IEEE, 2009, pp. 1025–1033.

[50] SiciliaeInnovazione, "Available at http://www.siciliaeinnovazione.it."

[51] R. Michel and L. Methlie, "Knowledge-based Decision Support Systems with Applications in Business," 1995.

[52] K. Wiig, *People-focused knowledge management: how effective decision making leads to corporate success.* Butterworth-heinemann, 2004.

[53] M. Minsky, "A framework for representing knowledge," 1974.

[54] W. Grosso, H. Eriksson, R. Fergerson, J. Gennari, S. Tu, and M. Musen, "Knowledge modeling at the millennium (the design and evolution of Protégé-2000)," 1999.

[55] V. Chaudhri, A. Farquhar, R. Fikes, P. Karp, and J. Rice, "OKBC: A programmatic foundation for knowledge base interoperability," in *Proceedings of the National Conference on Artificial Intelligence.* JOHN WILEY & SONS LTD, 1998, pp. 600–607.

[56] E. Hatcher and O. Gospodnetic, "Lucene in action," 2004.

[57] S. Brown, S. Li, R. Burdick, B. Sullivan, M. Nelson, S. Wilkinson, R. Johnson, G. Taylor, J. Falkner, B. Galbraith *et al.*, *Professional jsp.* Wrox Press Ltd. Birmingham, UK, UK, 2001.

[58] A. Carneiro, "How does knowledge management influence innovation and competitiveness?" *Journal of Knowledge Management*, vol. 4, no. 2, pp. 87–98, 2000.

[59] M. McIlroy, J. Buxton, P. Naur, and B. Randell, "Mass produced software components," *Software Engineering Concepts and Techniques*, pp. 88–98, 1969.

[60] I. Jacobson, M. Griss, and P. Jonsson, *Software reuse: architecture, process and organization for business success.* ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1997.

[61] E. Karlsson, *Software reuse: a holistic approach.* John Wiley & Sons, Inc. New York, NY, USA, 1995.

[62] W. Lim, *Managing software reuse: a comprehensive guide to strategically reengineering the organization for reusable components.* Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1998.

[63] N. Noy, R. Fergerson, and M. Musen, "The knowledge model of Protege-2000: Combining interoperability and flexibility," *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pp. 69–82, 2000.

[64] B. Swartout, R. Patil, K. Knight, and T. Russ, "Toward distributed use of large-scale ontologies," in *Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems*, 1996.

[65] B. Chandrasekaran, J. Josephson, and V. Benjamins, "What are ontologies, and why do we need them?" *Intelligent Systems and Their Applications, IEEE*, vol. 14, no. 1, pp. 20–26, 2002.

[66] J. Giarratano and G. Riley, *Expert systems: principles and programming.* Brooks/Cole Publishing Co. Pacific Grove, CA, USA, 1989.

[67] C. Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem* 1," *Artificial intelligence*, vol. 19, no. 1, pp. 17–37, 1982.

[68] H. Eriksson, "Using JessTab to integrate protégé and jess," *Intelligent Systems, IEEE*, vol. 18, no. 2, pp. 43–50, 2005.

[69] L. Machacha and P. Bhattacharya, "A fuzzy-logic-based approach to project selection," *Engineering Management, IEEE Transactions on*, vol. 47, no. 1, pp. 65–73, 2002.

[70] J. Klapka and P. Pios, "Decision support system for multicriterial R&D and information systems projects selection," *European Journal of Operational Research*, vol. 140, no. 2, pp. 434–446, 2002.

[71] J. Lee and S. Kim, "Using analytic network process and goal programming for interdependent information system project selection," *Computers & Operations Research*, vol. 27, no. 4, pp. 367–382, 2000.

[72] M. Fasanghari and F. Roudsari, "Optimized ICT project selection utilizing fuzzy system," *World Applied Sciences Journal*, vol. 4, no. 1, pp. 44–49, 2008.

[73] E. Danneels and E. Kleinschmidt, "Product innovativeness from the firm's perspective: its dimensions and their relation with project selection and performance," *Journal of Product Innovation Management*, vol. 18, no. 6, pp. 357–373, 2001.

[74] R. Thieme, M. Song, and R. Calantone, "Artificial neural network decision support systems for new product development project selection," *Journal of Marketing Research*, vol. 37, no. 4, pp. 499–507, 2000.

[75] J. Bennett, *International construction project management: general theory and practice*. Butterworth-Heinemann, 1991.

[76] K. Bubshait and W. Selen, "Project characteristics that influence the implementation of project management techniques: a survey," *Project Management Journal*, vol. 23, no. 2, pp. 43–47, 1992.

[77] S. Rowlinson, "An Analysis of Factors Affecting Project Performance in Industrial Building. Brunel University," Ph.D. dissertation, Ph. D. Thesis, 1998.

[78] P. Fitsilis, "Measuring the complexity of software projects," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 7. IEEE, 2009, pp. 644–648.

[79] D. Heckerman, A. Mamdani, and M. Wellman, "Real-world applications of Bayesian networks," *Communications of the ACM*, vol. 38, no. 3, pp. 24–26, 1995.

[80] J. Burke, *The knowledge web: from electronic agents to Stonehenge and back–and other journeys through knowledge*. Simon and Schuster, 2000.

[81] K. Fischer, M. Schillo, and J. Siekmann, "Holonic multiagent systems: A foundation for the organisation of multiagent systems," *Holonic and Multi-Agent Systems for Manufacturing*, pp. 1083–1084, 2004.

[82] C. Gerber, J. Siekmann, and G. Vierke, "Holonic multi-agent systems," 1999.

[83] G. Boella and L. van der Torre, "Norm governed multiagent systems: The delegation of control to autonomous agents," in *Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on*. IEEE, 2003, pp. 329–335.

[84] F. Dignum, "Autonomous agents with norms," *Artificial Intelligence and Law*, vol. 7, no. 1, pp. 69–79, 1999.

[85] M. Ulieru, R. Brennan, and S. Walker, "The holonic enterprise: a model for Internet-enabled global manufacturing supply chain and workflow management," *Integrated Manufacturing Systems*, vol. 13, no. 8, pp. 538–550, 2002.