Aix-Marseille Université

Ecole Doctorale en Mathématiques et Informatique de Marseille - ED184

THESE CIFRE

Présentée par :

Sara BOUZID

Pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITE D'AIX-MARSEILLE

Faculté des Sciences et Techniques de Saint-Jérôme

Discipline : Informatique

Approche Sémantique de Gestion de Ressources d'Information pour le Contrôle de Processus Industriels: Application au Processus de Fabrication chez STMicroelectronics

---

Thèse préparée au sein du Laboratoire des Sciences de l'Information et des Systèmes (LSIS) de Marseille en collaboration avec la société STMicroelectronics

Soutenue publiquement le 06 décembre 2013, devant le jury composé de :

M. Philippe Aniorté	LIUPPA (Université de Pau)	Rapporteur
Mme Christine Verdier	LIG (Université Grenoble I)	Rapporteur
M. Philippe Lahire	I3S (Université de Nice)	Examinateur
M. Mamadou Kaba Traoré	LIMOS (Université Clermont Ferrand II)	Examinateur
M. Patrice Bellot	LSIS (Université d'Aix-Marseille)	Examinateur
Mme Corine Cauvet	LSIS (Université d'Aix-Marseille)	Directeur de thèse
Mme Claudia Frydman	LSIS (Université d'Aix-Marseille)	Co-Directeur de thèse
M. Jacques Pinaton	STMicroelectronics	Encadrant industriel

LSIS - UMR CNRS 7296

A Semantic Approach for Resource Description and Retrieval for the Manufacturing Process Control: Application to the Process Control within STMicroelectronics

A mes chers parents

A mes frères

A mes amis Loïs et Nazik
qui m'ont fait l'honneur
d'être témoin à leur
mariage

Remerciements

Mes profonds remerciements vont tout d'abord à ma directrice principale de thèse, Madame Corine Cauvet. Je la remercie pour son investissement exceptionnel dans l'encadrement de ma thèse, sa rigueur scientifique et son professionnalisme. Ses conseils avisés et son soutien infaillible tout au long de ma thèse m'ont permis de garder le cap pour mener ce travail de bout en bout. La confiance qu'elle me témoigne depuis le début de ma thèse m'a beaucoup touchée, ainsi que sa bienveillance et sa gentillesse. Je tiens à ce qu'elle sache qu'elle a été une directrice de thèse remarquable.

Je remercie ma co-directrice de thèse, Madame Claudia Frydman, pour sa clairvoyance, son soutien et sa sympathie. Ses recommandations et remarques pertinentes m'ont bien aidée à avancer. Aussi, son expérience professionnelle et son sens du relationnel m'ont été d'une grande aide à plusieurs moments de la thèse.

Mes remerciements vont également à mon encadrant industriel, Monsieur Jacques Pinaton chef de l'équipe Process Control System chez STMicroelectronics, grâce à qui cette thèse a pu être financée. Je le remercie de m'avoir intégrée comme membre à part entière dans son équipe, de m'avoir formée au métier de l'entreprise et d'avoir soutenu mon travail. Son habileté, son sens des responsabilités ainsi que sa prévoyance m'ont beaucoup marquée. Je tiens aussi à le remercier pour les nombreux conseils qu'il m'a prodigués non seulement sur mon travail de thèse mais également sur mon avenir professionnel.

Je remercie énormément les rapporteurs, Monsieur Philippe Aniorté et Madame Christine Verdier, pour m'avoir fait l'honneur de rapporter mon mémoire de thèse et pour leur retour positif sur mon travail. Je voudrais aussi les remercier pour leur réactivité et leur disponibilité. Je remercie Monsieur Philippe Lahire et Monsieur Mamadou Kaba Traoré pour avoir accepté d'examiner mon travail de thèse et de faire partie du jury. Mes sincères remerciements vont également à Monsieur Patrice Bellot, Professeur et chef de l'équipe DIMAG au LSIS, qui me fait l'honneur de faire partie de mon jury de thèse.

Je tiens à remercier tout le service Process Control de STMicroelectronics qui m'a accueillie chaleureusement, à commencer par la chef de service, Madame Maryvonne Le galle. Je remercie tous les membres de l'équipe Process Control System avec qui j'ai partagé trois merveilleuses années riches en apprentissage à la fois sur le plan professionnel et humain. Je remercie en particulier Laurent, Eric, Gloria, Mohamad, Mohamed, Alexis et enfin Franck qui m'a aussi encadrée durant ma première année de thèse. J'ai une pensée particulière pour mon collègue Michel qui m'a beaucoup aidée durant ma thèse, par ses compétences techniques et son expérience professionnelle. Je le remercie beaucoup pour toute l'aide précieuse qu'il m'a apportée durant ma thèse.

Mes remerciements vont également à tous les membres de mon équipe DIMAG du LSIS et au personnel administratif, en particulier Béatrice Alcala, Valérie Mass et Sandrine Dulac. Mes remerciements les plus sincères s'adressent aussi aux doctorants du LSIS avec qui j'ai partagé des moments de joie et de réconfort. Je pense en particulier à mes amies Lama, Shereen, Sana et Myriam qui m'ont encouragée et soutenue moralement. Je remercie aussi Ali, Rabah, Talal, Pamela, Laura, et plus particulièrement Mehdi pour m'avoir réconfortée durant ma dernière année de thèse.

Enfin, j'adresse de très grands remerciements à mes parents qui ont toujours été ma source de motivation et à mon frère Said qui m'a été d'un très grand soutien moral. Je tiens à témoigner ma gratitude envers mon frère Abdellah pour avoir été patient, compréhensif et serviable pendant notre période de cohabitation durant ma thèse. Qu'il sache que sa sagesse et son humilité m'ont beaucoup touchée.

Résumé

Afin d'assurer la fabrication de produits conformes et à faible coût dans les industries, la maîtrise des procédés de fabrication avec des méthodes fiables et des indicateurs standards de suivi est devenue un enjeu majeur. Les systèmes d'information dans les industries sont assez complexes et les besoins métier évoluent en permanence, rendant ainsi difficile la recherche de ressources qui fournissent les informations manufacturières pour le suivi et l'analyse des procédés industriels. De plus, l'utilisation de plateformes logicielles commerciales dans les industries pour le traitement des données ne facilite pas l'accès à l'information produite car ces plateformes ne permettent pas la gestion sémantique de l'information.

Cette thèse défend l'idée qu'il faut réduire la « distance » entre les ressources disponibles et les besoins métier des experts qui assurent le contrôle de processus industriels.

L'approche S3 est proposée pour supporter la maîtrise des procédés de fabrication grâce à un système de gestion de ressources d'information manufacturière. Il s'agit d'une approche sémantique qui permet à la fois la description et la recherche de ces ressources. L'approche S3 repose sur deux stratégies de recherche complémentaires: une stratégie de type ascendante (« bottom up ») permettant la création de descripteurs sémantiques de ressources, et une stratégie de type descendante (« top-down ») permettant la capture des besoins métier dans des patterns de recherche. Deux structures sémantiques sont proposées pour supporter les mécanismes de description et de recherche: une ontologie « manufacturing process » et un dictionnaire « process control ». Chaque stratégie de recherche, appuyée par les structures sémantiques apporte un niveau de description différent et permet l'alignement de différents types de connaissances métier. Cette approche a été expérimentée au sein de l'entreprise STMicroelectronics et a révélé des résultats prometteurs.

Mots clés: Recherche sémantique, Maîtrise de procédés, Processus de fabrication, Ontologie métier, Pattern d'alignement

Abstract

In order to ensure the manufacturing of conforming products with the least waste, the manufacturing process control has ever more become a major issue in industries nowadays. The complexity of information systems in industries and the permanent evolution of the business needs make difficult the retrieval of the resources that provide manufacturing information related to the process control. In addition, the use of commercial software platforms in industries for the processing of data does not facilitate the access to the information produced, because these platforms do not support the semantic management of information.

This thesis argues the need to reduce the distance between the used resources in industries and the business needs of the experts that ensure the control of the manufacturing processes.

The S3 approach is proposed to support the control of the manufacturing processes through an original resource management system. This system is intended for both resource description and retrieval. The S3 approach relies on two complementary retrieval strategies: a bottom-up strategy enabling the creation of semantic descriptors of resources, and a top-down strategy enabling the capture of business needs in search patterns. Two semantic structures are proposed to support the resource description and retrieval mechanisms: a manufacturing process ontology and a process control dictionary. Basing on these semantic structures, each retrieval strategy provides different levels of description to the resources and enables the alignment of different types of business knowledge. The experimentation of the approach within STMicroelectronics showed promising results.

Key words: Semantic retrieval, Process control, Manufacturing process, Business ontology, Alignment pattern

Table of Contents

List of Figures

List of Tables

General Introduction

1. Problem Statement

Industrial companies have an increasing need of controlling their manufacturing processes with reliable manufacturing information in order to ensure the fabrication of conforming products. The process control is a set of tools and methods established to guaranty that the process resources and activities are monitored, are consistent and conform to production objectives. Indeed, the quality of an end product can be significantly impacted during a manufacturing process according to factors such as the proportion of one ingredient to another, the temperature of materials, etc. Controlling the process variables guarantees the manufacturing of conforming products with the least waste [Schippers 2000; PAControl 2006]. Improving the access to manufacturing information for the process control actors is necessary in this context. Most industries use commercial software platforms for engineering data management and analysis, mainly because they facilitate the access and exchange of manufacturing data among company engineers. However, the easy access and use of these platforms can rapidly entail information overflow and resource retrieval difficulties, in particular because the produced resources that contain the processed data always lack of semantics. Indeed, a manufacturing information resource can take the form of business indicators, business data, figures, data charts, etc. This form of resources is specific to the process control activities as used in industries. It is almost impossible for a business actor or any retrieval system to find this type of resources without descriptive meta-data or semantics. This lack of semantic description often engenders a significant gap between the business needs of the process control actors and the resources that meet their needs. Addressing this gap is capital to improve resource retrieval in industries, so to efficiently support the control of manufacturing processes.

This research work has been conducted with the STMicroelectronics Company, a French-Italian semiconductor manufacturer. The scope of the problem statement and the contribution of this work were set up by studying the process control of this company.

2. Objectives

This thesis aims at reducing the distance between the used resources in industrial companies and the business needs of the actors that ensure the control of the manufacturing processes. The lack of semantics in the resources is one of the main issues to tackle in order to reduce this distance. Also, the required semantics must bring a business-oriented description to the resources to enhance their retrieval. Considering the context of the process control and the resources used in this case, we consider that improving the resource retrieval must necessarily include the alignment of the low-level description of the resources with the high-level business needs of the business actors.

Ultimately, this work must deal with several objectives, as follows:

- it must address the lack of resource description by bringing the appropriate semantics to the existing resources in the company used for the manufacturing process control
- it must align the complex business needs of the business actors with the resources
- it must improve resource retrieval and the sharing of same needs among the business actors
- it must deal with the heterogeneity of the business vocabulary in the company
- it must enhance the retrieval of both existing resources and the new ones for a continuous manufacturing-process control

3. Contribution

We propose in the contribution of this thesis a semantic approach that we call S3, intended for both resource description and retrieval in the context of the manufacturing process control. S3 relies on two retrieval strategies: a bottom-up strategy and a top-down one. These strategies are complementary and contribute in filling the gap between the resources and the business needs related to the manufacturing process control.

The **bottom-up strategy** aims at building **semantic descriptors** of existing resources in a company through a mapping technique. The semantic descriptors associate the business descriptions that enable to identify the usage and role of the resources in the manufacturing process control. The type of description in these descriptors is identified with the help of the process-control experts of the company. A **mapping technique** is proposed to automate the creation of the semantic descriptors. This technique is based on two mechanisms: string-similarity computation and inference mechanisms.

The **top-down strategy** aims at capturing the business needs of the business actors and aligning them with the resources using **search patterns**. We distinguish two types of search patterns: **query patterns** and **alignment patterns**. The query patterns capture the user queries using keyword(s), goals and contexts. The solutions to such queries require the creation or the reuse of alignment patterns starting from the goal and context of each query pattern. The alignment patterns capture artifacts of business needs using goal-oriented mechanisms. The solutions of each set of alignment patterns lead to the resources that meet the user need. Several alignment patterns can be created and stored during resource retrieval for further use. At last, by combining a keyword search with alignment patterns, we put the business needs of the business actors in the heart of the retrieval process in a way to retrieve the relevant resources to their needs.

Two semantic structures are used to support the resource description and retrieval in the approach: a **manufacturing process ontology** and a **process control dictionary**.

The manufacturing process ontology provides a high level description of the manufacturing process according to four views of description: a function view, an organization view, a data view and a control view. The process control dictionary gathers and streamlines the vocabulary related to the process control activity. The proposed structures are semantically linked through the control view of the ontology. They are used to provide the required semantics for the creation and enrichment of the semantic descriptors and the search patterns on one hand, and to support the semantic alignment of the business needs with the resources during their retrieval on other hand.

The S3 approach is being implemented and tested in the STMicroelectronics Company using a specific implementation prototype based on the Topic-Maps paradigm [Pepper 2008]. This application offers three search functions: a resource mapping function which creates and enriches the semantic descriptors of resources, a pattern-based search function which creates and enriches the alignment patterns, and a Topic-Maps-based search which enables to explore the resource description through knowledge maps. This prototype is still at an experimental stage today in the company.

On the whole, the models, structures and techniques resulting from of this research work mainly consist of:

- a manufacturing process ontology providing a business-process-oriented description of the semiconductor industry (which is our case study)
- a process control dictionary that consolidates the process-control description of a manufacturing process
- a semantic mapping technique combining string-similarity algorithm and inference mechanism
- a meta-model of search patterns that captures business needs from a high-level to a low level
- a Topic-Maps-based application for resource description and retrieval.

4. Structure of the Thesis

This manuscript is organized into seven chapters, as follows:

- Chapter 1 presents the context of work and the problem statement related to the manufacturing process control in industries. We introduce in this chapter two knowledge domains, used as knowledge sources to enrich resource description. These two domains are the *manufacturing process* and the *process control*,
- Chapter 2 presents the state of the art of this work. Two research domains are studied: the semantic retrieval of heterogeneous resources and components on one hand, and the alignment-oriented approaches on other hand,
- Chapter 3 gives an overview of the S3 approach and lists the main findings of this work
- Chapter 4 presents the content of the manufacturing process ontology and the process control dictionary used as support for resource description and retrieval in the approach
- Chapter 5 presents the semantic descriptors of resources and how they are built with the semantic mapping technique
- Chapter 6 presents the meta-model of search patterns and explains how the patterns are used to capture and align the business needs with the resources during a search process
- Chapter 7 gives an overview of the application prototype used to implement the S3 approach. Some experimentations of the approach done with the prototype are also presented

Chapter 1: Context of Work

Résumé

Le contrôle de processus industriels est une tâche complexe qui demande la collaboration de plusieurs acteurs métier pour assurer la fabrication de produits conformes aux spécifications des clients, et donc assurer la maîtrise des procédés. Le travail quotidien des acteurs du métier dans ce contexte repose principalement sur le partage et la recherche de ressources contenant des informations et des données sur l'activité manufacturière. L'utilisation de plateformes logicielles commerciales dans les industries pour la production de ces ressources rend le travail des acteurs encore plus difficile car ces plateformes ne permettent pas une gestion sémantique des informations et ressources produites. Pour faciliter le travail des acteurs du métier, il est important de supporter le contrôle des processus industriels en améliorant l'accès à ces ressources.

Ce chapitre introduit ainsi le contexte de travail de cette thèse et la problématique de recherche. Nous montrons l'importance d'améliorer la recherche de ressources d'information manufacturière en industrie à travers l'étude du système « process control » de l'entreprise STMicroelectronics. En étudiant l'énoncé du problème, nous avons pu identifier deux objectifs scientifiques à réaliser pour améliorer l'utilisation des ressources: (i) enrichir la description des ressources en utilisant la connaissance à la fois sur le processus manufacturier et sur le « process control », et (ii) réduire la distance entre les ressources disponibles et les besoins métier.

1.1. Introduction

We introduce in this chapter the problem statement of this work related to the process control in industries and the resource retrieval difficulties. We show the importance of improving the retrieval of manufacturing information in this context through the study of the process control of the STMicroelectronics Company. Finally, we conclude this chapter by analyzing the problem statement compared to standard retrieval issues, and we identify the main research issues of this thesis.

1.2. Problem Statement

1.2.1. The Process Control in Industries

[Wheeler and Chambers 1992] defines the process control as «*the methods that are used to control process variables when manufacturing a product*».

In fact, the process control is a set of tools and methods established to guaranty that the process resources and activities are monitored, are consistent and conform to production objectives. Indeed, the quality of an end product can be significantly impacted during the production process according to factors such as the proportion of one ingredient to another, the temperature of the materials, how well the ingredients are mixed, and so on. Thus controlling the process variables guarantees the manufacturing of conforming products with the least waste. The process control is particularly important when the business process is complex or automated at large scale, because the cycle time is long and not necessarily linear.

If we take as example the semiconductor industry, the manufacturing of a product has an average of 600 steps, corresponding to a long cycle time of 8 weeks. The semiconductor products are manufactured in lots, where each lot is a set of silicon wafers used as support for the construction of integrated circuits. The manufacturing process spread over several work areas (*Photolithography*, *Etching*, *Implantation*, etc). Each work area consists of sequences of operations and steps that can be remade several times. According to this configuration of complex mass production, the manufacturing cannot be then efficiently ensured without a regular monitoring and control. Basically, manufacturing companies control the production processes for three reasons [PAControl 2006]:

- to reduce the variability of equipment in order to minimize production waste
- to increase the efficiency of production by improving the process
- and to ensure safety with the regular monitoring of the process

Main process control methods are based on statistical and engineering techniques. The widespread used control method is the Statistical Process Control (SPC) [Wheeler and Chambers 1992]. SPC is the application of statistical techniques to the monitoring and control of a process to ensure that it operates at its full potential to produce conforming products. SPC was the first process control method created by Walter A. Shewhart in 1920s and introduced by Edwards Deming in USA and to Japanese firms after the World War II.

The main idea of SPC relies on controlling the **variability**. In mass manufacturing, the quality of a product is ensured by post-manufacturing inspection of the product (also called quality control). Each product may be accepted if it well meets its design specifications or rejected ("scraped") in the other case (Figure 1.2-1). In contrast, SPC uses statistical techniques to monitor the achievement of the production process in order to predict significant variations which may result in the manufacturing of a product. A source of variation at any one point of a production process could be "common" or "special": common because a manufactured product is always subject to a certain amount of variations, and special when the variation is caused (accidently) by other factors that could not be anticipated. Common causes are usually known and inevitable, so they can be controlled. Special causes must be discovered with process monitoring in order to be addressed. In this way, the SPC method is based on a prevention strategy to avoid the manufacturing of unusable products.

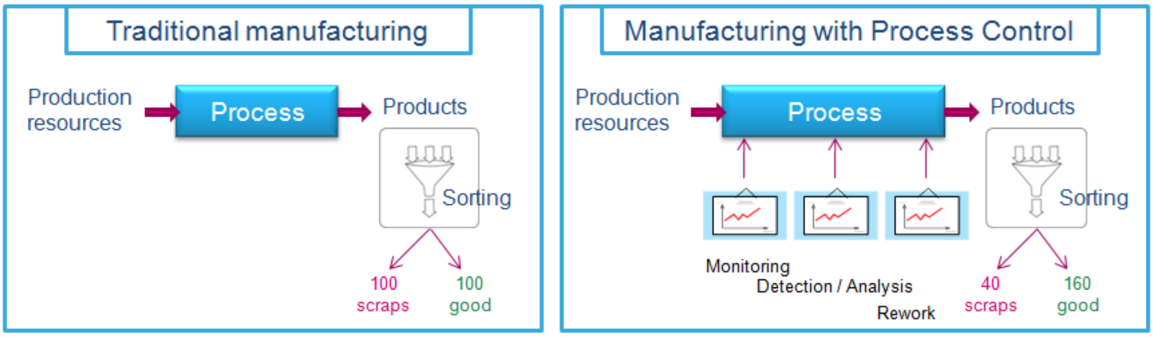


Figure 1.2-1 : The manufacturing with process control comparing to a traditional manufacturing

SPC relies on a number of (statistical) techniques to produce control indicators. The main used are:

- **Control charts** (also called Shewhart charts): a graphical representation of chronological events with upper ("UCL") and lower ("LCL") limits. When we get measured data outside these **limits of control**, we say that the product is "**out of control**". Basically, the control charts are used to detect the variability of special causes and to reduce the variability caused by common causes
- **CUSUM charts**: a graphical representation that takes as data the cumulative sum of quality characteristics. CUSUM charts are used to detect more sensible variations
- **Pareto charts**: are usually used to observe the correlation between measures to determine problem causes (regarding defects for example), predominant root causes, etc.
- **Capability index (CpK)**: is represented with a flow chart that predicts the ability of the process to produce outputs within specification limits. In fact, the capability index is based on measuring the variability of the output of the process and comparing that variability with the product specification. Thus, when a product is outside the upper ("USL") and lower ("LSL") **specification limits**, it is considered "**out of specification**".
- **Regression models**: focus on analyzing the relationship between dependent and independent variables to predict effects among the production process variables.

SPC is considered as part of the Advanced Process Control (APC). APC encompasses a practice which draws elements from many disciplines, ranging from control engineers, signal processing, statistics, decision theory and artificial intelligence [Golo 2011]. The most implemented methods in manufacturing companies (including SPC) are:

- **FDC** (Fault Detection and Classification): is an equipment control approach and system, based on multivariate data analysis to detect faults in the step by step evolution of the product fabrication. It enables to reduce scraps (defect products) and improve equipment performances

- **R2R (Run to Run)**: is an equipment monitoring tool aiming at improving process consistency by optimizing recipe parameters of a product between two machine “runs”. It uses feedback from process models, incoming variations and metrology data
- **MPO** (Manufacturing planning and optimization): works on how to arrive at optimal operating targets in the ongoing process activity. The optimal targets are, afterwards, implemented in the operating organization to improve the manufacturing process
- **Simulation-based optimization**: incorporates computer-based process simulation models to determine more optimal operating targets in real time and on a periodic basis ranging from hourly to daily (often considered as part of MPO)

The process control methods also rely on problem solving techniques, such as:

- **FMEA** (Failure Mode and Effect Analysis): is a failure analysis technique. It is mainly based on a qualitative analysis (e.g. experiences with similar products, failure logic, etc.) to study the failure modes and the potential problems that may arise in a process
- **8D**: is a structured problem solving approach propagated by Ford in the automotive industry. The approach is composed of 8 steps called “Disciplines”, which follow the logic of the **PDCA** (Plan, Do, Check, Act) cycle [Chardonnet and Thibaudon 2003] to resolve a problem
- **Ishikawa (cause-and-effect) diagrams**: is a problem solving technique (illustrated with a fish skeleton) created by Kaoru Ishikawa in 1968 to identify potential factors causing an overall effect or problem.

We can also find other methods or techniques defined by the manufacturing companies to respond to their specific control needs. Basically, in order to ensure the control task, most of control methods use a set of indicators, where an indicator is an aggregate of measures or data reprocessed with statistical methods. The control indicators give information about how the process runs.

The process control is applied in a manufacturing company by collecting (real time) data related to raw materials, production machines, products and manufacturing methods. The collected data are then processed and analyzed using software platforms.

Technically, the process control is mainly implemented with specific market platforms that are usually used to generate the required resources (indicators, data sheets, statistical components, etc.) for the control of the manufacturing process. Each company may use different software platforms to support the setting up of the process control, depending on its IT strategy.

1.2.2. Lack of Resource-Retrieval Support for the Process Control

A process control system is mainly implemented in industries through several market platforms, called **COTS (Commercial Off The Shelf)** software [Morisio et al. 2002]. COTS are software platforms and tools, designed for specific applications such as medical billing, chemical analysis, engineering data analysis, and so forth. They are usually implemented with some adaptations only. Acquiring such systems enables the companies to outsource the support of their non core-business tasks. Moreover, the modular and distributed aspects of these systems allow easy adaptation to changes and reduce long-term maintenance costs.

COTS platforms have user-friendly interfaces which offer several functionalities for data processing. The produced outputs are resources that give information for the control activity, such as monitoring indicators, data analysis with graphical representations, comparison of data populations, etc. The usability of these tools enables an easy access to data to the process control actors (i.e. engineers and technicians), but entails on other hand resource overflow and retrieval difficulties. The resulting COTS' resources essentially contain figures (e.g., control charts, histograms, etc.) and business data (data tables and sheets) which lack of semantics. Yet, sharing and exchanging manufacturing information among the process control actors is necessary to guarantee the well application of the process control in industries. Searching for such resources is almost impossible without knowing their business usage.

Indeed, an effective manufacturing information system must enable the easy access to data and the reuse of same resources by the users that have same needs in an activity domain. However, existing commercial platforms intended for industries only focus on the easy and user-friendly aspects, neglecting the knowledge sharing and information search aspects.

Hence, because of the lack of a semantic support of the COTS' resources in industries, many difficulties can be entailed:

- users can spend a lot of time in searching for a resource or instead re-develop from scratch new resources for existing needs
- when the users re-develop same needs, they can entail resource overflow in the company network in the long term. The resource retrieval becomes ever more difficult
- the COTS software often have their own vocabulary which is a bit different from the specific industries' business vocabulary. This heterogeneity in the vocabulary can lead to other retrieval difficulties and to misunderstandings in interpreting the processed data
- all knowledge and know-how around the business data and the process control experience remain in the human brain
- in case of lack of knowledge transfer, all knowledge around the resources will be lost

Finally, one other difficulty and constraint related to the process control context is that its implementation is specific to each industry and activity domain. In this case, the used COTS software and process control methods may differ in their use from a company to another.

To support our statements, we studied the process control system of the STMicroelectronics Company.

1.3. The STMicroelectronics' Context as Real Case Study

STMicroelectronics is a French-Italian semiconductor company, headquartered in Geneva, Switzerland. The company operates a worldwide network of front-end (i.e. chips' fabrication) and back-end (i.e. assembly, test and packaging). It has 12 manufacturing sites, including five in Europe and three in France (Rousset, Grenoble and Tours).

This research work was conducted in the Rousset site (near Aix en Provence).

1.3.1. Construction of Electronic Chips: Some Business Vocabulary

STMicroelectronics' products are manufactured in **lots**, where each lot consists of a set of silicon **wafers** (Figure 1.3-1) that serve as support for the construction of integrated circuits. Each area of a single complete integrated circuit is called a die. After the assembly process, a packaged die is referred to as an electronic chip.

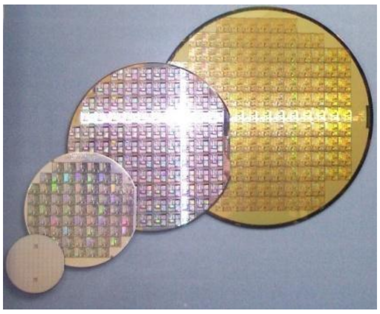


Figure 1.3-1 : Examples of wafers

The front-end semiconductor manufacturing process consists of hundreds of **manufacturing steps**, grouped somehow into **work areas**, in which the wafers are processed by adding layers of materials that are patterned into integrated circuits. Examples of standard manufacturing steps include *Oxidation*, *Photolithography*, *Etching*, *Implantation*, *Defect inspection*, etc.

The semiconductor process is a complex process where each work area[1] consists of many steps that can be remade several times. Thus, to organize the process, STMicroelectronics defines **routes** of production for its products. Each set of routes is associated to one given **technology**. A technology refers to the production techniques of the integrated circuits that have same characteristics[2]. Each route of production is composed of a sequence of ordered **operations** and a set of steps to follow in these operations. In fact, during the wafer processing in a given work area, such as the photolithography, the wafer is processed on several operations and steps, before being moved to other steps of other work areas.

Finally, the manufacturing of a semiconductor product has an average of 600 steps, corresponding to a long cycle time of 8 weeks. According to this configuration of mass production rather complex, with several non-linear stages and process automation at large scale, STMicroelectronics had set up, many years ago, the process control of its manufacturing activity, to ensure the well achievement of its manufacturing process and to continuously improve it.

1.3.2. The Process Control within STMicroelectronics

STMicroelectronics applies the process control on its manufacturing process since many years. The process control plays three major roles in the company: detection, prevention and process improvement.

The detection is based on the set of indicators that allow monitoring the manufacturing activity at several checkpoints, so to make corrective actions at the right time to minimize the impact of a given problem on all the production line.

---

[1] The main work areas within STMicroelectronics are: *Photo & Metrology, Etching, Diffusion & Wet, CVD, BEOL*

[2] In the Rousset plant, we can find as technologies *90nm* (nanometers) and *130nm*, among others

The prevention enables to anticipate the problems that may occur during the production and to assess risks related to changes.

The process improvement includes the improvement of the used production methods and tools to ensure cost-effective manufacturing.

The process control function is integrated as business entity associated to the front-end manufacturing of the company. This entity is organized according to the main control activities conducted in the company. The process control has also a transversal function on the front-end manufacturing, such as each control activity has interlocutors in the work areas of the manufacturing process, to ensure the well application of the process control on the whole process.

The process control activities are grouped into four main entities within STMicroelectronics:

- **Wafer Fab Yield (WFY) and Robustness**: includes the SPC activity, Quality-Task management, production yield, 8D and process change
- **Process control systems**: regroups the FDC and R2R tools, equipment control and risk control tools
- **EDA (Engineering Data Analysis) and statistical analysis**: all statistical methods used to control the process
- **Defectivity**: includes equipment control and production-line monitoring and support

The process control activities are organized according to the used standard control methods or according to the purpose of use of the applied ad-hoc techniques. The standard process control methods used in the company are SPC, R2R and FDC. The ad-hoc techniques are grouped into the following control domains:

- **Sampling**: a statistical technique consisting in selecting a random or representative subset of lots during the manufacturing process for testing
- **Risk control**: include a set of techniques and indicators for the monitoring and assessment of risk of errors that may happen during the manufacturing process. the Wafer at Risk (WAR) approach is one of the most efficient used method within the company
- **Change management**: includes a set of techniques that help in setting up changes and measuring their impact on the manufacturing. Examples of techniques include the procedures for validating the changes in production routes, population comparison, etc.
- **Yield control**: refers to the set of indicators used to monitor the production yield, called the Wafer Fab Yield (WFY)
- **Equipment control**: encompasses the set of tools and indicators used to detect and monitor equipment problems through the equipment alarms
- **Defect control**: a set of techniques used to control the wafer defects (e.g. past experiences, lot history, equipment history, etc.)

1.3.3. Lack of Effective Process Control Support in the Company

STMicroelectronics uses COTS software platforms for the manufacturing process control and several other activities. The produced COTS’ outputs are resources that give information for the control activity, such as monitoring indicators, data analysis with graphical representations, comparison of data populations, etc. The usability of these tools enables an easy access to data for the process control actors (i.e. engineers and technicians), but entails on other hand resource overflow and retrieval difficulties. The resulting resources – that we refer to as ***Manufacturing Information (MI) resources*** – essentially contain figures (e.g., control charts, histograms, etc.) and business data (data tables and sheets). These resources are accessible of different manners in the company (e.g. directly from the COTS platforms, from shared disks, document repositories, etc.).

Among the used resources for the manufacturing process control (Figure 1.3-2), we can cite:

- *Kla Ace recipes*, where a recipe is specific workflow program that runs on a platform called "Kla Ace XP". These recipes provide statistical calculations and graphical representations of manufacturing data. A recipe has an owner format with an extension ".rcp". Some outputs can be obtained with standard formats such as .txt, .pdf, .csv, .xls, etc. (main SPC resources are made in the Kla Ace platform)
- *Klarity defect*, enables to get wafer images to analyze the manufacturing defects
- *APF recipes*, are specific workflow programs for data extraction and reporting created with a software platform called "APF". A web version of the APF platform enables to have data sheets in csv format
- *BO (Business Object) reports*, the produced reportings mainly have .pdf formats
- etc.

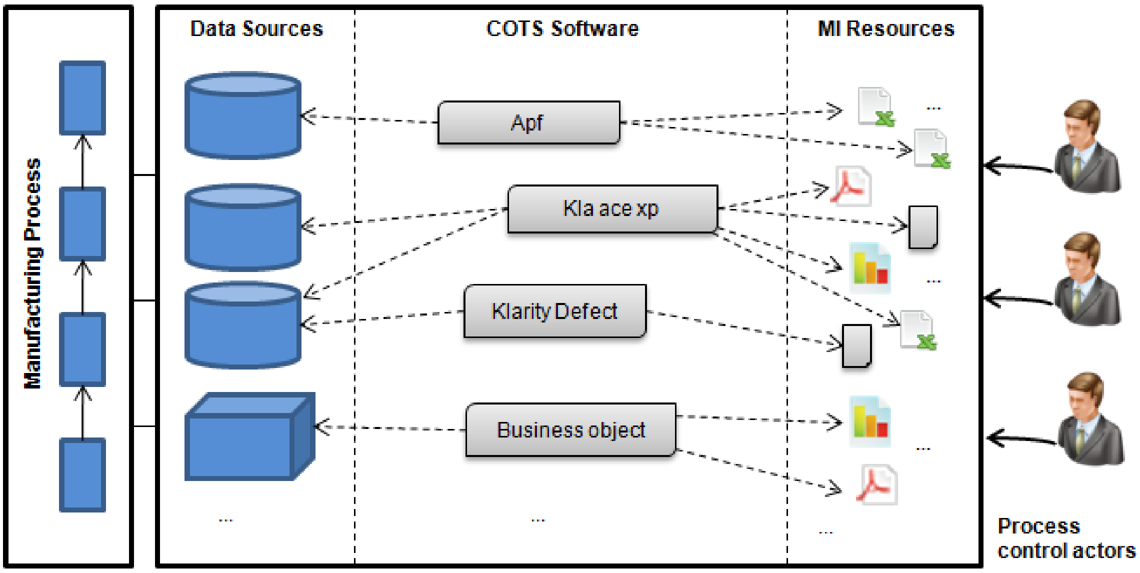


Figure 1.3-2 : Overview of the process control system of STMicroelectronics

Most of these resources are scheduled every day, week or month, meaning that the resources are regularly updated with new data from the manufacturing process. Other resources are usually created upon request for specific needs. Basically, we can categorize the MI resources[3] used for the process control into three kinds:

- software components (i.e. the software programs that have owner formats like workflow programs, executable files, etc.),

---

[3] In the rest of the manuscript, we use the term **MI resource**, to refer to a resource (of any format) that provides information (i.e. raw or processed data) about the manufacturing process

- documents (pdf, csv, xls and all standard file formats containing graphical representations and data sheets),
- and web resources (for the files that are accessible via web browsers like html files, web applications, etc.)

The study of the process control system of STMicroelectronics points out three major problems and lacks in using commercial software platforms.

- Resource overflow:

The COTS platforms were generally designed in a way to enable an easy access and processing of data with fewer considerations about how to manage the produced resources. Thus, an engineer can easily get and process data with the COTS platforms, but there is no way or approach to retrieve what he produced before or what other users produced as manufacturing information. The accessibility of these systems has rapidly entailed the increase of the MI resources in the company, leading then to retrieval difficulties.

- Lack of semantics and vocabulary heterogeneity:

The lack of business semantics in the resources makes difficult their findability within the company. Thus, because of this lack of semantics, the MI resources cannot be referenced for retrieval purposes.

In addition, most COTS platforms have their own business vocabulary mainly defined by the provider. The vocabulary heterogeneity also leads to semantic confusion and to a misunderstanding in interpreting the subjects of the resources by the end users. Also, we noticed that the users refer to the business names of the manufacturing data that are processed in the resources to understand the need addressed. However, the used concepts in the resources can have ambiguous meanings or do not appropriately express the business information handled by the resources[4].

To confirm these statements, we did a simple experimentation on a panel of users within STMicroelectronics. We proposed to a panel of 40 engineers which have different business profiles to try to find three different resources: *"control chart on lot"*, *"lot history"* and *"out of specs by products"*. These resources corresponded to three basic needs required for the core business activity of the process control. The results showed that only 3 users were able to find one appropriate resource among the three requested ones, 10 users were hesitant in their choice and the rest of the users didn't find them or have found non-corresponding resources.

- Diversity and complexity of business needs:

The business needs of the process control actors can be complex and related to a high-level business. The users cannot retrieve the resources that meet such complex and abstract needs without an appropriate retrieval system and technique. In addition, a resource can respond to several business purposes depending on the business profiles in the company. For example, an engineer associated to the work area "Photo" may need same extracted/processed data as an engineer from the work area "Etch", however they do not use these processed data for the same purpose. The diversity of use of the resources in industries must also be considered in the retrieval process.

---

[4] For example, the concept "operation" is called "level" in the software tool *Klarity defect*

In summary, due to the lack of a real resource retrieval support in industries, the manufacturing process control may be not efficiently guaranteed. The lack of semantics in the MI resources seems to be the key aspect to tackle, in order to provide a first basic support to the process control task.

1.4. The Research Issues

To ensure the monitoring and control of the manufacturing process in industries, it is important to give the stakeholders the efficient means to access to the resources used for this purpose. However, the lack of a real management support of the MI resources in manufacturing companies necessarily leads to resource retrieval difficulties. In fact, in a traditional retrieval system, the user queries (e.g. keywords, queries in natural language, etc.) are matched with data available in information sources such as database records, data in textual resources, meta-data associated to software components and so forth. In case of MI resources, the matching process is more complicated because of their heterogeneous formats, on one hand, and of their lack of meta-data (either in the resources or apart) on other hand. In fact, it is very difficult to perform effective search in information sources without having at least a bit of description about their usage.

Figure 1.4-1 shows the difference between a standard retrieval process and the retrieval process in case of resources handling manufacturing information (MI resources).

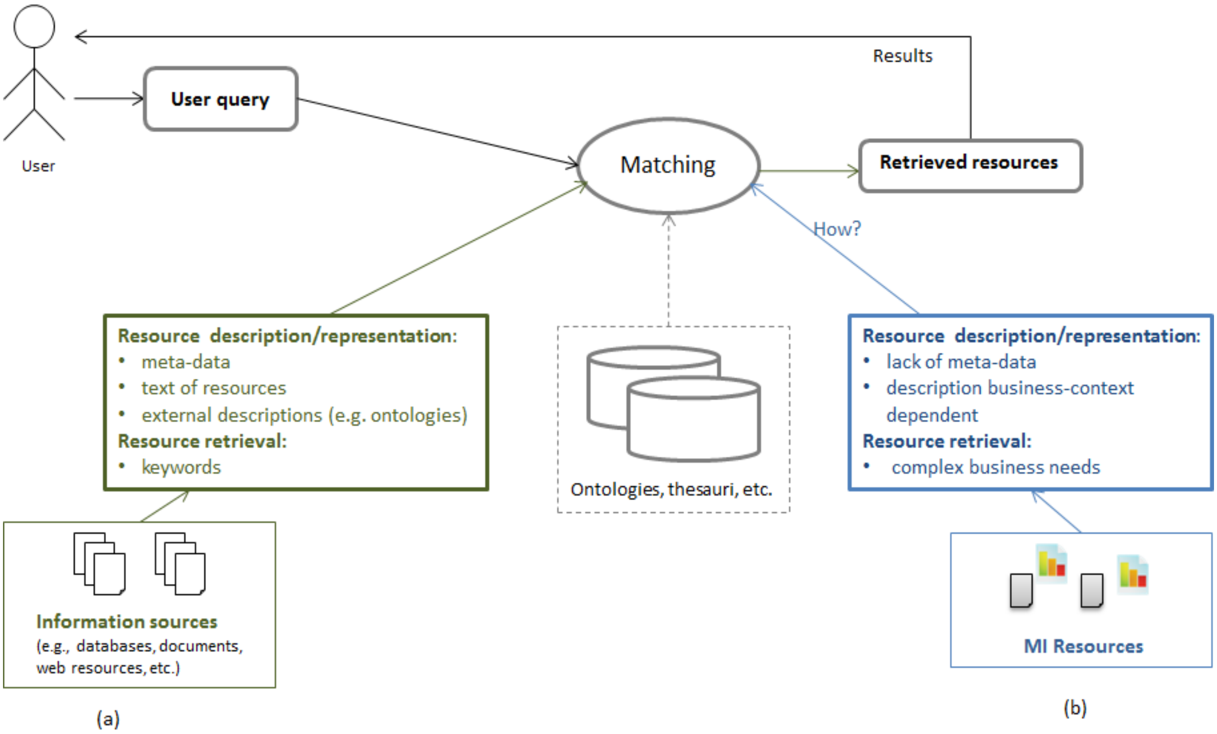

User
User query
Results
Matching
Retrieved resources
How?
Resource description/representation:
• meta-data
• text of resources
• external descriptions (e.g. ontologies)
Resource retrieval:
• keywords
Ontologies, thesauri, etc.
Resource description/representation:
• lack of meta-data
• description business-context dependent
Resource retrieval:
• complex business needs
Information sources
(e.g., databases, documents, web resources, etc.)
MI Resources
(a)
(b)

Figure 1.4-1: Standard retrieval process (a) versus MI-resource retrieval (b)

- <u>Case of a standard retrieval process</u>

In a traditional retrieval system, the matching of the user query is often done with the textual data and meta-data of information sources. In the recent developments of retrieval systems, ontologies [Gruber 1993] and other semantic sources (e.g. thesauri, lexicons, etc.) are used to enhance the retrieval process and render more accurate results.

The keyword search is the most used retrieval technique in Information Retrieval with the ontologies to provide semantic search. Ontologies are also used nowadays to address other issues in Information Retrieval, in particular to assist the user in expressing his query, to allow navigation along concepts to refine the user need, etc.

The information-source description is generally translated into a (semantic) representation to enable the matching with the user query. Typically resource representations are obtained by extracting keywords that are considered as content identifiers and organizing them into a given format. Query formatting into a representation depends on the underlying model of retrieval used.

- Case of MI-resource retrieval:

The MI resources contain raw data or processed in the form of business indicators, intended to meet business needs. Due to their lack of semantics, there is no way to know their business usage or which need they address, and thus, the search system cannot match the user query with the MI resources. Also, such resources cannot be easily matched with user queries even by using external semantic sources because the resource description is business-context dependent, as it is highly related to user needs in a business context. We can then notice a gap between the resources to retrieve and the business needs of the end users. As a result, setting up a resource retrieval support must necessarily address this gap, in order to ensure the effective resource search and retrieval over time. Bridging this gap implies rendering the MI resources close to the business needs, by capturing their business usage (i.e. the business goals they address).

Indeed, in a standard search system, the user query is often limited to keywords because it is the easiest way for the user to formulate his need. However, a keyword search is not necessarily suited to MI-resource retrieval because huge quantity of results can be obtained, besides that the business needs of the users are often complex and related to their assignments in the company. Furthermore, the users' needs are closely related to their business function in industry and this aspect should also be taken into account in the formulation of the user query. As a consequence, we can state that enhancing resource retrieval in this case should first tackle the high-level needs of the users and how to align the MI resources to these needs. Also, harmonizing the vocabulary heterogeneity in expressing the business needs is certainly worthwhile to consider, so to efficiently improve the resource retrieval.

We can conclude that the problem of resource retrieval in this thesis is not only related to the lack of description in the resources, but also a problem of distance between the MI resources and their business usage. We estimate that by filling the gap between different types of knowledge —from the resources to the business needs—, we can better ensure the MI-resource retrieval in industries.

To sum up, this research work must address the following research issues:

i. How to describe the existing resources in the company in a way to align them with the business needs?
ii. How to enhance their retrieval with respect to the process control purpose in the industry?

iii. How to deal with the heterogeneity of a business vocabulary that involves two distinct (but interrelated) knowledge domains (the manufacturing process and the process control)?

iv. How to propose a solution that deal with both existing resources and the new ones (a scalable solution)

1.5. Conclusion

We have seen in this chapter that the semiconductor manufacturing is a complex process that needs to be continuously controlled and improved for its well achievement. Facilitating the access to the manufacturing information for the process control actors is momentous to ensure an under-control production. Nevertheless, the generalization of the commercial software platforms in industries to get manufacturing data and schedule their processing can rapidly entail the increase of the quantity of the produced MI resources as observed within the STMicroelectronics Company. Retrieving such resources by the process control actors often seems impossible, in particular because they always lack of semantics. To deal with this problem statement, this work must necessarily rummage through the semantic retrieval approaches which rely on knowledge representation structures, such as ontologies. However, the main challenge in using a semantic approach in our context is to also bridge the gap between business needs and MI resources, to efficiently deal with in the retrieval purpose.

Chapter 2: State of the Art

Résumé

Ce chapitre présente l'état de l'art réalisé dans le cadre de ce travail de thèse. Deux domaines de travaux scientifiques sont étudiés : le premier réunit différentes approches sur la recherche sémantique, et le deuxième s'intéresse aux travaux d'alignement dans les systèmes d'information.

Diverses approches dédiées à la recherche sémantique existent de nos jours. Nous avons limité l'état de l'art à des approches s'inscrivant dans notre problématique. Ainsi, les approches de recherche de ressources portant sur des informations manufacturières et d'ingénieries ont été retenues en premier, en raison du type des ressources étudiées et du caractère à la fois scientifique et industriel des solutions apportées. Les approches dédiées à la recherche de composants logiciels et de services ont également été étudiées. Dans les deux cas, les techniques utilisées pour exprimer la sémantique sont abordées dans ce chapitre, ainsi que les techniques de recherche sémantique.

Les travaux d'alignement ont été étudiés dans l'objectif d'explorer les techniques et mécanismes utilisés pour réduire la distance entre les besoins métier et les solutions logicielles en entreprise.

Finalement, l'étude de cet état de l'art nous a permis de déceler un manque de travaux scientifiques dans le domaine de la recherche de ressources d'information dans le contexte d'un métier particulier. Nous avons pu conclure de cet état de l'art que la solution à apporter doit se baser sur des mécanismes de description et de recherche sémantiques mettant les besoins métier au cœur de la problématique de recherche.

2.1. Introduction

The retrieval issue of manufacturing information resources in industries has become an emerging field of study. Identifying the scope of the state of art of this work requires, in this case, understanding the meaning of a piece of manufacturing information and a resource, and why there is a difficulty to retrieve such resources in industrial companies.

The meaning of the term information varies according to different contexts. In Information Systems, the term information has been extensively contrasted with the terms data and knowledge [Caussanel 2000; Chu 2003]. According to our context of work, we mean by the term manufacturing information any piece of information, piece of knowledge that refer to manufacturing data, either in a raw format, or treated and transformed into business knowledge. Examples of manufacturing information include manufacturing indicators, alphanumerical data about the manufacturing process, data charts and so forth. Their format may differ according to the used manufacturing systems. We call then a Manufacturing-Information (MI) resource, any file, component or document that handles a piece of manufacturing information. These kinds of resources essentially contain figures and business data which lack of semantics.

The widespread used techniques to improve resource retrieval nowadays are semantic approaches. Semantic description techniques consist of a set of standards and approaches for knowledge representation based on the ontology paradigm and semantic web technologies. Examples of these technologies include the W3C standards (e.g., OWL, RDF, DARPA, etc.) and the ISO Semantic Web standards (e.g., Topic Maps, Dublin Core). They are used in several computer science disciplines to address sharing and retrieval issues and interoperability among software applications. First semantic techniques have been applied to web search, to improve information retrieval and knowledge sharing. With the emergence of ontology paradigm, these techniques have shown interest in other fields such as in document engineering and in the engineering of software components and services.

The main field of study that we found relevant to our problematic is the approaches related to engineering-document retrieval. Engineering documents contains data, knowledge and figures related to the design process of products in industries. Hence, to address the lack of research works in this field, we extended the study of the state of the art to information-resource retrieval and component retrieval. Moreover, as seen in the context of work (cf. Chapter 1), the semantic alignment of knowledge during the resource retrieval is another aspect to consider because of the distance between the MI resources and the needs they address. Alignment research-works aim at reducing the gap between the business environment and the IT (Information Technology) environment. Alignment techniques and approaches may relate to Information-System architectures, business processes, software applications, domain knowledge, etc. These approaches also tend to use semantic techniques to enhance the link between knowledge of different environments.

To sum up, the state of art of this work will focus on the following two fields: ***the semantic approaches for resource retrieval*** and ***the alignment-oriented approaches***.

The study of the first domain enables to identify the used techniques to enhance resource retrieval in enterprises using ontologies and other semantic structures. We include in this domain some engineering-document retrieval approaches and component/service retrieval approaches that we consider close to our purpose of work. The used techniques to build and structure the semantics will be investigated as well as the semantic retrieval techniques. The *alignment-oriented approaches will be explored to have an idea about the role of the semantics in reducing the gap between business knowledge and the software resources of a company.*

2.2. The Semantic Retrieval of Resources

2.2.1. Around Semantics, Semantic Web and Ontologies

In linguistics, Semantics is defined as the study of meaning [Moore 2000]. It focuses on the relation between signifiers, like words, phrases, signs, and symbols, and between what they stand for. In Computer Science, the term semantics refers to the meaning of programming languages, as opposed to their syntax[5]. The semantics provides here an interpretation of a computer-science expression [Euzenat 2007a; Euzenat 2007b]. The term semantics is also applied to certain types of data structures specifically designed and used for representing information content. They are known as semantic networks or semantic data models. They are used to describe particular types of data models characterized by the use of directed graphs in which the nodes represent concepts or entities of the real world, and the edges denote relationships between them.

The Semantic Web is a set of techniques and technologies for information description and knowledge representation. The semantic web is regarded as an integrator across different web contents, information applications and systems. According to the World Wide Web Consortium (W3C[6]), semantic web technologies provide a common framework enabling data to be shared and reused across applications and community boundaries.

Many standards have been developed over time in the context of the semantic web, all based on the XML technology. Examples of the widespread languages are OWL (Ontology Web Language) [Kalfoglou and Schorlemmer 2003], RDF (Resource Description Framework), RDFS (RDF-Schema), SKOS (Simple Knowledge Organization System), Topic Maps [Pepper 2010], DARPA Agent Markup Language [Cruz 2002] [Ding et al. 2001], and so on. [Kásler et al. 2006] classified the semantic-web standards into two main categories: the W3C standards such as OWL and RDF, and the ISO standards such as Topic Maps and Dublin Core. The resulting semantic models are what we usually call “ontology”.

- The concept of Ontology :

The formal definition of an ontology is “*an explicit specification of a conceptualization*” [Gruber 1993]. It represents a description of a domain of knowledge or discourse. Ontologies have been created to improve the communication between humans, between humans and computers and between computers by offering a unique and standardized vocabulary [Tixier 2001; Mellal 2007].

An ontology $\mathcal{O}$ is basically defined with 5-tuple which constitutes its entities:

$\mathcal{O} = < C, P, I, R, Cs >$ where C represents the classes, P the properties, I the individuals of classes, R the relations between the classes and Cs the constraints [Yang 2001; Noy 2004; Dou and McDermott 2006]. Each entity is defined as follows:

- ***Classes*** are concepts from specific-domain knowledge,

---

[5] http://en.wikipedia.org/wiki/Semantics
[6] http://www.w3.org/2001/sw/

- ***Properties*** are specific attributes related to classes,
- ***Individuals*** are instances of classes,
- ***Relations*** are ways in which classes and individuals can be related to one another,
- ***Constraints*** represent restrictions on the valid values of a certain property (e.g., the minimum and maximum of the value) [Li and Qiao 2012]. It can also include some axioms [Dou and McDermott 2006] or inference rules.

Ontologies are used in many disciplines of Information Science for diverse purposes. Most of scientific works [Tixier 2001; Noy 2004; Mellal 2007] agree that ontology usages focus on communication, interoperability between applications, description and specification of a business domain for knowledge reuse in system engineering. In fact, software systems must share same information or have same interpretation of information to interact properly. The notion of communication is also important between humans who want to communicate and share knowledge. Ontologies can take the form of referenciels facilitating the exchange of information. For this reason, they constitute nowadays the core of knowledge-based systems. With the advent of the Semantic Web, ontologies play an increasingly key role in managing semantics of information resources, on the web, as well as in company internal networks. In this case, they mainly serve to represent the meanings of information exchanged in enterprise-information systems [Tixier 2001].

The recognition of the importance of ontologies for the semantic web has led to the revolution and extension of the current web markup languages surveyed in [Ding et al. 2001]. Each of these languages focuses on some specific purposes of use, including interoperability, standardization, reasoning, semantic expressiveness, and so forth.

According to our study of semantic techniques used to improve the retrieval of information resources and components, we distinguished three standard technologies used for this purpose: *OWL*, *RDF* and *Topic Maps*.

- <u>The OWL language :</u>

OWL [Burgos 2011] is an ontology language for the Semantic Web, developed by the W3C. OWL was designed to represent information about categories of objects and how objects are interrelated. OWL is based on RDF syntax and relies on an XML specification. OWL describes the main components (classes, properties, relations, etc.) of an ontology with a set of annotations, axioms and facts where [Yang 2001]:

- Annotations represent meta-data about human and machine
- Axioms specify the characteristics of classes and properties
- Facts specify data about an individual or a pair of individual identifiers

In addition, the OWL language brings some operators for the comparison of classes and properties (equivalence, contrary, disjunction, etc.) [Mellal 2007], making it enough expressive for a description of a knowledge domain comparing to other semantic web languages [Bouzid et al. 2012a].

OWL is included in many ontology construction tools such as the famous Protégé editor, OntoEdit, DOE, WebODE, LinkFactory, etc.

- The RDF Syntax :

RDF is a standard model for data interchange on the Web. It handles the semantic description of web resources in order to improve their referencing and retrieval by users or applications. The syntax of RDF comprises a triplet of assertions (Figure 2.2-1): subject, predicate and object [Ding et al. 2001; Ellouze 2010]. The subject represents the resource to describe; the predicate is a type of property related to the resource and the object is the value of the property such as data or other resource.

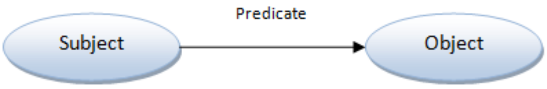


Figure 2.2-1 : RDF triplet

The general structure of RDF forms a structured graph with nodes and edges. The subject is the source node, the object is the targeted node and the predicate is the label of the edge. RDF allows in this way annotating web resources or any addressable subjects with URIs (Uniform Resource Identifier) using meta-data.

- The Topic-Maps standard:

Topic Maps are an ISO semantic web standard [Schneider and Synteta 2005] usually used to build semantic networks of data and concepts linked to heterogeneous resources. The key concepts of a Topic Map focus on topics, associations, occurrences and resources (Figure 2.2-2):

- a topic is a symbolic representation of a subject where a subject is a concept from a real world
- an association expresses a relationship between topics
- an occurrence is what links an information resource to a topic
- a resource is any technological support that handles information. It could be a document, a web page, a component, a database link, etc.

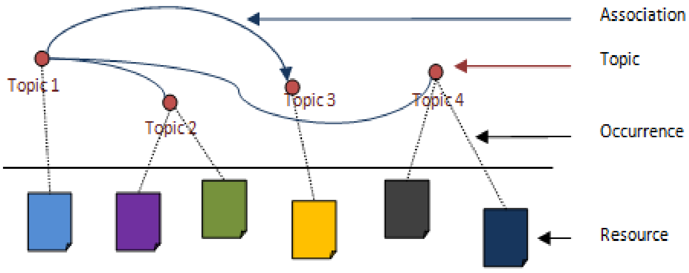


Figure 2.2-2 : Core concepts of the Topic-Maps standard

Topic Maps were designed to provide a suitable framework for resource navigation in order to solve the problem of large quantities of heterogeneous information contained in documents [Pepper 2010]. Topic Maps can be also used for designing ontologies because they capture subjects from the real world, giving as a result a description of a domain of knowledge [Bouzid et al. 2012a].

- Other semantic structures:

Other structures can be also used with semantic web standards to represent the semantics of information. Among the well-known structures used in computer science, we can cite the taxonomies, the thesauri, the lexicons and the dictionaries.

A **taxonomy** is a classification of concepts organized in hierarchy. It is usually used to organize the subjects of documents and web resources [Zacklad 2007].

A **thesaurus** is a list of organized and controlled terms of a knowledge domain. Each concept in a thesaurus is described with controlled terms (i.e. equivalent, generic, specific, associated, etc.) [Zacklad 2007]. There are many standard thesauri used in computer science, like Agrovoc (related to the agriculture), Delphes (related to the economical domain), MeSH (related to the biomedical domain), etc.

A **lexicon** is a stock of terms[7] and word forms used in a knowledge domain. The terms can be given with their equivalence in other languages.

A **dictionary** provides definitions of words with natural language and/or with controlled terms that express the meanings of the words (i.e. synonymy, polysemy, hypernymy, hyponymy, etc.). In the literature, the dictionaries also provide the etymologies, the phonetics and the pronunciation of words. One of the main well-known dictionaries used in computer science is WordNet[8] for the English language. A word in WorNet is generally defined with synsets that denote the standard senses or usages of the defined word. Each synset is a set of terms with similar meanings. An example of synset of the word *car*[9] is: *car, auto, automobile, machine, motorcar*. Several semantic relation types are used to organize the sense of the synsets. The main used are: synonyms, antonyms, hypernyms (i.e. general terms), hyponyms (i.e. terms included) and meronyms (descriptive parts of the defined word).

2.2.2. Engineering-Document and Resource Retrieval Approaches

Many research works tackle the use of semantic techniques to improve document sharing and retrieval. We precisely focus on the approaches dealing with industrial documents or information resources in enterprises. Accordingly, two categories of approaches are tackled here: engineering-document retrieval and information-resource retrieval in enterprises.

2.2.2.1. Engineering-Document Retrieval

Recent research works in document retrieval try to deal with the specific engineering resources which are widespread in manufacturing companies. Such resources usually result from advanced manufacturing systems (CADS, MES, CAPP, etc.) and constitute nowadays the heart of industries' information systems. Engineering documents in this case may include production data, engineering drawings, process sheets, control charts and so forth. Thus, the variety and complexity of the content of these documents makes their findability, shareability and understanding a little difficult for the engineers that use them. To address this difficulty in companies, first used techniques were annotations and meta-data in engineering documents [McMahon et al. 2004]. Afterwards, the popularity and efficiency of semantic web technologies have encouraged their use in industrial environments. Among

---

[7] http://www.thefreedictionary.com/lexicon

[8] http://wordnet.princeton.edu/wordnet/

[9] http://fr.wikipedia.org/wiki/WordNet

the approaches in this area, [Li et al. 2007] developed an engineering ontology and lexicon to tag engineering documents, so to improve information retrieval during the design process in manufacturing companies. [Yao et al. 2009] proposed an engineering ontology based on multi-source engineering-information and integrated it in an Engineering-Information Retrieval framework. [Li and Qiao 2012] proposed an ontology-modeling approach of manufacturing information and used it in a semantic retrieval framework. A detailed description of these approaches is in Appendix D.

- The Waypoint framework [McMahon et al. 2004]:

The Waypoint framework was initially developed by the former Chrysler Corporation and has been taken up by Airbus Deutschland. The authors developed an integrated retrieval system for engineering documents. This one provides a uniform access to heterogeneous information collections and multiple document sources stored on shared network drives. In the proposed framework, the engineering documents can be annotated using pre-identified concepts and retrieved using a faceted-classification mechanism [Mas and Zacklad 2008]. Moreover, the access mechanism allows both keyword searching and browsing of classification schemes of the document collections (hybrid-browsing technique).

An experimentation of the Waypoint framework was done in the Airbus UK site. The development of the faceted classifications was mostly based on the taxonomies of the directories that already existed within the company site, so to easily capture the concepts with which the engineers were familiar. The experimentation of the approach showed that engineers were able to more easily retrieve relevant documents using the Waypoint-hybrid-browsing approach, compared to the keyword search or to the conventional browsing of existing folder structures.

- The Engineering-Ontology (EO) search [Li et al. 2007]:

The authors proposed a semi-automatic ontology development methodology and integrated it in an engineering-information retrieval system, called EO-Search. The general approach was developed and experimented in the context of the automotive sector.

To implement the search system with the ontology-based approach, the authors proposed a methodology for the engineering-ontology development. The structure of the engineering ontology was based on a set of taxonomies of themes. An engineering lexicon was also used to store a list of words and derivations of each concept identified in the taxonomy. The ontology and lexicon were used to recognize concepts contained in the engineering documents. Each recognized word/sentence in a document is tagged by the corresponding concepts of the ontology.

The search interface of the EO-Search system is based on a standard keyword search and on navigation on concepts. The users' queries are processed with a concept disambiguation technique and a concept abstraction metric. The disambiguation technique consists in calculating correlations between all the keywords and the ontology concepts (including lexicon concepts) basing on their semantic closeness (a specific correlation formula is used). The abstraction metric exploits the structure and content of the engineering ontology in order to ferret out the true meaning (i.e. the target concept) of the query. The resulting list of documents are categorized and ranked according to the concept categories defined in the engineering ontology.

- Ontology-based EIR framework [Yao et al. 2009]:

The Engineering-Information-Retrieval (EIR) framework was developed in the context of an aerospace innovation project in China. The authors proposed a unified platform for search from multi-source documents (CAD drawings, CAPP[10] sheets, design manuals, PDF documents and some images and demo videos), to enable engineers to retrieve engineering information during the processes of product design, analysis and manufacturing. The framework comprises three main modules: ontology module, document analysis module and query processing module.

The ontology module is composed of an *engineering ontology* and an *application ontology*. The engineering ontology gives an abstract description (concepts and relations) of an engineering domain. The application ontology gives the description of the document resources basing on the concepts of the engineering ontology.

To obtain relevant information from the resources, the authors proposed a document-analysis module. It allows exploring information in the resources and supplying the application ontology in terms of concepts instances and relations.

The query-processing module processes the user queries, which are mainly composed of keywords. The system maps the user query with the concepts of the application ontology and the engineering ontology using multiple levels of semantic relations. An intention feedback mechanism is applied at the end of the query processing in order to make the query close to the user intention. The feedback is displayed to the end user in the form of a tree graph where the recognized query concepts, relations and semantic depth are proposed to the user for further navigation.

- Ontology-based modeling of manufacturing information and its semantic retrieval [Li and Qiao 2012]:

This approach seeks supporting the definition, the integration management and the retrieval of manufacturing datum in industries. A retrieval system was proposed basing on a manufacturing information ontology and a semantic similarity algorithm. The authors developed an UML conceptual model –called Manufacturing Information (MI) model– which describes and specifies information related to the manufacturing process. The main entities of this model arise from four manufacturing subjects: products, processes, resources and plants. The manufacturing information ontology was developed from this MI model.

The semantic similarity algorithm performs user queries using the concepts of the MI ontology. The proposed algorithm supposes that two concepts have certain similarity according to the type of relations and distance between them. A similarity distance between each two concepts is calculated accordingly in order to determine the closeness among the keywords of a user query and the concepts of the ontology.

2.2.2.2. Information-Resource Retrieval in Enterprises

Due to the growing quantity and heterogeneity of information resources in many companies, organizing their sharing and retrieval among users in the company network is a rising concern nowadays. Up to now, the widespread used technique is the use of commercial document-management systems to manage the indexation and search of

---

[10] Computer-Aided Process Planning

information resources. Such systems provide means to classify resources and annotate them, most of the time according to end-users' knowledge. The document-management systems are still used today because of their ease of use and maintenance. However, such systems often lack of semantic approaches to capture resources' description for indexation and search. Such lack becomes a real difficulty in industries where the resources have specific formats which cannot be systematically annotated and where the context of use of the resources is specific to business needs.

As a result, the semantic management of resources with ontologies has taken the lead on the standard document-management systems, especially because semantic technologies enable to analyze and interpret the user need in a precise way according to a business context [Zhang and Li 2008]. Among research works related to specific-business contexts, [Khelif et al. 2007] proposed in the biological domain an ontology-based approach (called MEAT) to create an annotation base from document sources and improve information search and resource retrieval using this base. Some approaches are based on the Topic-Maps technology to annotate resources and expose the subjects they contain. For example, in the software framework of [Kásler et al. 2006], the authors proposed a semi-automatic approach to capture the description of heterogeneous resources and structure it in a Topic-Maps model. In the HyperTopic approach [Zaher et al. 2006; Zaher et al. 2008], the authors — inspired by the Topic-Maps paradigm— used the concept of point of view and entity to co-build the description of information resources in a French company and facilitate the access to information they contain. More details about these approaches can be found in Appendix E.

On other hand, we noticed that Topic Maps are widely used in the context of e-learning to enable learners and instructors to share online courseware. Many research works are devoted to the e-learning field [Dichev et al. 2004; Ouziri 2006; Lavik and Nordeng 2004; Sridharan et al. 2009]. However, we consider that this field of study is out of scope of our work, so the e-learning approaches will not be included in this state of the art.

- The MEAT approach:

This approach [Khelif et al. 2007] was developed for biologists who work on DNA Microarray experiments, to support them in the validation and interpretation of their results. The authors proposed an approach and a system for the generation of ontology-based semantic annotations (called MeatAnnot), and a system for an advanced search by the biologists on the annotation base (called MeatSearch).

In the MeatAnnot system, the generation of ontology-based annotations on documents requires a lexicon of terms of the biological domain and an ontology describing this domain. The authors chose the semantic network of UMLS (Unified Medical Language System) [Hymphreys and Lindberg 1993] as upper-level ontology for the biomedical domain. The MeatAnnot system enables to generate a structured annotation based on the UMLS semantic network on one hand, and describes the semantic content of scientific documents (e.g. biological articles, scientific experiments, etc.) on other hand. The system processes texts and extracts interactions between genes and other UMLS concepts using the NLP technique with the Gate[11] tool. The MeatAnnot system generates an RDF annotation linked to each processed resource.

---

[11] http://gate.ac.uk/conferences/training-modules.html

The RDF annotation base supports user-queries' processing in the MeatSearch system. This one is based on the semantic search engine CORESE (Conceptual Resource Search Engine)[Corby et al. 2004] to enable users to query on the RDF annotation base. Furthermore, the search process is enhanced with an RDFS ontology which references the concepts of the UMLS.

- The Topic-Maps-based approach for resource navigation [Kásler et al. 2006]:

The approach of [Kásler et al. 2006] was created to improve the access to the NetWorkshop conference proceedings. The authors proposed a software framework to semi-automatically generate semantic representations of information present in a set of text files. The resulting semantic network is used to organize the access to the conference resources through a web portal. The Topic-Maps technology was used to handle the resulting network. This framework relies on four phases: data organization, analysis, Topic-Maps' population and content management.

In the first phase, meta-data were extracted from various information resources (e.g., MS office documents, PDF, etc.) and were stored in a structured way using the XML technology, so to have a uniform and formal structure. Afterward, a Topic-Map skeleton was built containing typed topics and topic keywords related to the extracted meta-data. Some statistical techniques were then applied to fill up the Topic-Maps skeleton with texts extracted from the resources. A Topic Map of information resources was obtained at the end and stored in a persistent database (XTM file or SQL database). The end-user can explore via a web portal the conference resources using the Topic-Maps resulting from the approach.

- The HyperTopic approach [Zaher et al. 2006; Zaher et al. 2007; Zaher et al. 2008]:

An HyperTopic [Zacklad et al. 2002] is a knowledge-representation model and language, created by Tech-CICO lab[12] for cooperative building of knowledge in the context of the Socio-Semantic Web [Guittard et al. 2005]. The HyperTopic model is considered as a specialization of the Topic-Maps model, created to guide users in structuring knowledge and retrieving resources in a same activity domain. This model is based on five main concepts to describe a knowledge domain: point of view, topic, entity, attribute and resource.

The HyperTopic approach was experimented in France Telecom and EADS companies. The main purpose was to share knowledge and resources about software projects. The Agorae tool [Zaher et al. 2006] was used to create different knowledge related to these projects and link them to resources (like documentations, project homepages, software links, etc.). The resulting HyperTopic knowledge map is what the authors call a "semiotic ontology". The end-users can consult the concepts of this map and navigate among several hundred of topics to explore resources.

2.2.3. Component Retrieval Approaches

We explore here the components engineering and service-engineering fields because their research issues also focus on sharing and retrieval purposes. In addition, because of the heterogeneity of software systems in industries, a manufacturing information resource can take the form of a software component.

---

[12] The Tech-CICO Lab is located in France, at the University of Troyes: http://techcico.utt.fr/fr/index.html

2.2.3.1. Software-Component Retrieval

In software engineering, the semantic description is used to enhance the sharing and reuse of software components among application developers. First used techniques were the facet-based classification [Lucrédito et al. 2005]. It consists in classifying components according to a set of subjects or characteristics of a given domain. Each characteristic is a facet described with some keywords. The chosen keywords can be defined from a controlled or uncontrolled vocabulary. In [Prieto-diaz 1991], the author proposed four general facets to classify software components: Domain, Functionality, Component Model, Component Type. The facet-based classification remains poor in terms of semantic expressiveness. It was used by many research works in the early 90’s. It is less used today, except for some works related to document classification [Mas and Marleau 2009].

With the advent of ontology paradigm, ontology-based approaches are ever more used in software engineering. Some approaches focus on the description of ontology models for software-component description [Li et al. 2008a; Peng and Zhao 2007; Graubmann and Roshchin 2006; Nianfang et al. 2010], and other research works focus more on the semantic-retrieval process of components [Yan 2010; Praphamontripong and Hu 2004; Shao and Zhang 2010; Li 2012]. We chose to focus here on some interesting works that provide a semantic approach for both component description and retrieval [Quan et al. 2007; Peng et al. 2009; Alnusair and Zhao 2010]. Some details about the approaches presented below can be found in Appendix F.

- The approach of [Quan et al. 2007]:

[Quan et al. 2007] developed an ontology scheme for software-component description with the OWL language. This approach was developed to support the component-based development method, a method that relies on assembling and composing already built software components. The authors proposed an ontology scheme to describe component information, so that it can provide semantic reasoning during the component retrieval process. They identified four facets of description: Component form, Application environment, Application functionality, Semantic information.

The architecture of the component retrieval system comprises three layers: a user-interface layer which displays the user interface for component search; a middle layer which allows parsing user queries and matching them with the domain ontology using semantic reasoning; and a resource layer which stores the components and the owl files related to user queries. In the proposed system, the user can specify its query in natural language. The system converts it into an owl representation and tries to match it with the domain ontology using the semantic associations of the ontology.

- The ontology-driven paradigm of [Peng et al. 2009]:

The authors proposed an ontology-driven paradigm for component representation and retrieval. They developed a component ontology composed of five facets: provider, environment, application domain, functions and interfaces.

The provider class records component providers’ name and points of contact.

The environment refers to the implementation information, i.e. hardware or software. Each environment type can import concepts from external ontologies (hardware ontology, software ontology).

The application domain describes the application scope of the component (i.e. the context of use).

The function class refers to the main function of the component.

Finally, the interface class describes the interface properties of a component, in particular, the input/output parameters.

To use the proposed component ontology in the retrieval system, the authors merged the component ontology and a domain ontology into a synthetic ontology for the implementation. They defined a retrieval algorithm based on syntactic matching and on the semantic association between the concepts of the resulting ontology and the developers' needs (related to any concept of the component description facets).

- The CompRE tool [Alnusair and Zhao 2010]:

The authors developed a component-search tool (called CompRE) as a plug-in for the Eclipse Environment. This tool is based on a knowledge base which gathers a source-code ontology, a component ontology and a domain-specific ontology.

The source-code ontology (called SCRO) captures the structure of an object-oriented library and helps understanding the relations and dependencies among source-code artifacts. *The component ontology (called COMPRE)* is an extension of the SCRO ontology. It gives additional component-specific descriptions. It also makes the link with the concepts of the domain ontology. *The domain ontology (called SWONTO)* conceptualizes the software libraries, by providing a common vocabulary with the unambiguous and conceptually sound terms that can be used to annotate software components.

The generated semantic instances of the proposed knowledge base were serialized using the RDF syntax, in order to allow SPARQL queries using the Jena framework[13]. The ontology-based-search mechanism in the CompRE tool is performed by semantically matching user queries with the component descriptions in the populated SCRO and SWONTO ontologies.

2.2.3.2. Service Retrieval

In service engineering - which is a recent research area of software engineering -, semantic services are used to add semantics to web services or to the software components that are captured in services. Ontologies are used in this field for many purposes, including service identification and creation [Delgado et al. 2010; Liu et al. 2005], service discovery and retrieval [Sycara et al. 2002; Gomez et al. 2006; Bernstein and Klein 2004; Mirbel and Crescenzo 2010], service composition and reuse [Budak Arpinar et al. 2004; Sirin et al. 2004], and so on. We are interested here in the service retrieval approaches dealing with semantic techniques. We identified three interesting ones: the LARKS framework [Sycara et al. 2002], the GODO approach [Gomez et al. 2006] and the SATIS approach [Mirbel and Crescenzo 2010]. More details about these approaches can be found in Appendix F.

- The LARKS framework [Sycara et al. 2002]:

The LARKS framework [Sycara et al. 2002] has been developed to provide a dynamic matchmaking among software agents for service search. The authors proposed a specification schema to define a service. The search request is based on the elements of this

---

[13] Jena is an Eclipse plug-in for managing RDF ontologies : http://jena.apache.org/

specification schema. Furthermore, the search framework uses three knowledge sources: an advertisement database for referencing the services, a domain ontology describing and referencing all the vocabulary used in the LARKS specification and an auxiliary database storing word distances and hierarchy types.

The search mechanism provides a combination of syntactic and semantic matching, according to a context of matching. It is based on five types of filters: context, profile, similarity, signature and constraint [Denayer 2004]. The composition of these filters allows establishing different degrees of correspondences.

- <u>The GODO approach [Gomez et al. 2006]:</u>

In [Gomez et al. 2006], the authors developed the GODO approach using a domain ontology and a goal-oriented approach. GODO is a goal-oriented-service discovery platform, developed to ensure the achievement of user intentions by means of semantic web services.

The platform is mainly composed of a goal-template repository, a Goal Loader and a Goal Matcher. The interface of the GODO system assists the end user in formulating his intention by proposing some recommendations to complete the statements expressing his intention. These recommendations are extracted from an external ontology repository. The semantic network stemming from the user query takes the form of a lightweight ontology. The latter is matched with goal templates of the goal-template repository, where different types of goals are stored. The Goal Loader of the GODO system retrieves the goal templates and transmits them to the Goal Matcher. This one compares the lightweight ontology to the description of the goal templates. From this matching, several goals are selected and composed in a way to build up the sequence of execution of the web services.

- <u>The SATIS approach [Mirbel and Crescenzo 2010]:</u>

The authors proposed the SATIS approach, a goal-oriented approach for web-service retrieval in a context of neuroscientist community. It mainly supports a neuroscientist seeking for web services to operationalize an image processing pipeline. The authors used three ontologies in this approach: a map ontology that captures user intention, an OWL-S ontology that describes web-services' functionalities and a domain ontology related to the application domain of the approach (neuroscientist in this case).

The search mechanism in the SATIS approach consists in operationalizing the image processing pipeline whose user intention was captured with the map ontology. The domain expert only selects the intention characterizing his/her image-processing pipeline and the system searches for the rules to use. The high-level intentional needs are created dynamically as needed during the backward-chaining process (such as temporary sub-goals) using the CORESE semantic engine[14], and this process is performed until the descriptions of the web services corresponding to the last sub-goals are found. Therefore, the captured user intention is considered satisfied. A set of web services' descriptions are given to the domain expert as a result.

2.2.4. Synthesis

We summarize the used approaches in information-resource retrieval and component retrieval with a set of criteria identified according to our problem statement and purpose of

---

[14] http ://www-sop.inria.fr/edelweiss/software/corese/

work. As reminder, the aim of this work is to enhance the sharing and retrieval of manufacturing information resources used to support the manufacturing process control. The main issue to address in this context is the lack of semantics. Accordingly, we identified two categories of criteria for this synthesis: the first focuses on the semantic description aspect and the second focuses on the resource retrieval aspect. These criteria are:

- Regarding semantic description:

1. ***Semantic structure***: related to the type of semantic structures used for resource retrieval, such as classification scheme, ontology, lexicon, etc.
2. ***Type of semantics***: type of knowledge described and handled by the semantic approach (e.g. domain concepts, resource meta-data, functionality, goals, etc.)
3. ***Semantic language/technology***: like semantic web standards (owl, rdf, etc.)
4. ***Semantic sources***: the knowledge sources from which the semantics was gathered
5. ***Semantic building-approach***: related to the used approach for building the semantic structure (e.g. an ontology-development method, ad-hoc approach, etc.)

- Regarding resource retrieval:

1. ***Type of resources***: related to resource types supported by the approach, such as documents, components, web resources, etc.
2. ***Resource-annotation approach***: the used technique to add semantics to the resources
3. ***Search inputs***: related to the input interface for search
4. ***Search outputs***: related to the output interface (how the results are displayed)
5. ***Search algorithm/mechanism***: used technique and mechanism to retrieve the resources with the semantics

The set of approaches of each field are synthesized according to the semantic description criteria in Table 2.2-1 and according to the resource retrieval criteria in Table 2.2-2.

Criteria **Approaches**	**Semantic structure**	**Type of semantics**	**Language / Technology**	**Semantic sources**	**Semantic-building approach**
Document retrieval approaches:					
The Waypoint approach [McMahon et al. 2004]	Faceted-classification	Domain concepts, resource meta-data (name, type)	XFML, relational database	Taxonomies and directories of the company	Manual ad-hoc approach (expert knowledge + company taxonomies)
The approach of EO search [Li et al. 2007]	Engineering ontology, lexicon	Domain concepts, lexical terms	XML	Engineering-knowledge resources (handbooks, textbooks, online catalogs, etc.)	Semi-automatic ontology-development approach in five standard steps
The EIR framework [Yao et al. 2009]	Engineering ontology, application ontology	Domain concepts	N/A	Engineering-knowledge base (lexicon, engineering standard, technology manuals, domain experts, online resources)	Manual ad-hoc approach
[Li and Qiao 2012]	Manufacturing-Information ontology	Domain concepts	OWL, XML	UML model of manufacturing information, expert knowledge, knowledge sources (handbooks, production experiences, etc.)	Manual approach: Top-down (expert knowledge) and bottom-up (knowledge sources)
The MEAT approach [Khelif et al. 2007]	Domain ontology, annotation base	Domain concepts, lexical terms	RDF, RDFS	Unified Medical-Language System (UMLS), lexicon of domain terms	Automatic approach: NLP technique to extract terms and relations, tokenization, concept mapping
The Topic-Maps-based framework of [Kásler et al. 2006]	Topic-Maps database	Resource subjects	XML, XTM, SQL database	FOLDOC source	Semi-automatic approach for Topic-Maps' construction
The HyperTopic approach [Zaher et al.	HyperTopic Knowledge base	Resource subjects, user point-of-view	XHT (i.e. XML for HyperTopic), MySQL	Experts' knowledge	Manual co-building approach

2006]			database		
Component/Service Retrieval approaches:					
[Quan et al. 2007]	Component ontology	Component form, environment, functionality, inputs, outputs, pre/post conditions	OWL	Component repository, expert knowledge	Manual ad-hoc approach
[Peng et al. 2009]	Component ontology, Domain ontology	Component provider, environment, application domain, functions, interfaces	OWL-DL	Hardware ontology, software ontology, domain ontology	Semi-automatic approach: import of concepts from the existing ontology sources
The CompRE tool [Alnusair and Zhao 2010]	Source-code ontology, component ontology, domain-specific ontology	Source-code, meta-data, software-library vocabulary	RDF	Software libraries, expert knowledge	Semi-automatic approach: source code capture and analysis, adding of meta-data, generation of semantic instances
The Larks framework [Sycara et al. 2002]	Advertisement database, domain ontology, auxiliary database	Domain context, variable type, inputs, outputs, constraints	LARKS (Language for Advertisement and Request for Knowledge), ITL (Information Terminological Language)	Expert knowledge	Description of services using the LARKS specification schema
The GODO approach [Gomez et al. 2006]	Domain ontology, goal-template repository	User intention, business goals	OWL-S, WSMO, METEOR-S	Expert knowledge	N/A
The SATIS approach [Mirbel and Crescenzo 2010]	Map ontology (of user intentions), web-service ontology, domain ontology	User intention, business goals, service functionality, domain concepts	OWL-S, RDF	Expert knowledge, user intentions, service repositories	Manual approach: use of the MAP formalism to capture user intentions

Table 2.2-1 : Approaches’ synthesis according to the semantic description criteria

Criteria / Approaches	Resource type	Resource-annotation approach	Search input	Search output	Search algorithm/ mechanism
Document retrieval approaches:					
The Waypoint approach [McMahon et al. 2004]	Text documents	Rule-based classification using the first sentence of the document, the document type and location	Key words, browsing of folder structures (with concepts selection), both (key words + browsing)	List of documents + short description (summary)	Syntactic matching
The approach of EO search [Li et al. 2007]	Engineering documents (engineering catalog, CAD drawings, technical reports, etc.)	Tagging of document concepts with the ontology concepts in xml files	Key words, concept navigation	List of documents categorized by ontology concepts	Concept disambiguation + concept-abstraction metric
The EIR framework [Yao et al. 2009]	Engineering documents (PDF, MS docs, CAD drawings, CAPP sheets, etc.)	Document-index creation by associating document resources with ontology concepts	Key words, phrases	List of documents, recognized concepts in a form of tree graph	Semantic matching of input concepts and ontology concepts
[Li and Qiao 2012]	CAD / CAPP resources, Manufacturing-Executing-System (MES) resources	N/A	Key words	Top-ten relevant resources	Syntactic matching + Similarity measure between types of relations
The MEAT approach [Khelif et al. 2007]	Biological documents	RDF annotation generation by mapping domain concepts with the concepts of the resources	Domain-concepts' selection	List of annotations with their document sources	Semantic search with the CORESE engine
The Topic-Maps-based framework of [Kásler et al. 2006]	Heterogeneous resources (folder taxonomies, MS docs, PDF, etc.)	Unsupervised classification of resource concepts + storage in a Topic Map	Navigation on topics and associations	Semantic network of resources	Navigation mechanism
The HyperTopic	Heterogeneous resources	Manual technique (by the	Navigation on topics	List of resources	Navigation mechanism

approach [Zaher et al. 2006]	(documentations, web pages, etc.)	user)	and associations	categorized by topics	
Component/Service Retrieval approaches:					
[Quan et al. 2007]	Software components	N/A	Key words	Set of components	Semantic matching between user query and the concepts of the domain ontology
[Peng et al. 2009]	Software components	N/A	Key words	Set of components	Semantic matching between user query and the concepts of the ontologies
The CompRE tool [Alnusair and Zhao 2010]	Software components	Tagging of components with the concepts of the domain-specific ontology	Concepts of the domain ontology, SPARQL queries	Set of ranked components	Matching of SPARQL queries with component annotations
The Larks framework [Sycara et al. 2002]	Web services	N/A	Key words, phrases	A set of services	Semantic matching of concepts according to the context of services
The GODO approach [Gomez et al. 2006]	Web services	N/A	Free text, goal selection	Set of web services	Input transformation into light-weight ontology + matching with goal templates
The SATIS approach [Mirbel and Crescenzo 2010]	Web services	N/A	Goals	Set of web services	Semantic matching with a backward-chaining technique using the CORESE engine

Table 2.2-2 : Approaches’ synthesis according to the resource retrieval criteria

According to these two syntheses, the common point that emerges from the approaches is that they deal with ontology structures using the semantic-web standards, in particular OWL and RDF, allowing hence, a semantic matching of resources regardless of their types. However, the semantic aspect is not treated in the same way in document retrieval as in component retrieval —mainly because the context is different—, whereas the retrieval mechanisms are approximately similar.

- Regarding the semantic description aspect:

We can notice that in all the approaches the authors tried to identify the types of ontologies to use for the retrieval purpose and according to the business context. The main technique used to build such ontologies is manual and ad hoc. Some of these approaches proposed automatic and semi-automatic techniques, but such means can only be applied to textual sources where concepts and relations can be extracted and deduced with NLP techniques.

On other hand, the document-retrieval approaches focus on how to capture and build the semantics from knowledge sources, while component/service approaches focus more on identifying and categorizing the types of concepts that will serve to structure the semantics. Indeed, the retrieval mechanisms in document approaches usually rely on business vocabulary —often because the documents contain it—, whereas in component/service approaches, the retrieval rely on software meta-data like the functionality of the component, its inputs and outputs, the implementation environment, etc. Furthermore, because such components are built to be reused in software development, the general structure of component ontologies always focuses on source-code and technical environment, on the form of components and on the functionalities and services they provide. This description rarely focuses on business concepts because the components and services are built in a way to be reused in any business context. In document retrieval, the capture and construction of the semantics in the ontologies highly depends on the business context.

Some of the recent approaches in service engineering tried to deal with business contexts, such as in the SATIS approach [Mirbel and Crescenzo 2010] and the GODO approach [Gomez et al. 2006]. The authors proposed a kind of goal ontologies and proposed to use the user intention as a key input in the retrieval mechanism. In the HyperTopic approach [Zaher et al. 2006], the authors proposed the concept of “point of view” to describe the resources from a user standpoint. We can note that these approaches try to handle users’ needs by means of business goals and points of view comparing to the other approaches which do not emphasize the end-users’ needs.

- Regarding the resource retrieval aspect:

The main technique used for resource retrieval in all the approaches is the semantic matching, which also includes a syntactic matching. The document-oriented approaches proposed to annotate the resources and create document indexes. These annotations and indexes seek to speed up the retrieval of documents during the user-query processing. These techniques are not applied in the component/service-oriented approaches except in the CompRE approach [Alnusair and Zhao 2010] where the authors tried to tag the components with meta-data in a way to be indexed.

Otherwise, the search inputs are mainly based on keywords and business concepts in most of the approaches. Some engineering-document-oriented approaches proposed navigation between concepts, either in the search input or in the output, in order to guide the user in discovering and retrieving the documents. However, the user need remains limited to subjects found in the documents comparing to the SATIS [Mirbel and Crescenzo 2010] and GODO [Gomez et al. 2006] approaches which proposed to express and capture user intentions with business goals.

Finally, in almost all the approaches, the search process is based on a similarity measure between the concepts of user queries and ontology concepts and relations. The output of the retrieval process in component/service-oriented approaches only gives a list of resources. Such results are not accurate, mainly because the user need is not enough precise. In the Waypoint approach, [McMahon et al. 2004], the authors asserted that the use of a resource-browsing mechanism with the keyword search is the most intuitive and effective technique for search. We can notice that visualization techniques are widespread in document-retrieval approaches compared to component-oriented approaches.

According to this synthesis and brief comparison, we identified three key points to focus on in order to set up a semantic approach for resource retrieval:

(i) the type of ontologies or semantic structures to use must well suit the business context

(ii) the semantics must be well identified and described in the semantic structures because the retrieval process mainly relies on their consistency

(iii) the types of concepts to use for capturing users’ needs must be business-oriented in particular when the retrieval purpose is business-context dependent

Indeed, the semantics plays here a major role in matching user needs with heterogeneous resources. It is necessary that the semantic solution helps in bridging the gap between the users’ needs and the resources that meet them. As a result, the alignment-oriented approaches would be worth exploring in order to see how alignment objectives are carried out at a conceptual level.

2.3. Alignment-Oriented Approaches

Alignment issues interest nowadays many disciplines of information sciences. The main purpose of the alignment is to make a consistent link between at least two entities [Regev and Wegmann 2004]. These entities may relate to corporate strategies, business processes, software architectures, codes, and so forth. The strategic alignment model (SAM)[Henderson and Venkatraman 1999] distinguishes internal alignment from the external alignment. The external alignment focuses on the link between business strategy and the strategy of the Information Technology (IT), while the internal alignment focuses on the relationship between business processes and the technological infrastructure that supports them. Accordingly, the main fields where the alignment methods are generally used are: Requirement Engineering, Enterprise Architecture and Service-Oriented Architectures.

Alignment-based techniques are also used in knowledge engineering, in particular to align ontologies and semantic structures. This type of alignment is related to terminological and linguistic aspects and relies on syntactic matching and semantic mapping techniques. This type of alignment is not presented here because it is out of scope of our work. This section only aims at exploring the conceptual architectures and frameworks used to make software resources aligned with business needs.

2.3.1. Alignment in Requirement Engineering

Main alignment methods are created in the context of Requirement Engineering for the construction of software architectures aligned with the business [Bleistein et al. 2004a; Regev and Wegmann 2004; Bleistein et al. 2005]. Some of alignment approaches focus on transforming an existing architecture, i.e. an “AS-IS” situation, to a “TO-BE” situation [Etien 2006]. [Rolland 2003] proposed a Map formalism to describe this transition. This formalism was also used in the INSTAL method [Thevenet 2009], an alignment method that deals with the enterprise strategy and its information system. [Bleistein et al. 2004a] proposed the B-SCP approach, a strategic alignment method that uses a goal-refinement mechanism. [Veres et al. 2009] extended the B-SCP approach with an ontology that captures the concepts modeled with the BMM model. Some other approaches focus on the alignment between software architectures and the business processes of the companies [Wieringa et al. 2003].

In **[Bleistein et al. 2005]**, the authors proposed a conceptual framework for modeling the business/IT alignment by describing the strategic goals of the company and its activities and processes. This framework relies on the “problem-frames” mechanism to capture goals and refine them from a strategic level to an operational level. The "Problem Frames" are a mechanism and notation based on a requirement graph which describes the properties of a software problem by representing the existing context and how the stakeholders would like the system to be. The goals of the strategic level are refined by the concept of progression between problems. In this alignment approach, the goals can be formulated at different levels of abstraction because they are usually too abstract to design a system solution [Bleistein et al. 2004b].

In **[Thevenet 2009]**, the author proposed the **INSTALL (INtentional STrategic ALignment) method** to model the alignment between the strategic level and the operational level in companies. The core content of this method relies on a “pivot model”, modeled with the Map formalism [Rolland 2003]. In order to have a pivot model in which the elements of the two levels to align are subsumed, the concept of “intention” is used. In fact, strategic concepts (like objectives, business plans, etc.) and operational concepts (like applications, business processes, etc.) have different levels of abstraction. The intentional-pivot model aims at addressing these problems of conceptual discrepancy and offset of levels of abstraction. Also, the approach considers that there must be a common intention shared between the elements to align, otherwise there is no alignment.

In, **[Wieringa et al. 2003]** the authors proposed a framework to analyze and design the alignment between application architectures and the business context. The authors consider that the alignment problem stems from the connection of three worlds: the *social world*, the *linguistic world* and the *physical world*. The authors proposed then a framework to construct an information system where these three different worlds and their components are aligned. This framework gives descriptions of various elements related to each world, traditionally used to model business processes or to develop information systems.

2.3.2. Alignment with Enterprise Architecture (EA)

Enterprise architects seek aligning enterprise processes with IT systems [Wegmann et al. 2005a]. We can identify three interesting approaches in this area: the SEAM approach [Wegmann et al. 2007], the Zachman framework [Zachman 2003] and the ARIS architecture [Ferdian 2001]. The SEAM approach [Wegmann et al. 2007] aims at building a “TO-BE” system where the market, the enterprise and its software systems are aligned.

[Zachman 2003] proposed a framework of Enterprise Architecture which aims at developing a holistic documentary vision of the enterprise using three main models: an enterprise model, a system model and a technological model. These models enable to organize the business alignment of the company with Information Technologies and support their integration according to six different perspectives. Each perspective underlies a question: Why (the motivation description), How (the function description), What (the data description), Who (the people description), Where (the network description) and When (the time description). Each perspective is used to describe an entity of the enterprise. The answers to these questions must be done according to five facets: contextual, conceptual, logical, physical and detailed description. This framework has become a reference for many EA frameworks (DoDAF[15], FEA[16], etc.) and approaches dealing with alignment in software engineering [Simonin et al. 2010].

The **ARIS framework [Ferdian 2001]** is one other popular EA architecture. ARIS is a framework for modeling business processes, usually used for industrial environments. ARIS relies on dividing business processes into separate views and integrating these views to form a complete view of the whole business process. These views are:

- ***Data view***: related to business entities or objects related to a given activity (for example an equipment is an entity of the manufacturing process activity)
- ***Function view***: related to the business activities, functions and goals of the company
- ***Organization view***: related to the organizational structure of the company (business areas, services, business actors, etc.)
- **Resource view** related to the IT components and software implementation
- ***Control view***: the processes that make the link between all ARIS views

In **[Wegmann et al. 2005a]**, the authors proposed the **Seam approach** which provides a systemic paradigm for analyzing enterprises and their IT systems and proposes, accordingly, a modeling method to align them, so to build a “TO-BE” situation. The alignment approach in SEAM is handled at three levels [Wegmann et al. 2005b]: between organizational levels, between functional levels and in the global Business/IT alignment. The organizational level represents a partial enterprise reality and each organizational level contains systems. The typical organizational levels include: *a business leve*l (the enterprise), *an operation level* (organizational units such as sections and divisions) and *an IT level* (IT platforms and systems). The functional level represents the behavior of the system to describe at any organizational level.

---

[15] http://en.wikipedia.org/wiki/Department_of_Defense_Architecture_Framework

[16] http://en.wikipedia.org/wiki/Federal_enterprise_architecture

2.3.3. Alignment with Service-Oriented Architectures (SOA)

The principle of a Service-Oriented Architecture (SOA) is to organize the information system into a set of services exposing business and software functionalities. Several works emphasize the importance of service identification in SOA, as it is a milestone in the business/IT alignment [Arsanjani 2005; Tsai et al. 2006; Bianchini et al.; Delgado et al. 2010]. In the recent works, the SOA approach is treated on the whole with the alignment issues at a conceptual level, giving rise to a new vision of the service-oriented architectures such as BITAM-SOA [Chen 2008] and SOAGM [Haki and Forte 2010].

BITAM-SOA [Chen 2008] is a three-layer architecture model based on the SOA paradigm. These layers are the *business model*, the *business architecture* and the *IT architecture*. They indicate the traditional separation of concerns of an Information System (strategic, operational and infrastructure) and stipulate the areas for abstraction and encapsulation in SOA architectures.

In **[Haki and Forte 2010]**, the authors proposed to move from a “AS-IS” situation to a “TO-BE” situation using an **SOA Governance Model (SOAGM)**. The migration approach is based on a set of steps (planning, analysis, organization and migration). The SOAGM model that governs this migration features three architectural layers: *the business-service layer*, *the information*-systems’ layer and *the infrastructure layer*. The business service layer decomposes business processes in order to determine business services. The information-systems layer supports the combination of services for determining application services. Finally, the infrastructure layer handles the environment in which application services (i.e. service directory, service provider and service requester) support the business services.

2.3.4. Synthesis

Table 2.3-1 presents a synthesis of the presented alignment approaches using four criteria:

- **Purpose of the approach**: is related to the objective addressed by the alignment approach regarding the Information-System research-areas
- **Type of levels**: refers to the type of levels identified in each approach to attain alignment objectives through conceptual architectures
- **Knowledge types**: focus on the type of knowledge handled by the approach for the modeling of the Business/IT alignment
- **Conceptual technique**: is related to the modeling techniques or formalism used to build the knowledge to align

Some of the presented alignment-oriented approaches are more described in Appendix G.

Criteria **Approaches**	**Purpose**	**Type of levels**	**Knowledge types**	**Conceptual technique**
The Goal-oriented framework [Bleistein et al.	Strategic-alignment modeling	Strategic, operational	Strategic requirement, goal, context	Goal refinement with the problem frames, strategy

2005]				modeling with the BMM model
The INSTAL method [Thevenet 2009]	Strategic-alignment modeling and evaluation	N/A	Concept of “Intention” between strategic concepts and operational concepts	Goal modeling with the MAP formalism, intention capture in a pivot model
The alignment framework of [Wieringa et al. 2003]	IT-system specification	Social, linguistic, physical	Social concepts (processes, business context), linguistic concepts implementation environment, application system), physical concepts (hardware network)	Concept abstraction from low level to high level
The SEAM approach [Wegmann et al. 2005a]	Enterprise description, IT-system specification	Business operational, IT-system	Business concepts, operational concepts, IT-system concepts	Concepts capture and modeling using activity diagrams
The Zachman framework [Zachman 2003]	Enterprise description and modeling	Contextual, conceptual, logical, physical, detailed	Motivation, function, data, people, network, time	Use of various specification models (list, process models, diagrams, rule specification, etc.)
The ARIS Architecture [Ferdian 2001]	Enterprise description and modeling	Process engineering, process planning and control, workflow control, application systems	Data, function, organization, resource, control	concepts’ modeling with activity diagrams, entity-relationship models, and process modeling with the EPC notation [Mendling and Nüttgens 2005]
The BITAM-SOA [Chen 2008]	Service-Oriented-Architecture construction	Strategic, operational infrastructure	Business-model concepts (strategic), business-architecture concepts (operational), IT-architecture concepts (infrastructure)	Service identification and modeling for each layer
The SOAGM [Haki and Forte 2010]	Service-Oriented-Architecture	Business-service, information	business-service layer, information-systems layer and	Service modeling approach

	construction	systems, infrastructure	infrastructure layer	

Table 2.3-1 : Alignment-approaches' synthesis

According to this synthesis we can classify the concepts used in alignment approaches on three categories: business, operational and IT system. The business type refers to the business-domain description, in particular strategic and business goals. The operational type refers to the description of the business process of the company whose activities, functions and data flow need to be handled by the IT system. The IT system encompasses the IT infrastructure and software implementations. Some approaches proposed concept classification for the enterprise description. The Zachman framework draw up a holistic description of the enterprise with the technique of five Whys and one H (Who, what, when, where, why, How), a technique known in problem-solving methods. In the ARIS approach, the authors proposed five types of concepts: organization, function, data, resource and control. These concepts refer to the main entities involved in an enterprise process. Otherwise, the main concept used in the Requirement-Engineering approaches is the goal/intention concept. Such approaches are specialized in capturing business needs and the concept of goal with the goal decomposition mechanism is a fundamental notion and technique used in these approaches to capture business needs and refine them.

To sum up, we can say that the three categories of concepts "business, operational and IT system" are quite appropriate for enterprise description, whereas the "goal" concept is more appropriate to describe business needs or requirements that need to be satisfied with software resources.

2.4. Conclusion

The state of the art in this chapter introduces some notions and approaches related to the problem statement of this thesis. This state of the art has an original aspect: it includes two fields of study that can intersect.

The first field introduces a set of approaches and techniques dealing with resource retrieval for the sharing and reuse of information in specific business contexts. These approaches have proposed semantic techniques to enhance the retrieval of resources, reconciling in this way the different terminologies of a domain, the users' needs and the resources' description. While the basis of the semantic approaches focus on ontologies, their usage in a retrieval process differs following the business context and the purpose of information retrieval. In document retrieval, ontologies are used to gather the concepts and lexical terms of a domain and improve the perception of their meaning. In component/service retrieval approaches, ontologies are used to semantify software components and contextualize their use. In both cases, the challenge in using ontologies revolves around the type of semantic content (i.e. business, software, linguistic, etc.), because it leads the retrieval process regardless of the used mechanism and search algorithm.

In the second field of study, the alignment-oriented approaches use the concept of goal with a refinement mechanism to meet business needs at any level of abstraction. This technique is used in some recent component/service retrieval approaches only. Yet, resource retrieval approaches try to address business needs; they must be able to capture these needs with appropriate techniques. Up to now, business needs are not well addressed in most existing approaches even when semantic techniques are used. This ascertainment leads us to well focus on this issue while outlining a semantic approach dealing with the context of our work.

Finally, by studying the two fields, we can notice that, apart the engineering-document approaches which are specific to industries, the component/service retrieval field addresses the closest scientific issues to our purpose. As mentioned in the research issues (Chapter 1), the solution to the retrieval problem must also consider the alignment aspect. However, the used approaches in component/service retrieval cannot be applicable as such, because the used resources in the process control context have their own constraints (including the heterogeneity of the resources, their proprietary format, the non-expert end-users, etc.) and are intended for business usages comparing to the software components and services which are intended for software development purposes. Also, because the process control methods and their implementation may differ from a company to another, bringing a generic solution in this context is certainly another constraint to consider and that can be a real challenge. Ultimately, these conclusions lead us to propose our own approach, called S3, aiming at supporting the description and retrieval of the MI resources used for the manufacturing process control in industries. The STMicroelectronics’ case study helped in establishing the basis of the proposed approach.

Chapter 3: Approach Overview

Résumé

S3 est une approche sémantique qui permet à la fois la description et la recherche de ressources d'information manufacturière. Cette approche repose sur deux stratégies de recherche complémentaires : une stratégie ascendante (« bottom-up ») et une stratégie descendante (« top-down »). L'approche ascendante permet de construire des descripteurs sémantiques de ressources existantes dans l'entreprise grâce à une technique de mapping de concepts créée à cet effet. L'approche descendante permet la capture des besoins métier et l'alignement des ressources avec ces besoins en utilisant des patterns de recherche et en ré-exploitant les descripteurs sémantiques. Les mécanismes de description et de recherche sont supportés par deux structures sémantiques : une ontologie « manufacturing process » et un dictionnaire « process control ». L'approche S3 vise ainsi à améliorer la recherche d'information manufacturière en industrie en construisant une description sémantique des ressources utilisées, et en mettant les besoins métier des acteurs qui assurent le contrôle des processus industriels au cœur des mécanismes de recherche.

Ce chapitre donne un aperçu général de l'approche S3 et des résultats obtenus à l'issue de ce travail de recherche. Trois piliers de l'approche sont introduits, à savoir, le support sémantique métier constitué de l'ontologie et du dictionnaire, les descripteurs de ressources et les patterns de recherche. Un prototype d'implantation de l'approche basé sur le modèle « Topic Maps » est également introduit. Chacune de ces contributions est détaillée dans les chapitres suivants.

3.1. Purpose

Because the information systems in industries are complex and the business needs are evolving, bringing a reliable support to resource retrieval and information sharing becomes then momentous to ensure an efficient and continuous control of the manufacturing processes. While most industrial companies use commercial software platforms and manufacturing systems to control their processes, they are not quite aware of the consequences behind the lack of an effective process-control support. As noticed in the context of work, the lack of resource semantics is one of the main aspects to tackle in order to make the MI resources findable and understandable by the end users. Besides the lack of semantics, other difficulties were observed and must be considered such as the vocabulary heterogeneity and the diversity and scalability of business needs.

Moreover, the basic problem facing any user searching for a resource is how to well capture his need and how to retrieve the resources relevant to his business objectives. The semantic solution that will be appropriate in this case, must consider not only the business semantics related to the resources but also the semantic alignment of knowledge during the process of MI-resource retrieval. To do so, we propose the S3 approach that:

a. overcomes the lack of semantics in the resources by bringing them the required semantics using an ontology and a dictionary
b. enhances the retrieval difficulties by making the resources close to the business needs during their retrieval basing on a pattern-based search

3.2. The S3 Approach

We propose a semantic approach that we call S3, to improve resource description and retrieval in the context of the manufacturing process control. The S3 approach relies on two retrieval strategies which use a ***Semantic support***, ***Semantic descriptors*** and ***Search patterns*** as follows:

- The first strategy uses a mapping mechanism to explore the existing MI resources in the company and associates by this mean a first level of semantics to the resources. The resulting semantics is captured in ***semantic descriptors***
- The second strategy provides a business-need-focused search that combines a keyword search with the user-query capture in ***search patterns***. This pattern-based search enables to associate a second level of semantics to the resources to achieve their alignment with the business

The retrieval strategies are supported by two semantic structures [Bouzid et al. 2013a]: a ***Manufacturing Process (MP) ontology*** and a ***Process Control dictionary (PC)***. The usability of these structures is twofold: they streamline the vocabulary related to the process control activity in a process control dictionary, and they gather the semantics related to the core business by means of a manufacturing process ontology.

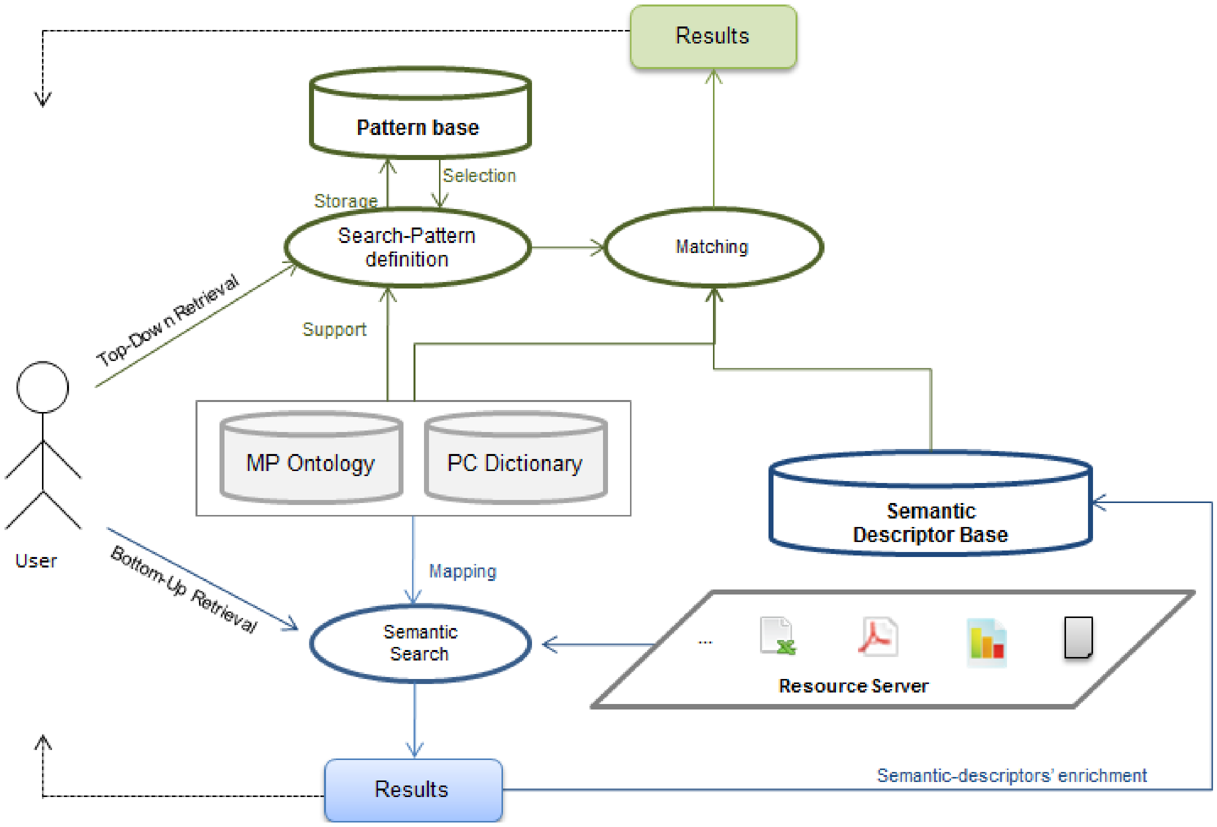


Figure 3.2-1 : Overview of the S3 approach

3.2.1. The Semantic Support

Most of the research works that tackle the semantic retrieval of resources emphasize the use of one or several business ontologies. Such ontologies provide the description of a domain as a whole, without differentiating between the types of used semantics and the linguistic aspect. We propose in our approach to separate the business description of a domain from the rationalization of its vocabulary in two semantic structures: a business ontology and a domain-specific dictionary.

- The manufacturing process ontology

The manufacturing process ontology provides the description of the manufacturing process of the company. It precisely captures the business concepts related to the business needs of the company. The captured concepts are structured in four views of description semantically linked through the ontology relations. The proposed views are inspired from the ARIS architecture [Scheer and Nüttgens 2000]. Indeed, the most difficult task in ontology development is to determine the types of concepts to capture and how to target them since the beginning of the ontology development process. We use the ARIS views because they enable to separate the description of a complex process into four coherent views, which can be reassembled at the end to form a complete descriptive view of the process. The used views in our ontology modeling are: the *organization view*, the *function view*, the *data view* and the *control view*.

The concepts of the MP ontology were captured in a first step with the conceptual design of each view using requirement engineering techniques, in particular the UML diagrams and the goal decomposition mechanisms [Ali et al. 2010]. In a second step, the captured concepts were filtered to only keep the relevant ones to the scope of the ontology.

Finally, we can note the particularity of this business ontology which handles two knowledge domains that address distinct business needs: those related to the manufacturing activity of the company and those related to the process control activity. These knowledge domains always intersect, because the process control objectives must address the manufacturing needs.

- The process control dictionary

We propose a simple dictionary structure for semantic definition of concepts related to an industrial activity domain. In the S3 approach, this structure gathers the vocabulary of concepts related to the process control, in particular the concepts related to manufacturing-data processing, as provided by the MI resources. The dictionary is precisely used in the approach to make the link between the raw and heterogeneous vocabulary related to the control domain and the standard process control description that is referenced in the ontology. Finally, the process control dictionary is linked to the manufacturing process description of the ontology through the process control view.

- Role of the semantic support

The proposed semantic structures are used to support both resource description and search, using *semantic descriptors* of resources and *search patterns*. In fact, because of the lack of semantics in the resources, semantic descriptors are used to associate meta-data to the resources, in particular their basic business usage. The search patterns capture the users' needs, in order to facilitate the mapping between the user queries (either simple or complex) and the resource descriptors. Thus, we can summarize the usages of the manufacturing process ontology and the process control dictionary in the following (Figure 3.2-2):

(1) they provide the required semantics for the definition and enrichment of the semantic descriptors and the search patterns

(2) they support the semantic retrieval of resources according to the two retrieval strategies

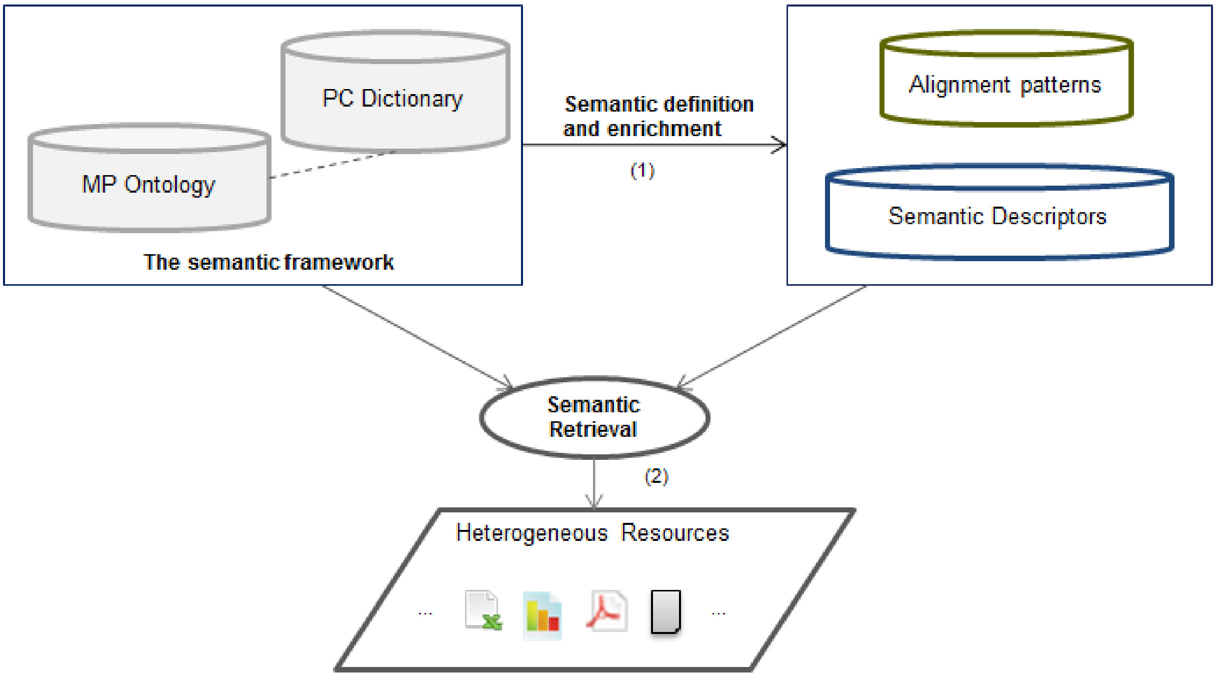


Figure 3.2-2 : Role of the semantic structures

3.2.2. The Semantic Descriptors of Resources

A semantic descriptor is associated to each MI resource in a way to semantify it. Indeed, a semantic descriptor captures the basic business semantics of each resource. It is precisely used to enhance the matching between the user queries and the business semantics of the resources. The targeted type of semantics that constitute the semantic descriptors is captured from the business ontology.

A semantic mapping technique [Bouzid et al. 2013b] is applied to achieve the construction of the semantic descriptors in the bottom-up retrieval strategy (Figure 3.2-3). This technique relies on mapping the basic business concepts that we can find with the MI resources (mainly uncontrolled vocabulary provided by the business experts) with the concepts of the dictionary and the ontology. The mapping is mainly based on string similarity measuring and reasoning on the relations between the ontology concepts. Basically, the mapping process is performed in three steps. At the beginning, the captured concepts from the resources are filtered and tokenized in order to isolate relevant concepts to the business. These concepts are mapped in a second step with the concepts of the dictionary entries (i.e. variants, synonyms, key concepts, etc.) using a string similarity algorithm that combines syntactic and statistical techniques. Two mapping techniques are used in the combined similarity formula that we call ***CSim***. These techniques are the **Sørensen-Dice coefficient** and the **Levenshtein distance**. The first technique calculates the similarity between two sets, whereas the second technique calculates the similarity between two strings. In this way, by combining the similarity measures between sets and between the concepts that constitute these sets we obtain a balanced similarity ratio over each two sets.

Finally, this mapping technique enables to determine the appropriate entry of the process control dictionary which better corresponds to the description of the MI resources. The selected entry in the process control dictionary indicates the control description associated to the resources. This control description enables to make the link with the manufacturing process ontology. All the concepts of the semantic descriptors are then identified using the relations between the ontology concepts and basing on some inference rules.

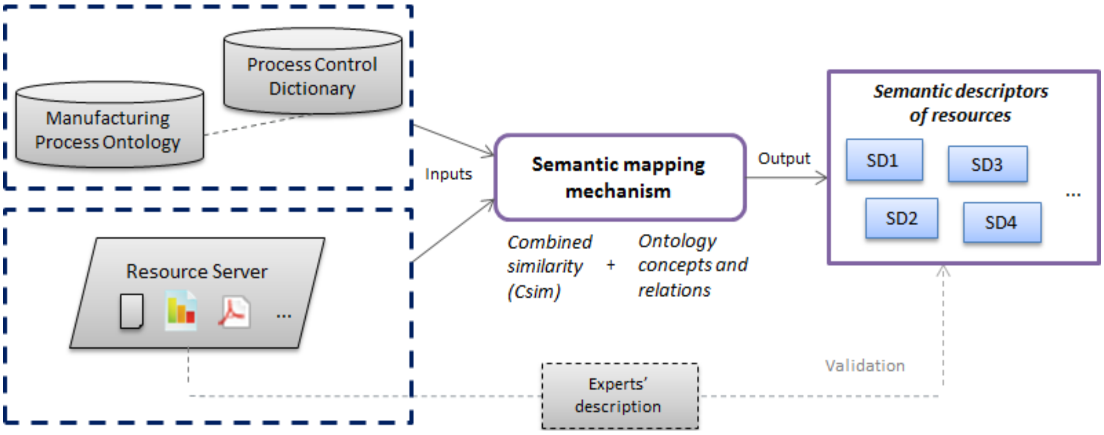

Process Control Dictionary
Manufacturing Process Ontology
Resource Server
...
Inputs
Semantic mapping mechanism
Combined similarity (Csim)
+
Ontology concepts and relations
Output
Semantic descriptors of resources
SD1
SD3
SD2
SD4
...
Validation
Experts' description

Figure 3.2-3 : overview of the mapping technique

A prototype for the semantic mapping of resource was implemented for experimentation [Bouzid et al. 2013c]. Expert descriptions of a set of selected MI resources were prepared in the STMicroelectronics Company, to validate step by step the findings of the bottom-up approach for the construction of the semantic descriptors. This semantic mapping technique also enabled to correct and stabilize the content of the semantic support for its effective exploitation.

3.2.3. The Search Patterns

The search patterns provide a top-down strategy of MI-resource retrieval and enrich their description with high-level business needs. In the proposed approach, the pattern-based search combines a keyword search with goal-oriented mechanisms carried by alignment patterns. In fact, two types of search patterns are used: query patterns and alignment patterns. The query patterns assist the capture of the business needs – either simple or complex – of the end users while the alignment patterns support their satisfaction using goal decompositions. The whole pattern-based search is supported by the semantics of the manufacturing process ontology and the process control dictionary.

The interesting asset of the pattern-based search is that business need artifacts are progressively captured in alignment patterns and stored for further reuse. The creation of the alignment patterns is supported by three types of goal-oriented mechanisms: goal decomposition, goal-sibling decomposition and goal abstraction. The goal-decomposition mechanisms are handled with alternatives of decompositions that depend on business contexts.

Figure 3.2-4 depicts the role of the pattern-based search in the retrieval system of the S3 approach. The end users do not have a direct access to the MI resources, they use instead alignment patterns by creating and/or reusing existing ones.

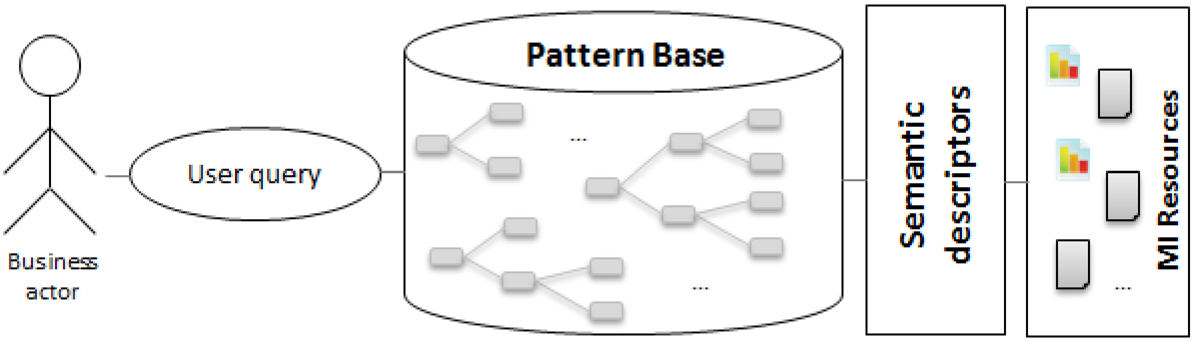


Figure 3.2-4 : Role of the patterns in filling the gap between a user need and MI resources

In this way, the alignment patterns carry the expression of business needs up to their satisfaction in a given context. They provide another level of description that achieves the alignment between the users' needs and the resources. The alignment patterns are also linked to the semantic descriptors of resources through the business goals and contexts, enabling then to identify the resources corresponding to each given business need.

3.3. The Topic-Maps-Based Application

We propose as implementation of a part of the approach a Topic-Maps-based application for resource retrieval. Each set of mapped resources is progressively referenced in a knowledge base in order to be efficiently retrieved by the end users. This knowledge base is based on the Topic-Maps' model.

Three search functions are proposed in the application:

- Resource mapping and referencing: implements the automatic mapping technique for the construction of the semantic descriptors. This function also enables to reference the resources (i.e. their location) with their semantics in the knowledge base
- Pattern-based search: enables resource retrieval using the search-patterns' technique
- TM-based search: helps the end users understanding the business semantics related to the resources that are referenced in the knowledge base. This function takes as input a set of keywords and builds accordingly a specific Topic Map with all the concepts and resources that respond to each user query [Bouzid et al. 2012b].

The proposed prototype has a particular originality. It uses the Topic Maps' paradigm for resource referencing and search in the knowledge base. The Topic-Maps' paradigm allows handling any type of knowledge regardless of the context of use. This generic aspect enables to easily manipulate the semantics in the knowledge base model without real constraints, and to display it in a flexible way to the end user. Furthermore, the maintenance of the knowledge base is ensured over time without any change in the conceptual model.

3.4. Approach Findings

On the whole, five main results were produced through this research work:

- A manufacturing process ontology that captures the description of the manufacturing process. The specificity of this ontology is that it structures business concepts in four views of description, where the link between the views semantically converges to business needs in an activity domain

- A process control dictionary: this one rationalizes the used concepts and terminologies related to the process control of any manufacturing activity
- An automatic mapping technique to construct semantic descriptors of resources using the manufacturing process ontology and the process control dictionary
- A meta-model of search patterns: these patterns provide a business-need-focused search that realizes, by the way, the alignment between the needs of the end users and the MI resources
- A Topic-Maps-based application: this one implements the construction of the semantic descriptors of resources and provides search functions for resource retrieval basing on kinds of knowledge maps. The main specificity of this application prototype is that it uses the Topic-Maps paradigm as knowledge model for the storage of the semantic descriptors of resources.

The major assets of this approach with respect to the business context of this work consist in the following:

- It supports both resource description and search: in fact, to enhance resource retrieval, we necessarily must tackle the source of the problem which is the resource description. These two purposes cannot be dissociated. Also, the S3 approach has an original aspect in dealing with these two purposes. It associates semantics to the resource during their retrieval (through the bottom-up retrieval strategy) and align them with the business needs by associating another level of description through the search patterns (i.e. the top-down strategy)
- It handles the existing resources in the company —which lack of semantics— and the new ones. Thus, the S3 approach is a scalable solution suitable for both exiting needs and new needs
- It centralizes the description of the business context (i.e., the manufacturing process activity and the process control activity) in two complementary semantic structures
- It continuously aligns the business needs of the process control actors with the resources produced with the COTS platforms through goal-oriented mechanisms
- It deals with the vocabulary heterogeneity even related to the resources or to the business needs

In summary, the S3 approach proposes a unique resource retrieval approach suitable to the process control context, which combines semantic retrieval techniques with an alignment aspect. Indeed, by considering the business needs in the core of the solution, either in the resource semantics (the manufacturing process ontology is business-need oriented), or in the search techniques (the search patterns captures the business needs), we continuously reduce the distance between the MI-resources and the users' needs with this approach.

Chapter 4: The Business Semantic Support

Résumé

Ce chapitre présente le support sémantique utilisé dans l'approche S3 pour la description de ressources dans le cadre du contrôle de processus industriels. Ce support sémantique est constitué de deux structures : une ontologie « Manufacturing Process »(MP) et un dictionnaire « Process Control » (PC).

L'ontologie MP fournit une description abstraite de l'activité de fabrication dans le domaine du semi-conducteur. Cette description est organisée autour de quatre vues : une vue fonction, une vue organisation, une vue données et une vue contrôle.

Le dictionnaire PC a pour objectif de « sémantifier » les terminologies et le vocabulaire associés aux informations manufacturières fournies par les ressources. Il rationalise en réalité les terminologies de la vue contrôle de l'ontologie.

L'usage de l'ontologie et du dictionnaire proposés est mis en évidence dans les chapitres qui suivent, notamment avec la création des descripteurs sémantiques de la stratégie de recherche « bottom-up » et avec la création des patterns d'alignement de la stratégie de recherche « top-down ».

4.1. Introduction

Building a semantic support in a company is not an easy task ; it requires a deep understanding of the business environment and the role of the semantics in this context. Business ontologies are usually identified as the basis of semantics. In our approach, the content of the semantic support has been identified step by step, by analyzing the business scope of the company resources and the business vocabulary related to these resources. Accordingly, a *business ontology* has been identified and built in a first step and a *domain-specific dictionary* has been identified in a second step.

The business ontology gives a description of the business activity of the company. In order to associate business semantics to MI resources, this one must be described elsewhere. A business ontology is used for this purpose. In the context of STMicroelectronics, the identified ontology is related to the manufacturing process activity and its process control description.

The domain-specific dictionary is proposed to gather the vocabulary related to the resource subjects and the vocabulary associated to the processing of the manufacturing information. In the context of STMicroelectronics, we proposed a process control dictionary.

This chapter presents the core content of the semantic support of our approach. After a brief overview of ontology construction approaches, we present the manufacturing process ontology and the process control dictionary.

4.2. The Business Ontology

4.2.1. Main Ontology Construction Techniques

Ontologies can be developed with different manners depending on some factors, such as the type of approach (e.g. with a global ontology, with local ontologies), the used strategy (e.g. from scratch, from existing knowledge sources), and so forth. [Visser et al. 2001] identified three standard approaches for ontology construction from existing sources: *single* (a global ontology is used with specialized ontologies), *multiple* (several local ontologies are used for each identified knowledge source) and *hybrid* (combines the single and multiple approaches).

Each approach of ontology construction can be developed using a type of strategy. Three strategies are well known in this field of study: top down, bottom up and combined strategy.

- **Top-down strategy**: the ontologies are built from scratch, basing on experts' knowledge and on the analysis of knowledge domain
- **Bottom-up strategy**: relies on the analysis of existing knowledge sources such as expert knowledge bases, business handbooks, thesauri, lexicons, enterprise taxonomies, etc.
- **Middle-out strategy**: is the combination of the top-down and bottom-up strategies

Furthermore, each ontology approach is achieved with a type of strategy using a mean of construction. Three means can be used in this case [Subhashini and Akilandeswari 2011]: manual, semi-automatic and automatic.

- **Manual mean**: the whole process of ontology construction is based on the human work
- **Semi-automatic mean**: the development process is based on some automatic techniques and on human intervention
- **Automatic mean**: the whole process is automatic. In this case, automatic processing techniques such as NLP and data mining are used for terms' extraction, filtering and classification.

In [Baazaoui et al. 2007], the authors proposed a classification of ontology construction approaches according to the use or non-use of a priori knowledge (e.g. thesauri, existing ontologies, etc.) and according to learning methods (e.g. NLP techniques, clustering and statistical techniques, etc.). Therefore, they distinguished four categories of approaches: construction from scratch, ontology re-engineering, collaborative construction and learning method using knowledge sources (e.g. texts, dictionaries, schemata, knowledge bases, etc.).

4.2.2. Construction of the Manufacturing Process (MP) Ontology

According to the problem statement of this work, we seek through the Manufacturing Process (MP) ontology to get a standard description of the semiconductor process. This description is also used to associate business semantics to each MI resource required in the manufacturing-process control. To well respond to our retrieval purpose through the MP ontology, it was important to investigate some ontology construction techniques with respect to our context of work, in order to identify how to build this ontology. The standard phases proposed in some well-known ontology construction approaches like the TOVE approach [Gruninger and Fox 1995], METHONTOLOGY [Fernandez et al. 1997], [Uschold and Gruninger 1996], On-To-Knowledge methodology [Sure et al. 2004] and so forth, mainly consist in: (a) specifying the purpose and scope of the ontology, (b) acquiring the concepts from knowledge sources and structuring them, (c) building (coding) and implementing the ontology, and (d) finally evaluating the consistency of the ontology. These phases are a good starting point to state the problem and structure the design process.

However, in our case, the main difficulty in building the MP ontology does not focus on the type or the number of phases required to design the ontology, but rather on how to identify the type of semantics (from knowledge sources and/or from experts) and how to structure it in a way to be efficiently used for reducing the gap between business needs and MI resources for the retrieval purpose. Also, this ontology must consider two knowledge domains: the manufacturing process and the process control. The study of the alignment-oriented approaches gave us an idea about the types of semantics needed to build an Information System aligned with the business. One of the approaches that have caught our attention is the ARIS approach. ARIS is used to describe business processes in a way to carry out the Business/IT alignment. It also seeks reducing the complexity of modeling business processes through its views, which also include a control view. Accordingly, we estimated that using the ARIS approach would be worth for the manufacturing-process description that includes a control description.

In this way, we chose to base the MP ontology on the ARIS views as a starting point for capturing the business concepts of the semiconductor activity (basing on the STMicroelectronics' case study). The design of the MP ontology was done following four main steps:

- **Scope identification**: we identified here the purpose and scope of the desired ontology. The scope is related to the manufacturing process description and in particular the concepts related to the ARIS views
- **Conceptual design**: encompasses knowledge capture and structuring through requirement modeling techniques. The modeling process facilitates the identification of the classes and relationships that will constitute the content of the ontology
- **Ontology construction**: we constructed the ontology basing on the classes and instances identified in the previous step
- **Evaluation/Validation**: the validation of the ontology was done in a first step with the help of business experts. The experimentation of the general semantic approach within STMicroelectronics contributed in evaluating the consistency of the content of the MP ontology.

Because of the complexity of the domain to describe, a first slight version of the ontology was developed. It was, afterwards, enriched progressively during the implementation of the S3 approach. We applied a middle-out strategy to capture a high-level and rational business description of the STMicroelectronics' activity. This description was captured through UML diagrams. Goal decomposition models were also used to capture business objectives.

4.2.2.1. Scope identification

The aim of this phase is to identify the type of concepts to capture, relative to the usage of the ontology. Basing on the ARIS concepts, the scope of the MP ontology lies on four views of description [Bouzid et al. 2013a]:

- *Organization view*: references the business activities of the company
- *Function view*: describes the manufacturing objectives of the company for each business activity
- *Data view*: refers to the business entities involved in the manufacturing activity
- *Control view*: describes the process control objectives and methods as implemented within the company

These views represent, in fact, the upper concepts of the MP ontology (Figure 4.2-1).

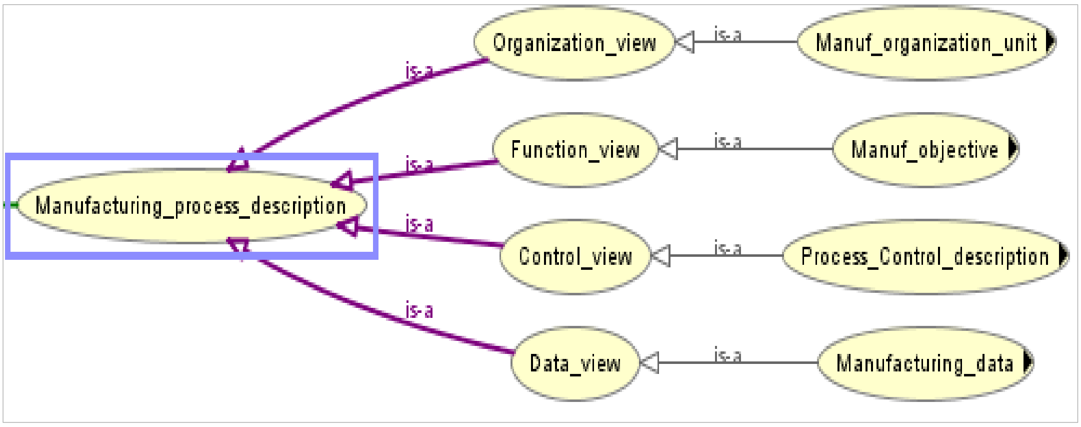


Figure 4.2-1 : The upper-concepts of the MP ontology

4.2.2.2. Conceptual design

Different techniques were used to develop the requirements of the four views of the manufacturing process description.

- Organization view

The concepts of the organization view were captured from the official documentation of the company. Each organization is an activity or a business function in the company. We kept the standard business functions related to the manufacturing process of the company. As examples of these functions within STMicroelectronics we can cite: *Process engineering, production, device, process control, defectivity, quality assurance, maintenance*, etc. We add to these functions, the main activities of the semiconductor manufacturing process, such as *Oxydation*, *Etching*, *Photolithography*, etc.

- Function view

The function view exposes business objectives. We use here the decomposition mechanism used in Requirement Engineering approaches to refine goals (*AND* decompositions). Two main knowledge sources were used here: experts' knowledge and the Top Pages[17] of the company. High-level manufacturing objectives are captured in this step. Figure 4.2-2 shows some manufacturing objectives of the STMicroelectronics Company.

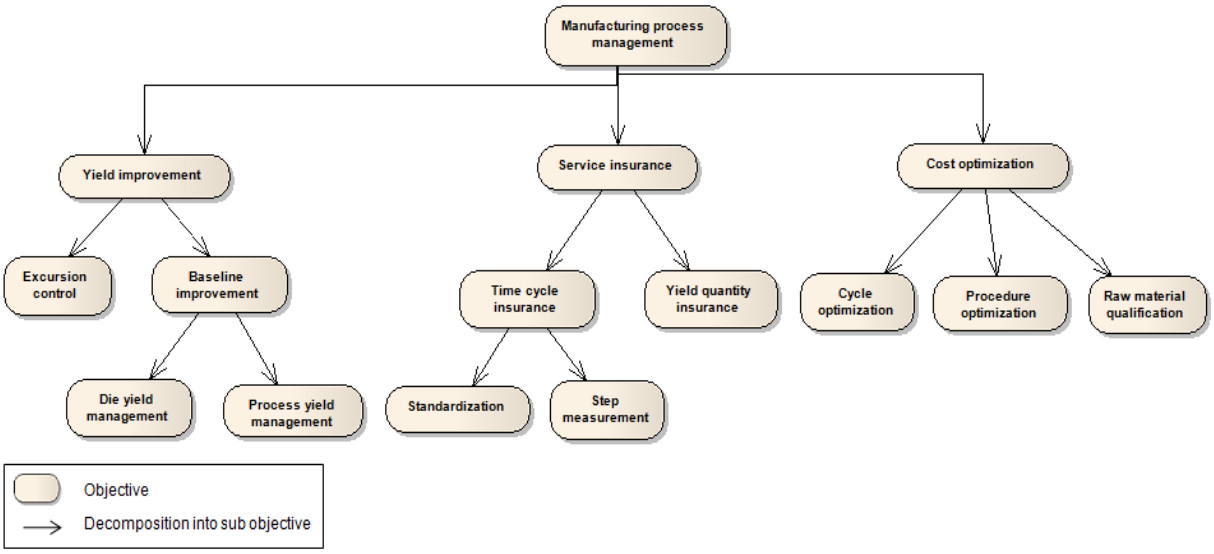


Figure 4.2-2 : Examples of manufacturing objectives within STMicroelectronics

- Data view

The data view was modeled in our approach with an UML class diagram. In the ARIS framework, the entity-relationship model is used since it was the most widespread designing method in the area of data modeling. We used, instead, the UML class diagram for its expressiveness in conceptual modeling. An overview of the STMicroelectronics' data that are monitored with the process control is given in Appendix A.

---

[17] The Top Pages in a company contains quantified business objectives refined by each activity and sub activity

- Control view

Regarding the control view, this one links between the views organization, function and data. It someway makes the link between the design results of the views that were designed separately for reasons of simplification. The control view of the ARIS approach is based on the EPC notation (a workflow-oriented language) [Mendling and Nüttgens 2005] to model the process flows[18]. In our context, we do not use the ARIS notation because it is not suitable to our context. In fact we do not need to describe a control flow in the control view; we need to describe instead the knowledge domain of the process control. Thus, our control view consists of a description of the manufacturing process control. Basically, this view represents the process control function as implemented within the company. The process control activity (described in the context of work, cf. Chapter 1) is mainly based on a set of control methods and analysis techniques to support control objectives.

We identified three sub types of concepts for the control view: *control objectives*, *control methods* and *analysis techniques*. The control objectives were modeled with a goal decomposition model, as done for the manufacturing objectives (Figure 4.2-3).

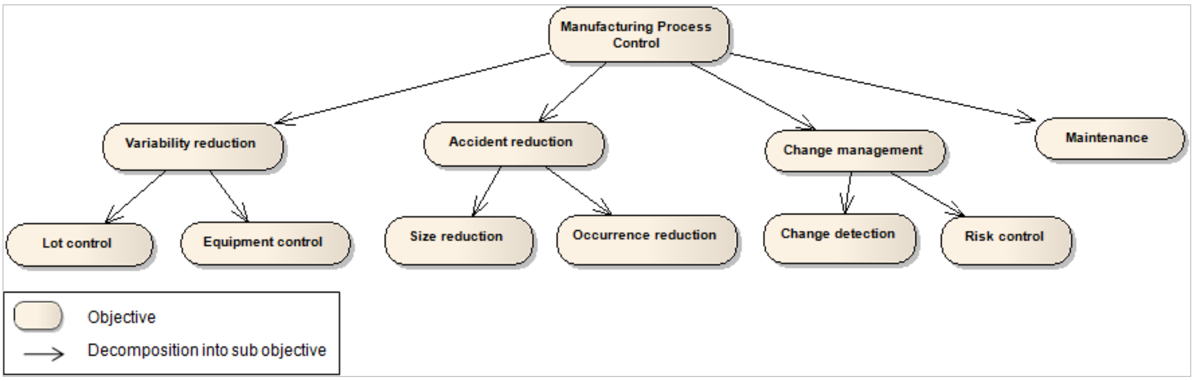


Figure 4.2-3 : Examples of process control objectives

With the help of domain experts, an inventory of control methods with their key indicators was done to identify the control methods, the indicators and the techniques used for the process control in the company.

Figure 4.2-4 presents an overview of the process control description of STMicroelectronics. Each indicator (e.g. OOC, OOS, CpK, etc.) generated in a MI resource is based on a standard control method (e.g. SPC, FDC, etc.) that uses statistical techniques (e.g. variance decomposition, population comparison, etc.). When an indicator is not based on a standard method, it is then specific to a control domain identified within the company (like production yield, risk assessment, etc.). The used methods and techniques in a control domain are often created for a specific need in the company (ad-hoc methods and techniques). These methods can be subject to changes and improvements.

---

[18] An example of this description as proposed by ARIS [Ferdian 2001] is presented in Appendix A

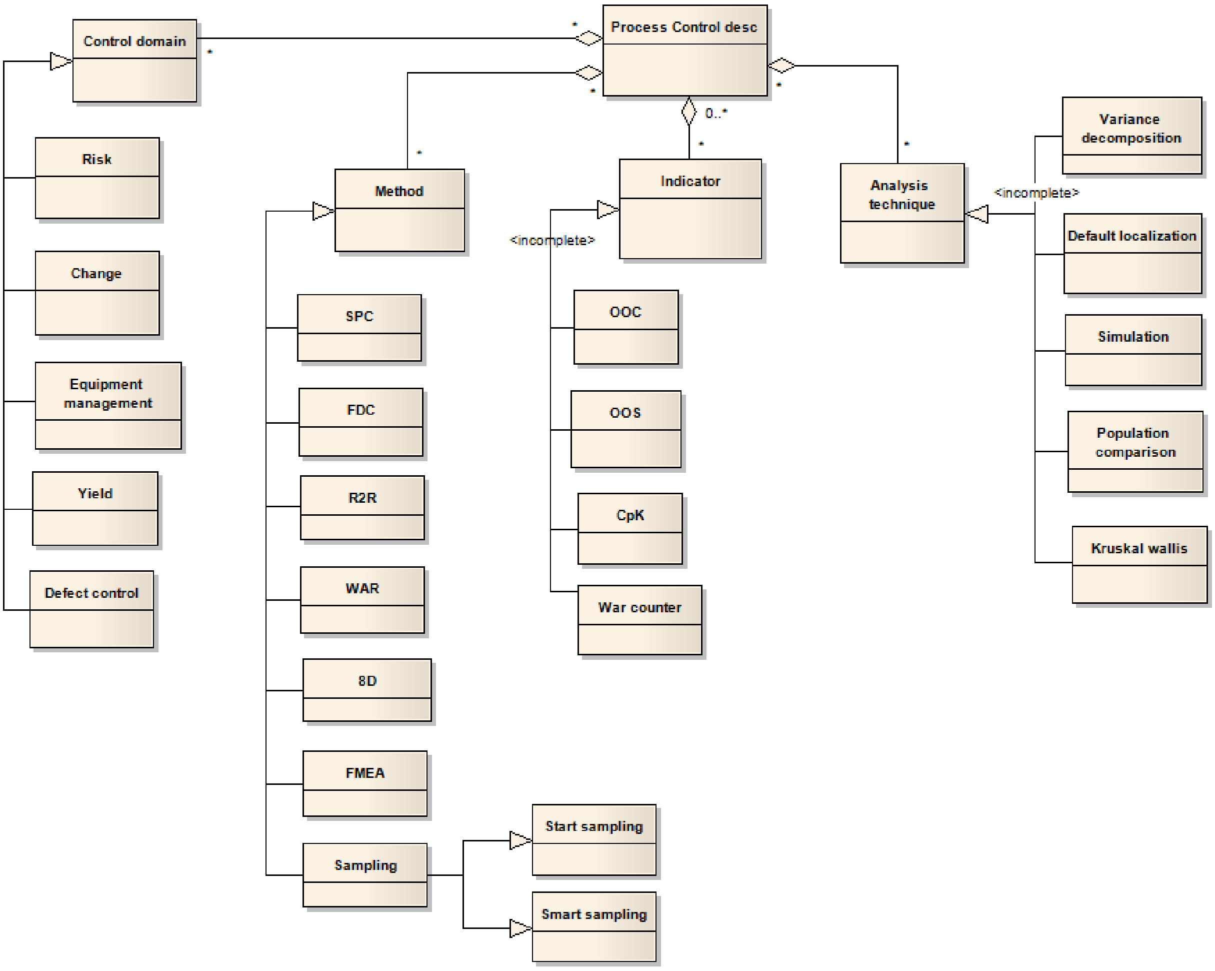


Figure 4.2-4 : Conceptual description of the process control of STMicroelectronics

Knowing that the MI resources result from process control methods and techniques, the control view constitutes hence the link between the descriptive views of the manufacturing process and the concerned MI resources of the company.

4.2.2.3. Ontology construction

Basing on the concepts captured in the previous step, we structured them into classes, sub classes and instances, under the upper concepts of the ontology. We show examples of concepts in Table 4.2-1. Each class of a view is in fact a type of business concept. The set of business activities identified in the first step constitutes the instances of the organization view. The first hierarchy of objectives of the goal model is used as sub classes in the function view (Manuf_objective). The rest of the objectives of the goal decomposition constitute instances of this view. The hierarchical relations between the objectives (either the manufacturing objectives or the control objectives) are expressed afterwards with relations.

Classes	Sub classes	Instances
Manuf_organization_unit	-	*Process engineering, Production, Defectivity, ...*
Manuf_objective	Yield_improvement	*Excursion control, Baseline improvement, Die yield management, Process Yield*

		management
	Service_insurance	*Cycletime insurance, Standardization, Step measurement, Yield quality insurance*
	Cost_optimization	*Cycletime optimization, Procedure optimization, Raw material qualification*
Manufacturing_data	-	*Lot, Wafer, Equipment, Chamber, Technology, Operation, Step, ...*
Process_Control_description	Control_objective	*Variability reduction, Lot control, Equipment control, size reduction, Occurrence reduction, Change management, Risk control, Maintenance*
	Control_method	*SPC, FDC, R2R, WAR, 8D, FMEA, Start sampling, Smart sampling, Risk, Yield, Equipment management, Defect control*
	Analysis_technique	*Population comparison, Variance decomposition, Kruskal wallis, ...*

Table 4.2-1 : Classes and sub classes of the MP ontology with some examples of instances

With the proposed upper concepts, each view is linked at least to another view. Globally, the concepts of the control view constitutes the main link between the concepts of the data, function and organization views of the MP ontology, as depicted in Figure 4.2-5.

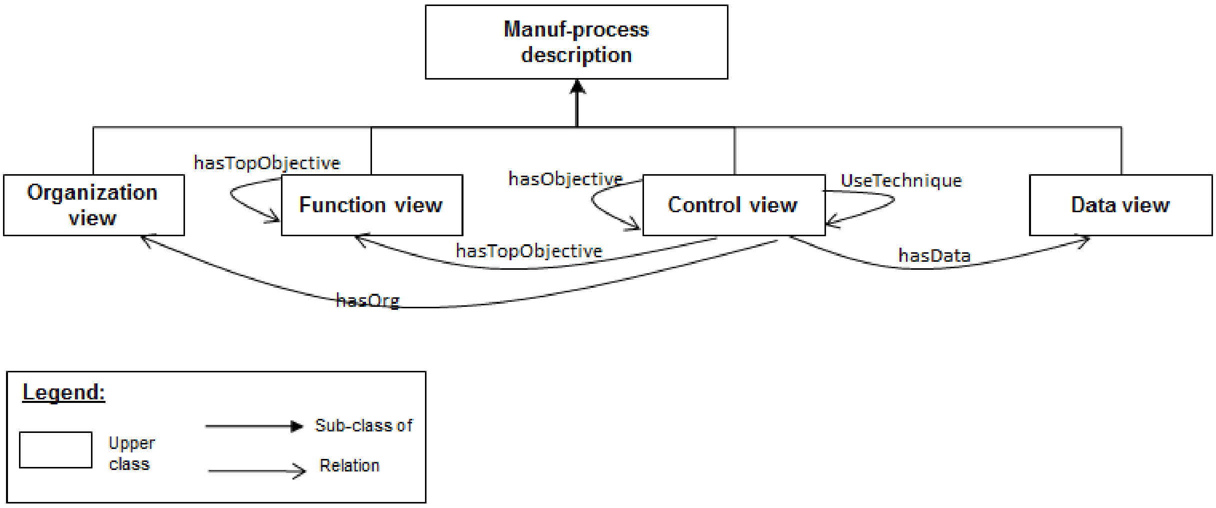


Figure 4.2-5 : Relations between the upper concepts of the MP ontology

Table 4.2-2 summarizes the identified relations between the MP ontology concepts with some examples.

Relations	**Sources**	**Targets**	**Examples**
hasObjective	Control_method,	Control_objective	*hasObjective* (SPC, Lot_control)
	Control_objective,	Control_objective	*hasObjective* (Lot_control, Variability_reduction)
hasTopObjective	Control_objective	Manuf_objective	*hasTopObjective* (Variability_reduction, CycleTime_optimization)
	Manuf_objective	Manuf_objective	*hasTopObjective* (Cost_optimization, Cycletime optimization)
hasData	Control_objective	Manufacturing_data	*hasData* (Lot_control, Lot)
	Control_method	Manufacturing_data	*hasData* (FDC, Equipment)
hasOrg	Control_method,	Manuf_organization	*hasOrg (SPC, process_control)*
	Control_objective	Manuf_organization	*hasOrg (Equipment control, Defectivity)*
useTechnique	Control_method	Analysis_technique	*useTechnique (FDC, Population_comparison)*

Table 4.2-2 : The types of relations identified for the MP ontology

We present in Figure 4.2-6 an UML representation of the resulting ontology. We show in this representation both the class level and the instance level with examples of relations between concepts.

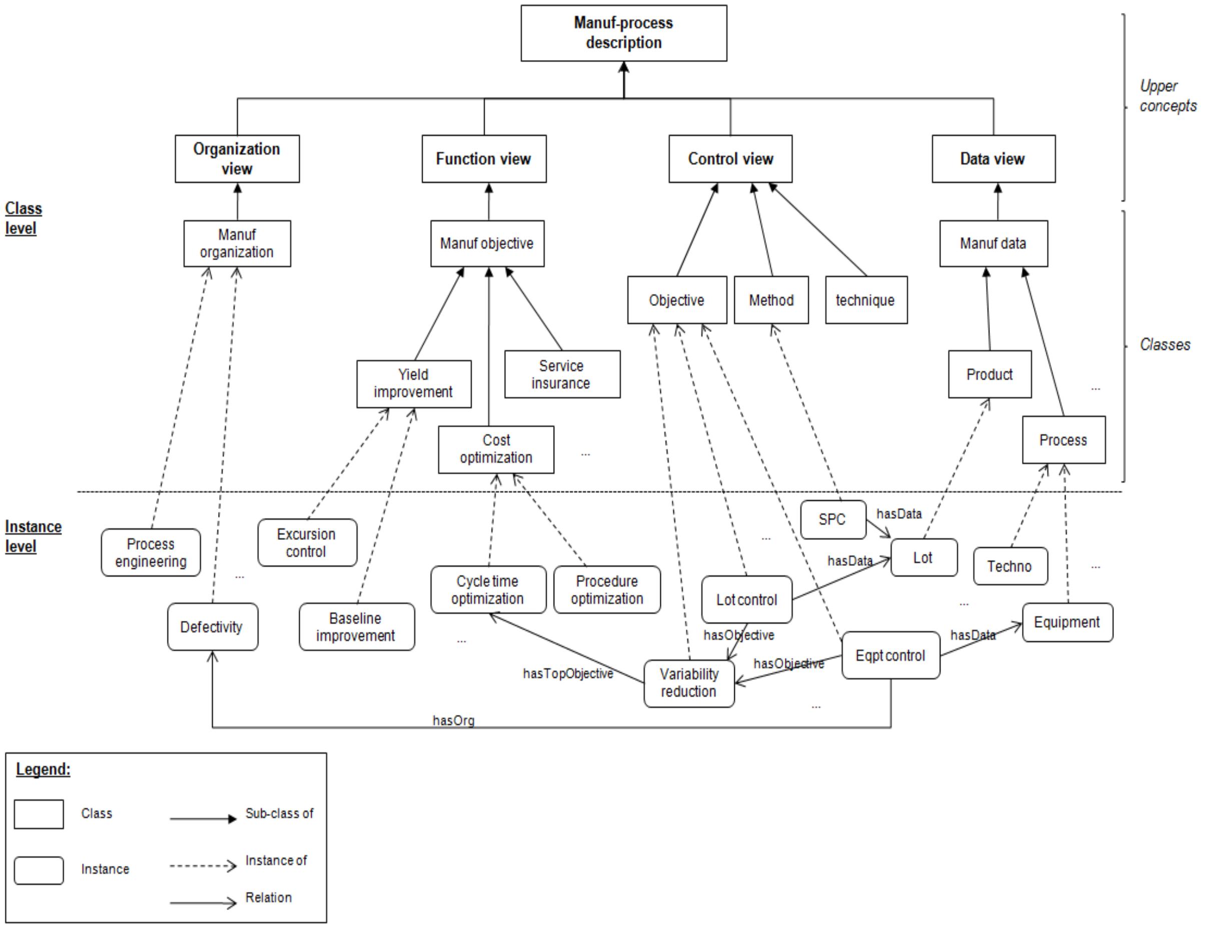


Figure 4.2-6 : Conceptual representation of the classes and instances of the MP ontology of the semiconductor industry

4.2.2.4. Validation

The MP ontology is used to enable the capture of the necessary semantics for resource description and retrieval. This ontology is precisely used to ferret out the role of these resources in the manufacturing process by mapping the ontology concepts with the MI resources. However, it is a bit difficult at this stage to validate this ontology without using it for the description and retrieval purposes. We checked and validated the content of the ontology in a first step manually with the help of business experts, which are obviously the most qualified for this task in the company.

Thus, by reviewing the MP ontology with the business experts, we noticed a big lack of vocabulary related to the control terminologies used with the MI resources. Indeed, there are a lot of declinations of words and lexical terms (e.g. variants, abbreviations, etc.) related to the control domain and usually used by the business actors in the company. These lexical terms result from the vocabulary used in manufacturing systems and by the COST tools used to produce the MI resources. Such terminologies are subject to change. Even so, they are often reused and popularized by the experts and engineers of the company. The MP ontology is mainly used to provide a business description of the business process, i.e. a manufacturing process description including a standard description of the process control. It will be unmanageable to put all the vocabulary related to this domain of study in the MP ontology, in particular because there are many semantic aspects, which are sometimes informally used by the process control actors. Using a second semantic structure will be a more interesting solution. We propose then a domain-specific dictionary to reference the vocabulary related to the process control activity in order to gather all the used terminologies, to rationalize the heterogeneity of this vocabulary and standardize its use with the MI-resource description and retrieval.

4.3. The Domain-Specific Dictionary

4.3.1. Purpose

There are several dictionaries in the literature used in Information System research-works. They are widely used to improve information retrieval on the web and to realize the interoperability between applications. Main existing dictionaries are linguistic ones and aim at defining the vocabulary of a domain. WordNet[19] is a widespread linguistic dictionary, related to the English language. However, such a dictionary is more suitable to generic terminologies related to a commercial domain, a training field or other, than to the vocabulary that is used in manufacturing companies. For example, the WordNet dictionary gives the definition of *statistics*, of a *process* and of the *control*, but it does not provide the meaning of the *statistical process control (SPC)*, whereas *SPC* is a standard method used to monitor and control a process, in particular in manufacturing companies. Thus, we chose to use a domain-specific dictionary to reference and unify the used concepts of the business domain supported by the MI resources of the company. According to our context of work, the company resources are related to the process control activity. Thus, the required dictionary will focus on this activity domain.

We chose to use a dictionary instead of an ontology because an ontology is intended to give a whole description of a domain using a set of classes, individuals and relations, so to create and infer more semantics with business rules. The dictionary that we propose only aims at gathering and unifying the vocabulary related to a business-specific domain. Moreover, its basic structure and its usage come closer to the notion of dictionary. Actually, the proposed dictionary differs from other known semantic structures, like ontologies, thesauri and lexicons, in the following points:

- It provides the vocabulary of concepts to a domain including a linguistic aspect (e.g. definition, variants, synonyms, etc.)
- It surrounds a concept with a set of concepts to include its scope of definition and scope of use
- The structure of the dictionary is not hierarchic. The only used relations consist of the hypernym and hyponym concepts, which are usually used in semantic structures.

4.3.2. General Definitions

We propose in our approach a simple dictionary model for semantic definition of concepts related to an industrial activity domain.

We define a domain-specific dictionary $\mathcal{D}$ with a set of entries E that include concepts.

$$\mathcal{D} = \{E_1, E_2, \dots, E_n\}$$

---

[19] http://wordnet.princeton.edu/wordnet/

An entry E represents a concept c defined in the dictionary with a set of entities which constitutes its semantic description S. Thus, an entry E is defined with the 2-tuple:

$$E = < c, S >$$

Each semantic description S of a concept c is a 6-tuple:

$$S = < d, V, N, y, H, K >$$

Where:

d is the definition of the concept c expressed in natural language, such that for each concept c there is one definition d

V is the set of variants of the concept

N is the set of synonyms of the concept

y is the hypernym of the concept

H is the set of hyponyms of the concept

and K is the set of key concepts that define the scope of use of the concept

Examples of these concepts are presented in the next sub-section.

The integrity constraints that are applied to the dictionary entries are the following:

- $E = (\mathrm{c}, \{d \neq null, \mathrm{y} \neq null, \mathrm{K} \neq \emptyset\})$

  When an entry is created, at the initial state, the definition, the hypernym and the key concepts are required. The variants, the synonyms and the hyponyms can be empty (i.e. a minimum of semantics is required to create an entry in the dictionary).

  - Each entry is unique in the dictionary
  - $\forall c_i \in \mathcal{D} \rightarrow \exists 1\ d_i \in E_{c_i}$, where d_i is the definition of c_i

    Each concept c_i of the dictionary entry has one definition d_i. In fact, we can find in linguistic dictionaries several definitions for a same concept. These definitions depend on the context of use of the concept. In our case, there is only one context for a concept, because this context is related to a specific activity. For this reason, a concept here has one definition.

  - $\forall c_i \in \mathcal{D} \rightarrow \exists c_j \in \mathcal{D}\ such\ that\ ((y_{(c_i)} = c_j) \vee (y_{(c_i)} = c_i))$

    Each concept c_i has a hypernym which can be another concept of the dictionary or can be the concept itself. In the last case, the concept is not included in a more general concept

  - $\forall k_i \in \mathcal{D} \rightarrow k_i \neq y \wedge (k_i(c_i) \notin \{\mathrm{H}, \mathrm{V}, \mathrm{N}\})$

    A key concept k_i is not a hypernym, and does not belong to the set of hyponyms, variants or synonyms. In fact, the key concepts are other business concepts often associated to the defined concept c_i when it is used

These definitions enable to set up a domain-specific dictionary that can be used with the retrieval strategies of the S3 approach. Furthermore, the proposed structure is defined in a way to coherently lead the mapping technique of the bottom-up strategy of the S3 approach.

4.3.3. The Process Control Dictionary

The PC dictionary [Bouzid et al. 2013a] is composed of a list of concepts related to the process control activity. Concretely, a process control concept could be a control indicator, a standard control method, or a control domain identified within the company. Examples of concepts are illustrated in Table 4.3-1.

Considering the example of OOC, it has two variants (e.g. out of control, oc) and one synonym (e.g. not under control). The difference between a variant and a synonym is that the variant represents another form of a concept, like an abbreviation. For example, the *OOC* concept is also written as *Out Of Control*.

The concept *OOC* represents a concept of the SPC method, then, its hypernym is *SPC.* In fact, the hypernym encompasses one or a set of concepts, such a kind of generalization. The hyponym is then a kind of specialization. For example, the hyponyms of *SPC* are *OOC*, *OOS*, *CLM*, *Cpk* in the example.

Otherwise, the concept OOC is an indicator that usually takes the form of a control chart with two control limits and a target line. Thus, the key concepts that provide its scope of use include *Limit*, *target* and *control chart*.

Currently, the PC dictionary of STMicroelectronics contains 35 entries and 232 concepts in whole (including variants, synonyms, hypernyms, hyponyms and key concepts).

Entries	**Entities**	**Instances**
E1	**Concept id**	***ooc***
	Definition	*An ooc happens when the production measures exceed the control limits*
	Variants	*out of control, oc*
	Synonyms	*not under control*
	Hypernym	*spc*
	Hyponyms	-
	Key concepts	*limit, control chart, ucl, lcl, target*
E2	**Concept id**	***spc***
	Definition	*A standard method for the statistical control of (manufacturing) processes. It uses production measures in control charts to predict the variations that can result during the production*
	Variants	*statistical process control*
	Synonyms	-
	Hypernym	*spc*
	Hyponyms	*ooc, oos, clm, cpk*
	Key concepts	*variability, control chart, limit, target, production measure*

Table 4.3-1 : Examples of instances of the PC dictionary

We can notice that the domain-specific dictionary handles the business vocabulary related to the company resources, whereas the business ontology formalizes the link between a business vocabulary and business needs. The proposed dictionary helps, in this way, in bridging the gap between the resources and their basic business usage in the S3 approach. It plays a key role in harmonizing the control vocabulary during the construction of the semantic descriptors of resources. Morphological forms, abbreviations and synonyms, etc. can be found in the dictionary.

The MP ontology and the PC dictionary are linked through the process control methods and control domains. The PC dictionary references the vocabulary related to the process control activity that is used in MI resources (including the type of methods, indicators' concepts, etc.), whereas the MP ontology gives the link between a process control method/domain and how it is used in the manufacturing activity.

4.4. Conclusion

The semantic support presented in this chapter relies on two semantic structures: a MP ontology and a PC dictionary. The role of the MP ontology is to capture the semantics related to the core business, so to make the resources close to business needs. The role of the PC dictionary is to define and reference the heterogeneous terminologies of the process control domain in a standard schema structure. It also bridges the gap between the vocabulary related to the MI resources and their low-level business usage.

On other hand, one interesting asset of the proposed MP ontology is that its structure tackles a business description focused on the purpose of use of the ontology. This structure covers the business needs in the company through the function and control views. Also, the ARIS views that were used for capturing the business description of the company allow supporting, in a single ontology, different levels of description coherently linked. Moreover, it was important in this context to have these levels in the same structure, in particular because we have two distinct business knowledge domains (i.e. the process control and the manufacturing process) but inevitably interrelated in many aspects. Indeed, a process control activity always addresses manufacturing needs.

Finally, this semantic support has been centralized in this way to enable its flexible use in the S3 approach. The manufacturing process ontology and the process control dictionary are used in the retrieval strategies to support the construction of semantic descriptors of resources and to support resource retrieval with goal-oriented mechanisms.

Chapter 5: The Semantic Descriptors of Resources

Résumé

Ce chapitre décrit la stratégie de recherche ascendante qui repose sur la création de descripteurs sémantiques à partir d'un ensemble de ressources sélectionnées. Chaque descripteur associe en réalité une description métier à une ressource pour fournir son rôle et son usage dans la maîtrise des procédés de fabrication. Cette description comporte une partie relative au « process control » et une partie relative au processus métier (le processus de fabrication). Une technique de mapping de concepts est proposée pour permettre l'automatisation de la création de ces descripteurs sémantiques. Cette technique est basée sur deux catégories d'algorithmes : un algorithme de calcul de similarité entre concepts et un algorithme d'inférence de concepts.

L'algorithme de calcul de similarité permet d'identifier la description « process control » la plus appropriée des ressources pour exploiter les quatre vues de l'ontologie et analyser les relations entre concepts. Une formule de calcul de similarité basée sur la distance de Levenshtein et le coefficient de Dice est proposée.

L'algorithme d'inférence de concepts permet d'inférer les concepts métier qui décrivent le mieux les ressources, en utilisant l'ontologie et un ensemble de règles métier préétablies. Cet algorithme est supporté en partie par la technique du chainage avant (« forward chaining »).

Cette technique de mapping a été utilisée pour sémantifier les ressources existantes dans l'entreprise, afin de faciliter leur localisation et leur recherche pour les acteurs du « process control ».

5.1. Introduction

This chapter describes the bottom-up retrieval strategy of the S3 approach which relies on building semantic descriptors of resources. The bottom-up retrieval is required for resources that are already used for the process control in the company but need semantics. Thus, we propose in this chapter a **semantic descriptor model** for the MI resources and an automatic **semantic mapping technique** to build the descriptors. The mapping technique deals with abstract inputs composed of a set of concepts related to the resources. The output represents the concepts that constitute the content of the semantic descriptors of the MI resources. The MP ontology and the PC dictionary are used to support the construction of the semantic descriptors.

Note that, because of the specific format and content of the MI resources in our context (STMicroelectronics), any concepts in the MI-resources' names, the used entities of the processed data and the business taxonomies of the resource location are used to start the creation process of the semantic descriptors.

5.2. The Semantic Descriptor Meta-Model

We propose a semantic descriptor to capture the main types of semantics needed for the resources. This semantics constitutes the first level of description that makes the resources close to the business needs. This description is used in the retrieval system to match the MI resources with the low-level business needs of the end-users.

Figure 5.2-1 depicts the meta-model of the proposed semantic descriptor.

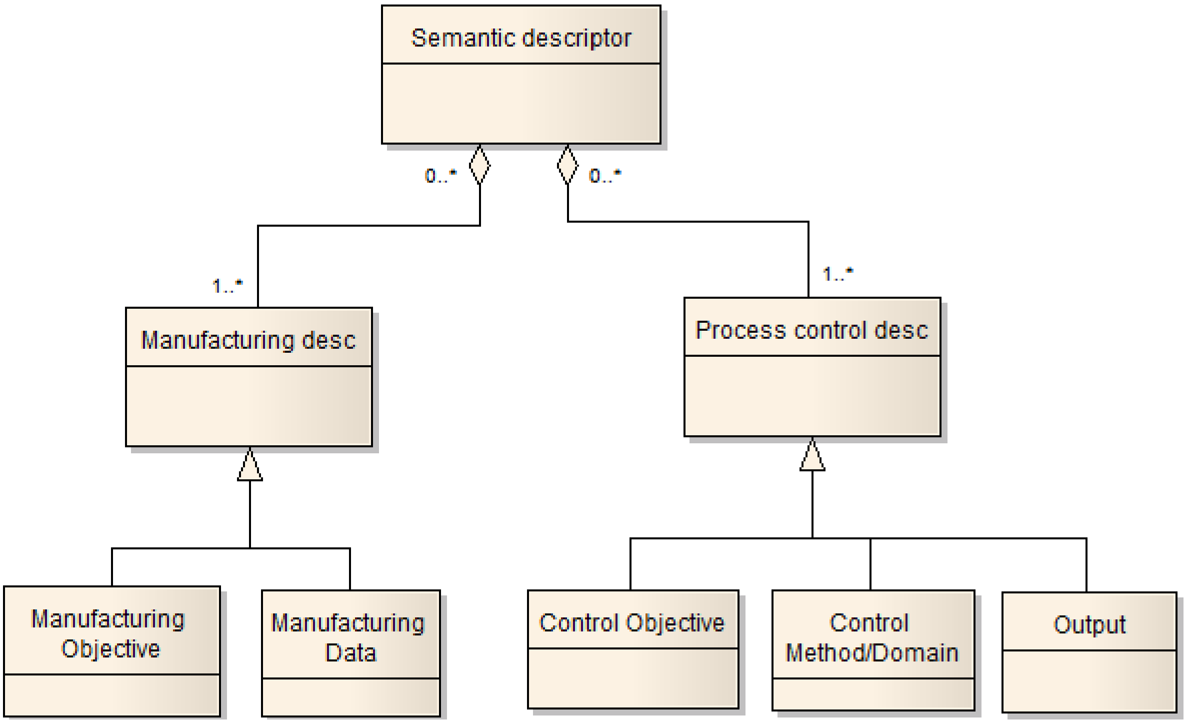
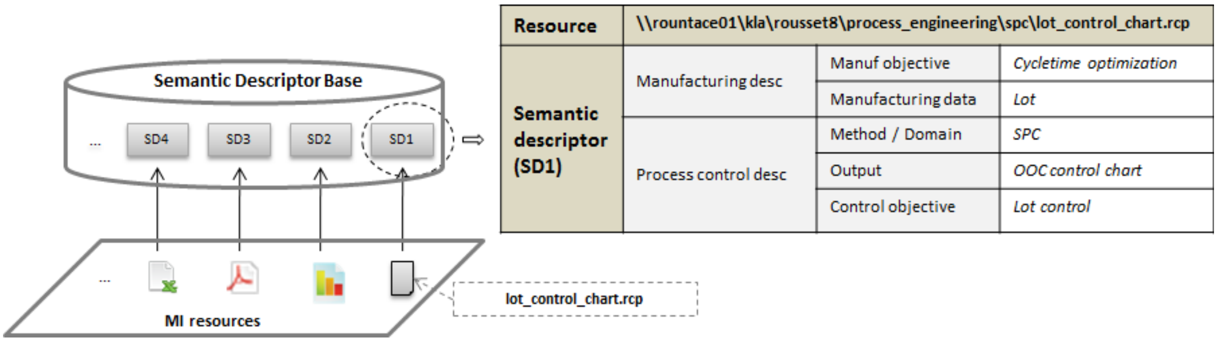


Figure 5.2-1 : The semantic descriptor meta-model

We can notice that a semantic descriptor is composed of two types of knowledge: manufacturing type (manufacturing objectives and manufacturing data) and process control type (control methods and domains, indicators and statistical techniques). The whole description mainly focuses on the low-level business usage of the MI resources, i.e. their control usage and which business purpose they address.

In this way, a semantic descriptor is composed of standard ***control methods*** (and control domains), the ***data*** processed with the control methods and the type of ***outputs*** which are mainly types of control indicators or specific outputs related to statistical techniques. The process control has also ***control objectives*** directly associated to ***manufacturing objectives***. In fact, each control objective is intended to satisfy a manufacturing objective independently of any manufacturing activity or organization.

In the example of Figure 5.2-2, *lot_control_chart.rcp* is an example of a resource that provides *control charts* as output. This resource is based on the *SPC* method, uses *lot* measures as data to *control lots* (control objective) during the manufacturing process in order to *optimize the cycle time* of the wafer processing (manufacturing objective).



Resource	**\\rountace01\kla\rousset8\process_engineering\spc\lot_control_chart.rcp**		
Semantic descriptor (SD1)	Manufacturing desc	Manuf objective	*Cycletime optimization*
		Manufacturing data	*Lot*
	Process control desc	Method / Domain	*SPC*
		Output	*OOC control chart*
		Control objective	*Lot control*

Figure 5.2-2 : Example of a resource captured in a semantic descriptor

All the concepts identified or that can be identified for a semantic descriptor belong to the manufacturing process ontology. They are captured from this one. This is why it was important in the approach to build a business ontology enough representative of the business activity, so that the semantic descriptor of each resource can be easily built.

Basically, a semantic descriptor has two usages in the approach:

- resource description: to make the resources understandable for the end users and searchable by retrieval systems
- User-query matching: the matching between user queries and the MI resources can be effectively done (at least) with the semantic descriptors because they provide a business-oriented description to the MI resources.

5.3. Semantic-Descriptors’ Creation Using Semantic Mapping

5.3.1. Around Mapping

Mapping techniques aim at identifying the correspondences between concepts. These techniques rely on a combination of IR techniques, statistics and artificial intelligence [Lin 2007]. We briefly review some of these techniques.

IR techniques and statistics are used to determine the syntactic matching of concepts, known as string similarity. In the first IR research-works, [Belkin and Croft 1987] distinguished two main retrieval techniques: *exact match* and *partial match*. The *exact match* is the most simplest and used technique. It requires that the concepts to match exactly have the same syntax. The disadvantage of this technique is that it does not take into account the relative importance of concepts, even by using dictionaries and thesauri. When this technique is used in a search engine, it could miss relevant information resources whose representations match the query only partially. The *partial match* technique seems to be more interesting since it broadens the scope of matching. This technique is mainly based on statistical notions, in particular for document retrieval. Examples of used techniques include the TF-IDF metric, edit distances, the vector-space model, pattern matching, and so on. These techniques can be strengthened using auxiliary resources like ontologies, dictionaries and thesauri which give semantics to the concepts to match [Lin 1998; Cohen and Fienberg 2003; Lin 2007; Zhong et al. 2009].

Globally, we classify main mapping techniques into syntactic matching and reasoning matching techniques. The well-known and used techniques are:

- Syntactic matching (string similarity):

Levenshtein distance: known as edit distance [Ristad and Yianilos 1998]. The edit-distance between two strings a and b is defined as the minimum number of edits needed to transform the string a into the string b. The edits' operations are insertion, deletion or substitution of a single character. The levenshtein distance is defined as follows:

$$lev_{a,b}(i,j) = \begin{cases} \min(\mathrm{i},\mathrm{j}) = 0 \\ \min \begin{cases} \mathrm{lev}_{\mathrm{a,b}}(\mathrm{i}-1,\mathrm{j})+1 \\ \mathrm{lev}_{\mathrm{a,b}}(\mathrm{i},\mathrm{j}-1)+1 \\ \mathrm{lev}_{\mathrm{a,b}}(\mathrm{i}-1,\mathrm{j}-1)+\delta(\mathrm{a_i},\mathrm{b_j}) \end{cases} \end{cases}$$

The first minimum formula corresponds to deletion operation from a to b, the second minimum formula corresponds to insertion operation and the third formula corresponds to matching or mismatching between a and b where $\delta(\mathrm{a_i},\mathrm{b_j}) = 0$ if $a = b$ and 1 otherwise.

Hamming distance [Chapman 2008]: given two strings a and b having a length n, the Hamming distance calculates the minimum number of substitutions required to change the string a into the string b. For example, a = nice and b = face, then the Hamming distance is 2.

Jaro-winkler distance [Chapman 2008]: this distance is suitable for short strings. It is a variant of the Jaro-distance metric. The Jaro metric is based on the number and order of common characters between two strings. The Jaro distance d_j of two strings a and b is:

$$d_j = \frac{1}{3}\left(\frac{m}{|a|}+\frac{m}{|b|}+\frac{m-t}{m}\right)$$

where m is the number of matching characters and t is the half number of transpositions such as for each two characters from a and b, they match only if $a_i = b_j$ and not farther than $\left\lfloor\frac{max(|a|,|b|)}{2}\right\rfloor - 1$

In fact, the number of transpositions identifies if the common characters of the two strings are in the same order, otherwise no transpositions are needed.

Based on the Jaro distance, the Jaro-winkler distance increases the rating to strings that match from the beginning of the strings. It uses a scale p (usually set to 0.1) and a length l related to the length of the common characters up to a maximum of 4 characters from the beginning of strings (e.g. a= data and b = daily, then l = 2):

$$d_w = d_j + \left(l * p(1 - d_j)\right)$$

Jaccard similarity coefficient [Chapman 2008] [Hai 2005]: it provides a kind of average similarity value between two sets. Thus, the similarity between the sets of strings A and B is calculated using the size of intersection and union of the sets:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Sørensen-Dice coefficient [Chapman 2008][Hai 2005]**:** Dice is a metric used to compare the similarity of two samples. It is a bit similar to the Jaccard coefficient. The similarity measure is the twice ratio of the number of elements shared by the two samples (i.e. that correspond) over the total number of set elements:

$$Dice(A,B) = \frac{2|A \cap B|}{|A| + |B|}$$

Hybrid distances [Lin 2007]: e.g., Level2JaroWinkler, Level2Levenshtein, etc. They are based on the combination of two similarity distances. The basic idea relies on dividing the strings a and b into subsets of strings and applying a secondary distance function on the subsets.

- Reasoning-based technique:

Use of relations [Lin 2007]: if two relations and related concepts are similar, then the two compared concepts may be similar or equivalent

Example: *If (located_at(hotel,area) and (located_at(hotel,city)) then area ≡ city*

Sibling concepts rule [Lin 2007]: given two sibling concepts C_1 and C_2, and two other sibling concepts C_3 and C_4, if $C_1 = C_3$ then $C_2 \equiv C_4$

Super(sub)-concepts rules [Lin 2007]: if the direct super-concepts or sub-concepts of two concepts are similar, the two compared concepts may be also similar.

Logical formulas: formal knowledge-representation languages can be used to express logic like DL [Baader et al. 2003], FOL [Smullyan 1995a], propositional logic [Smullyan 1995b], etc. The basic idea relies on deducing new facts starting from on a set of facts and rules. The reasoning is mainly performed with the help of kinds of reasoners or expert systems [Parsia and Sirin 2000; Ardelt 2004; Corby et al. 2004].

5.3.2. Overview of the Process

The general mapping system involves three parts: a set of inputs, the semantic mapping process and type of outputs. Figure 5.3-1 shows an abstract representation of the inputs and outputs of the semantic mapping system.

The inputs represent set(s) of concepts selected from a raw lexical chain or key words (related to MI-resource subjects). The mapping process is based on a syntactic matching semantically supported with the PC dictionary. The outputs represent the set(s) of concepts corresponding to the input set(s). These concepts are identified and selected from the ontology. In fact, the purpose of the mapping mechanisms here is to search for the corresponding description to each MI resource. Thus, the output types must be identified before applying the mapping process in order to target them in the ontology. In our approach, the target concepts to obtain in the output are the concepts of the semantic descriptor meta-model, because this mapping technique is intended to build semantic descriptors.

We identified two interesting string similarity techniques in this approach for the syntactic matching of concepts:

- *the dice coefficient:* this metric calculates the number of matched concepts between the input concepts (which are a set of strings) and the concepts related to each entry in the domain-specific dictionary. The dice coefficient is used here to assess the similarity of two sets
- *the edit distance*: the *levenshtein* distance is used to measure the similarity between two strings. It calculates the distance between the input concepts and the concepts of the dictionary. Only the best edit-distance between two concepts in a set is selected.

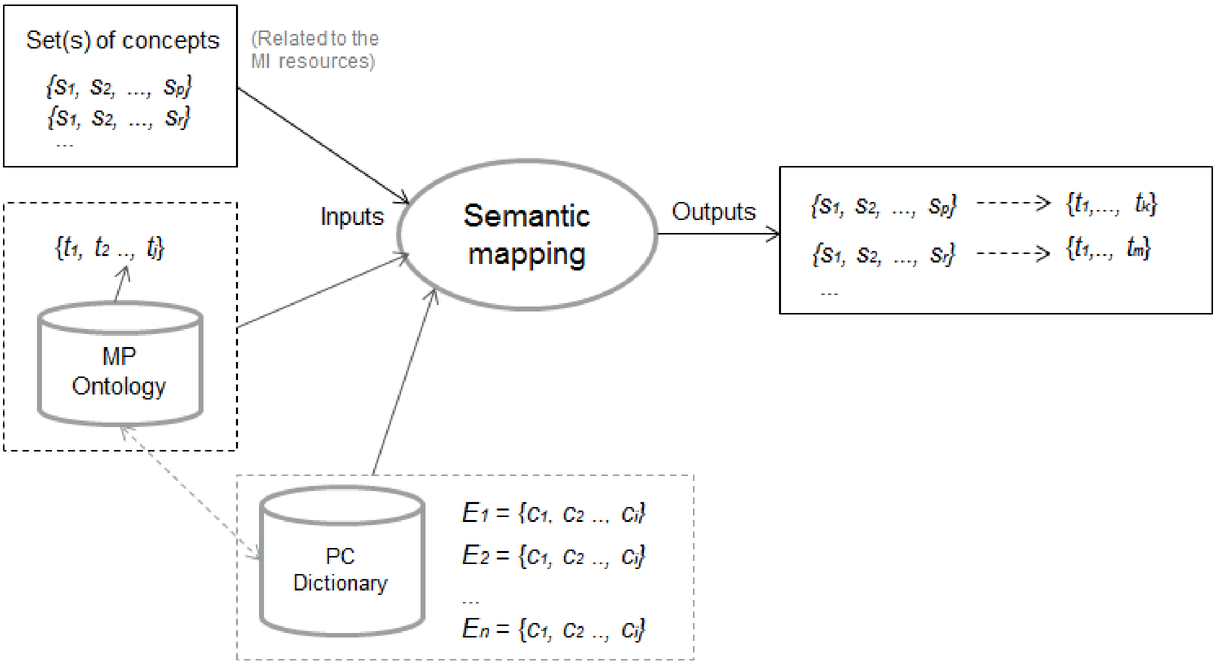


Figure 5.3-1 : Abstract representation of inputs and outputs of the mapping process

These techniques were chosen according to our context and purpose of mapping. In fact, the mapping techniques are generally used for the mapping of semantic structures (like mapping of two ontologies), whereas we use the mapping[20] techniques here to identify correspondences between MI resources and their eventual business description.

The mapping process relies on three major steps:

- ***Concepts' filtering***: consists in tokenizing the inputs and eliminating stop words, numbers, symbols, and so on, in order to keep relevant concepts only
- ***String-similarity measuring***: the domain-specific dictionary is used in this step. It provides the required semantics to enhance the syntactic matching of concepts. A combined similarity measure is used. The best similarity measure between the input sets and the dictionary entries is selected at the end

---

[20] We use the term mapping in its general sense in our approach and in the rest of the manuscript, unrelated to ontology-mapping works

- ***Reasoning with ontology relations***: the aim here is to find the target concepts whose types were identified before applying the process. The business ontology is used in this step. Thus, this step consists in searching in the relations between the concepts resulting from the previous step and the ontology concepts in order to get the ones that correspond to the types of the concepts targeted. An inference algorithm is also used to enhance the search of the corresponding target concepts in the ontology.

These steps are repeated for each set of concepts (S_{ci}) used as input. Algorithm 1 shows the sketch of the mapping process.

Algorithm 1 : Sketch of the mapping process

```
Input: sets of concepts $\mathbb{S} = \{S_{c1}, S_{c2}, \ldots, S_{cx}\}$ where $\forall i \in [1 \ldots x]\ S_{ci} = \{s_1, s_2, \ldots, s_p\}$, Dictionary $\mathcal{D}$, Ontology $\mathcal{O}$
Output: sets of target concepts $\mathbb{T} = \{T_1, T_2 \ldots, T_{ty}\}$ where $\forall j \in [1 \ldots y]\ T_j = \{t_1, t_2, \ldots, t_r\}$ such as $\forall S_{ci} \in \mathbb{S} \rightarrow \exists T_j \in \mathbb{T}$
begin
for each $S_{ci} \in \mathbb{S}$
        1. Filter $S_{ci}$.
        2. Measure the similarity between the concepts of $S_{ci}$ and the entities of $\mathcal{D}$, and select the best measure.
        3. Construct the relation-based reasoning with the results of (2), the entities of $\mathcal{O}$, and eventually $S_{ci}$, and deduce the corresponding target concepts ($T_j$).
end for
return $\mathbb{T}$;
end
```

5.3.3. Mapping Steps

- <u>Concepts' filtering</u>

The filtering technique is a typical transformation in Information Retrieval, usually used to reduce the size of the text, to simplify the syntactic matching for the retrieval process. The filtering step in the mapping process performs the following actions:

Tokenization: the lexical chain is transformed into simple tokens. A list of delimiters is provided to cut the lexical chain. Example:

<OOC analysis by Techno / Operation> → <OOC> <analysis> <by> <Techno> </> <Operation>

Removal of common words using a list of stop words: stop words (e.g. and, or, by, of, etc.) are usually used for word connection in a lexical chain. We also remove here the symbols and numbers.

Example: *<OOC> <analysis> <by> <Techno> </> <Operation> → <OOC> <analysis> <Techno> <Operation>*

Transformation into lowercase: to unify the case, only the lowercase is used. Example: *<OOC> <analysis> <Techno> <Operation> → <ooc> <analysis> <techno> <operation>*

- String-similarity measuring

A combined similarity measure is used: the *dice coefficient* with the *edit distance*.

The first similarity measure used in the approach relies on calculating the similarity of two sets of strings using the Dice coefficient. With this one, the individual similarity values do not influence the overall similarity of the sets. Thus, the Dice coefficient is used in the approach to compare each set of string inputs (related to resource subjects) with the sets of entries of the dictionary (Algorithm 2). As reminder, given two sets $A = \{a_1, \ldots, a_i\}$ and $B = \{b_1, \ldots, b_j\}$, the Dice coefficient of the two sets is:

$$\mathrm{DiceCoeff}\ (\mathrm{A}, \mathrm{B}) = \frac{2 * (A \cap B)}{A + B}$$

For example: *A = {day, week, month}* and *B = {week, month, year}*. There are two shared concepts in this example. The Dice coefficient would be (2*2/(3+3)) = 0,67. It ranges from 0 to 1.

One particular asset of using this coefficient in our approach is that we use a partial matching technique between the elements of the sets, meaning that two strings are considered similar if they have at least a minimum of similar characters in the same order. For example, *weekly* and *week* are considered similar. Basically, we test if one concept contains the other and vice versa.

Algorithm 2 : Dice Coefficient

```
Input: set of concepts $S_c = \{s_1, \ldots, s_i\}$, set of concepts $E = \{c_1, \ldots, c_k\}$ of the dictionary $\mathcal{D}$
Output: coefficient dice
begin
number ← 0;
for each $s_i$ in $S_c$ do
    for each $c_k$ in $E$ do
        if ($s_i$ contains $c_k$ or $c_k$contains $s_i$) then
            number ← number + 1;
        end if
    end for
end for
dice ←(2* number)/($i$ + $k$);
return dice;
end
```

The second similarity measure used in the approach focuses on the similarity between two strings (instead of two sets). We chose to use the edit distance, and in particular, the Levenshtein distance [Ristad and Yianilos 1998] for its suitability for short strings. In addition, compared to other string-similarity measures, the edit distance emphasizes the syntactic closeness between two strings by only taking the minimum distance. The ratio of similarity based on the edit distance for two strings *a* and *b* is denoted S_{edit} as follows [Zhong et al. 2009]:

$$S_{edit}(a, b) = \frac{1}{1 + editDist(a, b)}$$

where $editDist(a,b)$ denotes the edit quotient between the strings *a* and *b*, which is based on the Levenshtein distance. Thus, the $editDist$ quotient corresponds to the Levenshtein distance (the minimum of edits) divided by the maximum length of the two compared strings.

$$editDist(a,b) = \frac{levenshtein\_dist(a,b)}{\max(length(a), length(b))}$$

For example, consider two strings *day* and *may*, the edit distance would be 0.33, corresponding to one operation (substitution) on the three characters to obtain the same word. Then the corresponding similarity S_{edit} is computed as 1/(1+0.33) = 0.75. S_{edit} ranges from 0 to 1. The standard algorithm of the Levenshtein distance is explained in Appendix H.

The edit distance does not provide an exact matching, a threshold must be then set in order to take relevant similarity results and reduce errors related to weak similarities. We set the threshold to 0.7 after several experimentations.

The final similarity measure used in the approach combines the Dice coefficient with the best edit-distance measure among those that exceed the threshold. The combined formula for the given sets $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_m\}$ is: $\forall i \in \{1, \dots, n\}$, $\forall j \in \{1, \dots, m\}$

$$CSim_{(A,B)} = \frac{2 * (A \cap B)}{A + B} \times \max\left(S_{edit}\left(A(a_i), B(b_j)\right)\right)$$

By combining the similarity measure between sets and between the concepts that constitute these sets we obtain a balanced similarity ratio over the sets. The combined similarity *CSim* is used in the mapping approach to identify the two sets the more similar among the given set of input concepts and the set of entries of the dictionary. The best combined similarity is selected.

For example, given the sets *A = {day, week, month}*, *B = {week, month, year}* and C = {*day, half, tier, quarter*}. We want to know which of the sets *A* and *B* or *A* and *C* are the most similar. Imagine we obtain the following combined similarities: $CSim_{(A,B)} = 0.75$ and $CSim_{(A,C)} = 0{,}35$, then we can conclude that the sets *A* and *B* are more similar than *A* and *C*.

Regarding the mapping approach, *CSim* is used as a matching strategy when there is a minimum of similarity between sets. In fact, to calculate the combined similarity, the S_{edit} must be greater than the threshold and the Dice coefficient must be greater than 0, otherwise, no similarity is considered.

The general algorithm of the similarity matching (Algorithm 3) has as input a given set S_c of concepts (related to the filtered resource subjects in our approach) and the sets of concepts $\mathbb{E}$ of the dictionary, corresponding to the dictionary entries. In fact, each entry here represents a whole description of a concept, including the hypernym, the hyponyms, the synonyms, the variants and the key concepts. All these entities provide concepts. Thus, an entry of the dictionary is considered in the algorithm as a set of concepts E_j = {c_1,..., c_k} where each c_k can be one of the cited entities . The output of the similarity matching is the entry E_j of the dictionary that matched the best with S_c. The algorithm computes the similarity following these steps:

- It first considers that there is a similarity that represents the maximum of the similarities found denoted *max_sim*. This variable is initialized at 0 at the beginning. It also considers a variable *matchSet* which stores the potential set E_j that better match with the set of concepts S_c. These two variables are updated each time a new maximum of similarity is found.

- for each entry E_j of the dictionary, it calculates the Dice coefficient of E_j and the set of strings S_c. The result of this step must be greater than 0 to continue the process.
- thus, if there is a result with the dice coefficient, the algorithm calculates the edit-distance ratio S_{edit} following the defined formula. In fact, because this ratio is applied to strings, the algorithm calculates for each element c_k of the entry E_j its edit distance with the concepts of the set S_c. Each measure is stored in the set *tedits*. However, because many irrelevant measures can be obtained (up to $i * k$ measures where i is the number of elements of S_c and k is the number of elements of E_j), we apply the threshold 0.7 as mentioned before to select the best measures. Thus, only the measures greater than the threshold are memorized. Afterwards, the best measure (max value) is selected to be integrated with the dice coefficient in the combined similarity measure *CSim*. If there is no measure S_{edit} that exceeds the threshold, only the dice coefficient is used as a similarity measure.
- finally, each combined similarity obtained for an entry E_j and S_c is compared with the next similarity obtained for E_{j+1} and S_c. The aim is to identify and memorize the maximum similarity value (*max_sim*) throughout the process. The set E_j corresponding to the maximum value is determined in this way.

At the end of this step, the entry of the dictionary found (in particular the hypernym) enables to make the link with the ontology from which the target concepts will be selected in the next step.

Algorithm 3 : String-similarity matching

```
Input: set of concepts $S_c = \{s_1, \ldots, s_i\}$, sets of entries $\mathbb{E} = \{E_1, \ldots, E_n\}$ of the dictionary $\mathcal{D}$ where $\forall j \in [1 \ldots n]$ $E_j = \{c_1, \ldots, c_k\}$
Output: set $E_j$

begin
//initialization
max_sim ← 0;
matchSet ← ∅;
//combined similarity processing
for each $E_j \in \mathbb{E}$ do
        dice ← Dice_Coefficient($S_c$, $E_j$) ;
        if dice > 0 then
                for each $c_k$ in $E_j$ do
                        for each $s_i$ in $S_c$ do
                                Calculate_$S_{edit}$($s_i$, $c_k$);
                                if $S_{edit} \geq$ threshold then
                                        tedits ← $S_{edit}$;
                                end if
                        end for
                end for
                if number_of_elements(tedits) ≥ 0 then
                        max_Sedit ← Select_Max (tedits);
                        CSim ← Calculate_Combined_Similarity(dice, max_Sedit);
                else
                        CSim ← dice;
                end if
        end if
        if CSim > max_sim then
                max_sim ← CSim;
                matchSet ← $E_j$ ;
        end if
end for
return matchSet;
end
```

- Reasoning with the ontology relations

This step focuses on searching the target concepts that correspond to the description of a resource. A meta-model was proposed in the beginning of this chapter to organize the resource-description types. Our approach focuses on finding the target concepts related to the business and operational levels. The MP ontology is used in this step to identify the target concepts. The PC dictionary used in the previous step enabled to find an entry of the dictionary. This entry makes the transition to the ontology. In fact, each entry in the PC dictionary has one hypernym. As reminder, each concept c of an entry of the dictionary can represent a control indicator or a control method, and the hypernym always represents the control method (or domain) to which belongs a control indicator. Thus, for each entry found in the previous step (in the string-similarity computation), its hypernym is selected, because this one represents a *control method* referenced in the ontology.

At this stage, two scenarios are possible. The first is that the relations between the classes that lead to the target concept are direct and their logic is simple. The second is that there is a complex reasoning to do on the relations in order to find the target concept, meaning that several situations or conditions can be found and we need to determine which condition leads to the target concept. The definition of the relations in the ontology enables to know which scenario can happen. For example, given the concepts *FDC* and *SPC* of the type of concept *control_method*, imagine the target concept to find is a *control objective* type. *FDC* and *SPC* have the following relations with the type *control_objective*:

hasObjective(FDC, equipment_control)

hasObjective(SPC, lot_control)

hasObjective(SPC, equipment_control)

We can see that *FDC* has one objective defined whereas *SPC* has two objectives. The *FD*C case corresponds to the first scenario where there is one and direct relations between a concept and a target concept. *SPC* corresponds to the second scenario where there are some or several target concepts, or where the target concept to find cannot be directly selected with the existing relations. In this case we need some inference rules to deduce the target concept.

The algorithm that performs this step (Algorithm 4) first retrieves the relations related to the hypernym which is also referenced as a concept in the ontology. We will call this concept c. Thus, the algorithm counts the number of relations that use the concept c and that may have in its definition the type of the target concept t_c (or not). If there is at most one relation (between each two concepts related to c) the first scenario is applied, otherwise the second scenario is applied.

Algorithm 4 : target concept search

```
Input: concept $c$ of the ontology $\mathcal{O}$, type of target concept $t_c$, set of input concepts $S_c = \{s_1, \dots, s_i\}$
Output: target concept $t$

let $Rl$ be an empty set of relations
begin
// selection of all the relations related to the concept c
$Rl \leftarrow$ select_relations($c$);
nb ← count_relations($Rl$, $c$, $t_c$);
if (nb≤1) then
        //we apply here scenario 1
        $t \leftarrow$ scenario_1($Rl$, $c$, $t_c$, $S_c$);
else
        //we apply here scenario 2
        $t \leftarrow$ scenario_2($Rl$, $t_c$, $S_c$);
end if
return $t$;
end
```

Scenario 1

The first scenario relies on navigating throughout the direct binary relations between the types of concepts of the ontology, until finding the target. For example, if we take again the concept *FDC* in the MP ontology (Figure 5.3-2), there is a direct relation defined between a control method and a control objective, which is *hasObjective*, thus the control objective will be identified through this direct relation[21].

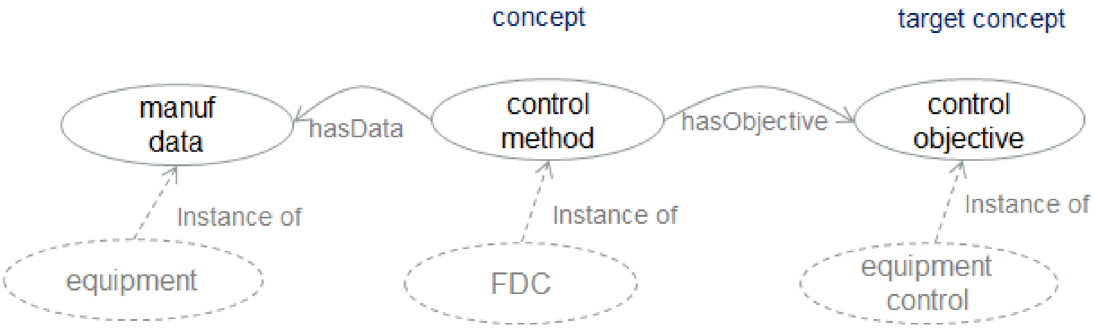


Figure 5.3-2 : Example of the first scenario

Thus, in Algorithm 5, if the type of target concept t_c is in the definition of the relation r of the set Rl (the relations identified in Algorithm 4), then the target concept is directly selected. If there is no target found, the algorithm will search if there are intermediate relations between the concept c and the type of target concept to find and then Algorithm 5 is recalled (or scenario 2 if there are more than one direct type of relations between each two intermediate concepts).

---

[21] There is at least one shared concept between two distinct relations in the MP ontology

Algorithm 5 : scenario_1: search of target concept using one type of direct relations

Input: set Rl that contains a relation r_i (related to the concept c), concept c, type of target concept t_c, set of input concepts $S_c = \{s_1, \ldots, s_i\}$
Output: target concept t

```
begin
for each $r_i$ in $Rl$ do
   if $t_c$ is in the definition of $r_i$ then
         $t \leftarrow$ select_target ($c, r_i$);
   end if
   //we search here if there are intermediate relations that enable to find $t$
   if $t$ is empty then
         $R \leftarrow$ search_intermediate_relations(type_of($c$), $t_c$);
         $ic \leftarrow$ select_intermediate_concept($R$);
         nb $\leftarrow$ count_relations($R$, $ic$, $t_c$);
         if (nb$\leq$1) then
               recall algorithm 5
         else
               $t \leftarrow$ scenario_2($R$, $ic$, $t_c$);
         end if
   end if
   return $t$;
end for
end
```

Scenario 2

The second scenario is supported with a set of rules that enable to infer the target concept. An inference algorithm based on the forward chaining principle is used [Brachman and Levesque 2003]. In this inference technique, the well-known logical rule *Modus Ponens* [Schechter and Enoch 2006] is used. Most logical deductions in knowledge-based systems rely on this standard rule.

The Modus Ponens rule is stated as:

$$\frac{p \Rightarrow q,\ p}{q}$$

Meaning that, if p implies q and p is true, then q is true. The statements or conditional claims that the system reasons on before reaching the conclusion are known as *premises*. In the Modus Ponens rule, we distinguish two premises: $p \Rightarrow q$ and p

For instance:

$\forall x \forall y\ mother(x, y) \Rightarrow olderThan(x, y)$

$mother(Mary, Tom)$

Conclusion: $olderThan(Mary, Tom)$

The Modus Ponens is also applied to statements in the form of *Horn* clauses [Brachman and Levesque 2003]. A Horn clause is an implication that can have several conditions but allows only one conclusion at a time, such as:

$$cond_1 \wedge cond_2 \wedge \ldots \Rightarrow assertion$$

This kind of implication is executed as the *If Then* statement in programming: $IF < condition(s) > then < consequence >$. All the conditions must be satisfied in order to reach the consequence.

Example: $\forall x \forall y \forall z\ mother(x, y) \wedge aunt(z, y) \Rightarrow sister(x, z)$

In the forward chaining, the Modus Ponens rule can be repeatedly applied to statements until reaching the desired goal (conclusion). In fact, starting from the premises, the system applies the rules on a set of facts (like $mother(x, y)$ and $aunt(z, y)$) to produce all the conclusions until a solution is found. This type of inference technique typically needs a fact base and a rule base. The fact base usually contains the true statements which may be used as premises of the problem. The rule base contains the general rules that are used to solve the problem.

The general idea of the forward chaining algorithm is presented in Algorithm 6. It takes as input a fact base F and a rule base SR and returns the fact base modified with eventually new facts. In this algorithm, we use a queue Q to avoid unnecessary loops. Thus, all the facts of F are first added in the queue Q. Each fact q of the queue that is tested with the rule conditions is removed from the queue, so to only keep the facts that are not yet used. The system checks then if a fact q belongs to the condition of a rule r of the rule base. If it belongs and the conditions of r already belong to the fact base but the conclusion does not belong to this base, the system will add this conclusion in the fact base. In this way, new facts will be stored.

Algorithm 6: forward_chaining

```
Input: set of facts $F$, set of rules $SR$
Output: F modified

let Q be a queue that stores facts
begin
Q ← F;
while Q is not empty do
        q ← first(Q);
        Q ← remove (q, Q);  //each time a fact is used, we remove it from the queue
        for each rule r in the rule base SR do
                if (q ∈ conditions (r) and conditions (r) ⊆ F and conclusion (r) ∉ F) then
                        add conclusion(r) at the end of Q;
                end if
        end for
        F ← last(Q);
end while
return F;
end
```

Example:

Suppose we have the following rule in the rule base:

r: $mother(x, y)\ \wedge\ aunt\ (z, y)\ \Rightarrow\ sister(x, z)$

Suppose we have two facts in F (and then in Q):

f1: $mother(Mary, Tom)$

f2: $aunt(Jenny, Tom)$

The first fact q of the queue Q would be $mother(Mary, Tom)$. We can see that q belongs to the conditions of r. The other condition of r which is $aunt(Jenny, Tom)$ also belongs to the facts F but the conclusion of r do not belong to F. Thus, we will obtain $sister(Mary, Jenny)$ that will be added in Q (in order to immediately use it for other rules) and then in F. Regarding the second fact $aunt(Jenny, Tom)$, the system will find that the conclusion already exists in the fact base, then it will not be added.

Regarding our mapping approach, the premises (facts and rules) of our inference algorithm are made with the ontology entities and their relations which constitute the predicates of the premises (e.g. $hasData$, $hasObjective$, ...), and the ontology instances are used in the predicate variables (e.g. *SPC*, *lot*, ...).
For example:

$$control\_method\ (SPC)\ \wedge hasData(SPC, lot) \wedge hasObjective(SPC, lotControl) \\ \Rightarrow control\_objective\ (lotcontrol)$$

Regarding the rules, we use a specific rule base that stores all the necessary rules for the inference algorithm. The forward chaining function is integrated in the scenario 2 (Algorithm 7) of the general mapping algorithm. Also, in scenario 2, we use once again the set of input concepts S_c that were extracted from the resources. We temporarily construct a kind of fact base in a dedicated structure F using the concepts S_c and the entities obtained before (i.e. with Algorithm 4 and scenario 1) which are the set of relations of Rl. The forward chaining function is then called and executed using this structure F and a set of rules SR defined apart, in a dedicated base[22]. At the end of the process, the system searches in the new structure F[23], the concept that eventually corresponds to the target t_c.

Algorithm 7 : scenario_2: search of target concept using inference rules

```
Input: set of relations $Rl$, type of target concept $t_c$, set of input concepts $S_c = \{s_1, \dots, s_n\}$
Output: target concept $t$

let F be an empty structure
let SR be a set of rules
begin
for each $s_i$ in $S_c$ do
        add the concept $s_i$ with its entity type in the structure F;
end for
for each $r_i$ in $Rl$ do
        add the relation $r_i$ with its content (concept and target concept) in the
        structure F;
end for
//call of the forward chaining function
F ← forward_chaining(F,SR);
//selection of the target concept related to the type $t_c$
$t$ ← select_concept(F, $t_c$);
return $t$;
end
```

---

[22] A specific file is used in the implementation where all the necessary rules are defined (cf. chapter 7)

[23] The structure of facts F is created temporarily for the inference algorithm of the mapping process. It is not actually stored

Example:

Consider S_c ={Lot, OOC, analysis} and the type t_c of the target concept t to find is the control objective.

We will have the following facts in F[24], which correspond to the concepts of S_c and the relations Rl:

f1: control_method(SPC)

f2: useTechnique(SPC,OOC)

f3: hasObjective(SPC, lotControl)

f4: hasObjective(SPC, equipmentControl)

f5: data(lot)

f6: hasData(lotControl, lot)

Consider the following rules:

r1: $control\_method\ (x) \wedge hasObjective(x,y) \wedge hasData(y,z) \wedge data(y) \Rightarrow hasData(x,z)$

r2: $control\_method\ (x)\ \wedge hasData(x,y) \wedge hasObjective(x,z) \wedge hasData(z,y) \Rightarrow control\_objective\ (z)$

After processing with the forward chaining, we will obtain:

(r1 with f1, f3, f6, f5) → f7: $hasData(SPC, lot)$

such that:

$$control\_method\ (SPC) \wedge hasObjective(SPC, lotControl) \wedge hasData(lotControl, lot) \wedge data(lot) \Rightarrow hasData(SPC, lot)$$

(r2 with f7) → f8: $control\_objective(lotcontrol)$

such that:
$control\_method\ (SPC) \wedge hasData(SPC, lot) \wedge hasObjective(SPC, lotControl) \wedge hasData(lotControl, lot) \Rightarrow control\_objective\ (lotControl)$

t is then “lot control”.

5.3.4. Example of Synthesis

Many examples from STMicroelectronics helped to develop and ascertain the technical aspects of the approach. Let’s see how works the mapping process by an example dealing with the whole approach. The subjects of a set of MI resources must be selected for this purpose. The types of concepts used as description for the resources and the expected results must also be identified before applying the process.

Consider two MI resources containing the following sets of key words or lexical chain:

$\mathbb{S} = \{S_{c1}, S_{c2}\}$ where:

S_{c1} ={Scraps wafer-fab-yield by module area}

---

[24] Note that the term analysis does not correspond to any entity in the ontology, so no relation related to this concept will be found

S_{c2} ={Control-charts on lots and spec limits}

To simplify the illustration of the semantic mapping process, we consider only the following three types of target concepts (of the semantic-descriptor meta-model): *pc_domain* related to the control methods used (or domains of control), *pc_objective* related to the control objective and *manuf_objective* related to the manufacturing objective.

Then the outputs of the mapping system would be:

$\mathbb{T} = \{T_1, T_2\}$ where each set of targets must have the following target types: t(*pc_domain*), t(pc_objective) and t(manuf_objective).

Figure 5.3-3 shows the inputs and the expected outputs of the mapping process following the example. An overview of the mapping process for one set of concepts is schematized in Appendix C.

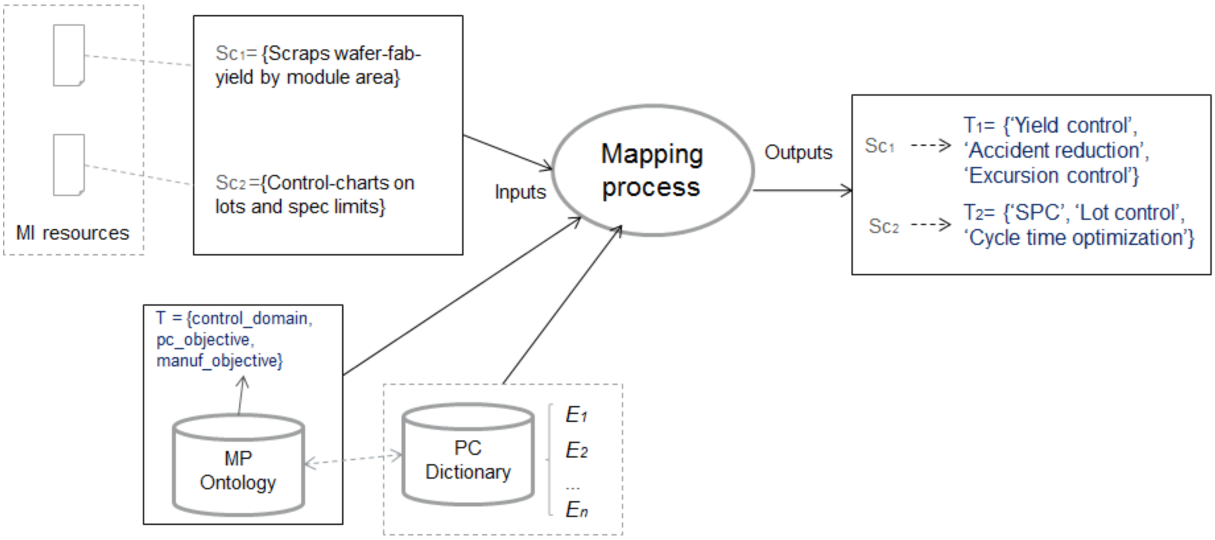

MI resources
Sc1= {Scraps wafer-fab-yield by module area}
Sc2 ={Control-charts on lots and spec limits}
Inputs
Mapping process
Outputs
Sc1 ---> T1= {'Yield control', 'Accident reduction', 'Excursion control'}
Sc2 ---> T2= {'SPC', 'Lot control', 'Cycle time optimization'}
T = {control_domain, pc_objective, manuf_objective}
MP Ontology
PC Dictionary
E1
E2
...
En

Figure 5.3-3 : Inputs and outputs of the mapping process for the example

The first step is to filter the sets S_{c1} and S_{c2}. Thus, we obtain:

S_{c1} ={scraps, wafer, fab, yield, module, area}

S_{c2} ={control, charts, lots, spec, limits}

Afterwards, the system computes the similarity between the filtered sets and all the entries of the dictionary following to the combined-similarity formula CSim. Imagine we have 50 entries, we will then obtain 50 measures. To simplify, consider the following three entries:

$E_1 =$ *{concept('oos'),variant('out of specs'),hypernym('spc'),key-concept('product'), key-concept('target'), key-concept('specification'), key-concept('limit'), key-concept('chart)}*

$E_2 =$ *{concept('war'),variant('wafer at risk'),hypernym('risk control'),key-concept('counter'), key-concept('coverage'), key-concept('limit'), key-concept('down')}*

$E_3 =$ *{concept('wfy'),variant('wafer fab yield'),synonym('production'),hypernym('yield control'),key-concept('scrap'), key-concept('wafer'), key-concept('accident'), key-concept('baseline'), key-concept('ppm')}*

Thus, the similarity measure will be calculated between S_{c1} and E_1, between S_{c1} and E_2, and between S_{c1} and E_3. This process is repeated for each set in $\mathbb{S}$. Figure 5.3-4 illustrates the computation of the similarity measures for each set.

Sets		**Dice coefficient**
S_{c1}	E_1	(2x0)/14 = 0
S_{c1}	E_2	(2x1)/13 = 0.15
S_{c1}	E_3	(2x6)/15 = 0.8
S_{c2}	E_1	(2x4)/13= 0.61
S_{c2}	E_2	(2x2)/12 = 0.33
S_{c2}	E_3	(2x1)/14 = 0.14

Sets		**S_{edit} distances > 0.7**	**Max**
S_{c1}	E_1	-	-
S_{c1}	E_2	1	1
S_{c1}	E_3	0.85, 1, 1, 1, 0.85, 1, 1	1
S_{c2}	E_1	0.83, 0.85, 0.85	0.85
S_{c2}	E_2	0.85	0.85
S_{c2}	E_3	-	-

Combined similarity measure
$CSim(S_{c1}, E_1) = 0$
$CSim(S_{c1}, E_2) = 0.15$
$CSim(S_{c1}, E_3) = 0.8$
$CSim(S_{c2}, E_1) = 0.51$
$CSim(S_{c2}, E_2) = 0.28$
$CSim(S_{c2}, E_3) = 0.14$

Figure 5.3-4 : Computation of the similarity measures

The results of the similarity show that the entry E_3 is the more similar to S_{c1} and the entry E_1 is the more similar to S_{c2}. Therefore, the hypernyms of each entry are systematically considered as the referred concepts representing the domain of interest of each input set. In fact, each hypernym represents a control domain/method in the PC dictionary. According to the dictionary entries in the example, the selected hypernyms are:

hypernym(S_{c1}) = t(*pc_domain*) = {'*yield control*'}

hypernym(S_{c2}) = t(*pc_domain*) = {'*spc*'}

At this stage, the first target concept, i.e. the control domain, is considered identified. Let's search now for the other target concepts using the MP ontology. The hypernyms found makes the link with the ontology concepts, where *yield control* and *spc* are control methods in the ontology.

Regarding the concept *yield control* of the set S_{c1}, we retrieve from the ontology its relations with the first type of target, which is the *control objective*, and afterwards, with the second target i.e. the *manufacturing objective*. The process checks if each type of target is in the definition of the relations of the MP ontology and related to *yield control*. Figure 5.3-5 depicts the relations found.

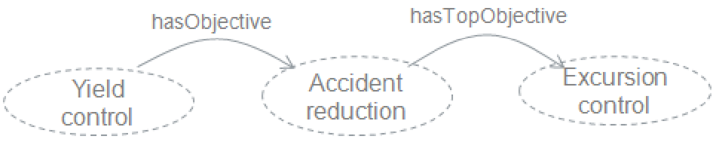


Figure 5.3-5 : Relations of the concept *yield control*

This case corresponds to scenario 1 of the target-concept search in the mapping process. Thus, the target of the relation *hasObjective()* whose concept is *yield control* (i.e. the first part of the relation) will be retrieved.

R: hasObjective(‘*yield control*’, ‘*accident reduction*’)

then t(pc_objective)= {‘*accident reduction*’}

The manufacturing objective is retrieved with the relation type *hasTopObjective()*. However, in the first step of the search, no target will be found (*t is empty*) because there is no relation between the concept y*ield control* and a manufacturing objective. The scenario 1 (Algorithm 5) is then recursively repeated by taking as input the relations related to the intermediate concept found, which is, in this case, the control objective *accident reduction*. The manufacturing objective will be then retrieved.

R: hasTopObjective(‘*accident reduction*’,’*excursion control*’)

then t(manuf_objective)= {‘*excursion control*’}

Regarding the second set S_{c2}, we retrieve the relations relating to the two types of target for the control method *SPC*. Figure 5.3-6 illustrates the relations found.

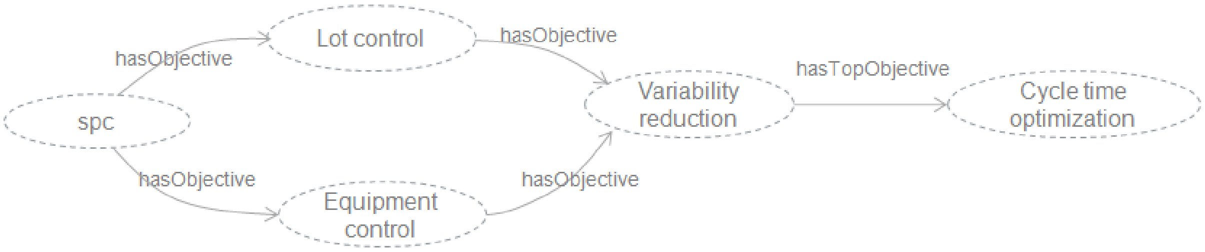


Figure 5.3-6 : Relations of the concept *spc*

This situation corresponds to scenario 2 of the mapping process. We apply then the forward chaining technique to infer the appropriate target concepts. By using these relations and by extracting other relations from the ontology related to the concepts of S_{c2}, we obtain the following facts:

f1: hasObjective(SPC, Lot control)

f2: hasObjective(SPC, equipment control)

f3: hasObjective(Lot control, variability reduction)

f4: hasObjective(equipment control, variability reduction)

f5: hasData(SPC, Lot)

f5: hasData(Lot control, Lot)

f6: hasTopObjective(variability reduction, Cycle time optimization)

If we take again the rule:

r2: $control\_method\ (x)\ \wedge hasData(x,y) \wedge hasObjective(x,z) \wedge hasData(z,y) \Rightarrow control\_objective\ (z)$

we will obtain the following implication:

$$control\_method\ (SPC) \wedge hasData(SPC, lot) \wedge hasObjective(SPC, lotControl) \\ \wedge hasData(lotControl, lot) \Rightarrow control\_objective\ (lotControl)$$

As a result, the control objective and the manufacturing objective would be:

t(pc_objective)= {‘*lot control*’}

t(manuf_objective)= {'*cycle time optimization*'}

Finally, if we compare the obtained results with the expected ones in Figure 5.3-3, we can see that the mapping process leads to the same results, mainly through the PC dictionary, because it represents the link with the MP ontology. Imagine we obtained inappropriate results about the control domain, therefore, all the reasoning done in the MP ontology to find the target concepts may be fallacy. Such findings confirm the importance of the dictionary in rationalizing the terminologies in the approach.

5.4. Conclusion

This chapter presented the semantic mapping techniques and algorithms used for a bottom-up construction of semantic descriptors of resources. These descriptors are intended for resources that already exist in the company and are used as support for the process control. Basically, the creation of the semantic descriptors is carried out in three steps and the mapping mechanism is based on a combined syntactic similarity measure. The mapping is also enhanced with a forward chaining technique that supports the reasoning on the ontology concepts and relations. Furthermore, we can underlie the generic aspect of the algorithms used in this process, which perform the mapping regardless of the used concepts in the inputs and outputs. We have shown with an example from STMicroelectronics that the mapping process can work well as long as the PC dictionary and the MP ontology are consistent. Finally, the mapping techniques were improved progressively by experimenting the approach within STMicroelectronics.

Chapter 6: The Search Patterns

Résumé

Ce chapitre décrit la stratégie de recherche descendante de l'approche S3. Cette stratégie est basée sur des patterns de recherche permettant de placer les besoins métier des utilisateurs au centre du système de recherche. Un méta-modèle de patterns de recherche est proposé. On distingue dans ce méta-modèle deux types de patterns: des patterns requêtes et des patterns d'alignement.

Les patterns de type requête permettent de capturer les besoins des utilisateurs grâce à des mots clés, un but et un contexte lié à l'activité métier de l'utilisateur. La solution à un besoin est construite en créant ou réutilisant des patterns d'alignement conduisant progressivement aux descripteurs de ressources et ainsi aux ressources.

Les patterns d'alignement permettent la capture d'artefacts de besoins métier pour faciliter la satisfaction des besoins des utilisateurs quel que soit leur niveau de complexité (besoins haut niveau ou bas niveau). Ces patterns sont constitués d'un but, d'un contexte et d'une solution, qui peut être à son tour constituée de buts métier ou d'un (de) descripteur(s) sémantique(s). Les patterns d'alignement sont créés et stockés au fur et à mesure que les acteurs du métier effectuent des recherches de ressources d'information manufacturière. Les ressources qui répondent aux besoins des utilisateurs sont ensuite sélectionnées grâce à leurs descripteurs sémantiques identifiés comme solution des patterns d'alignement.

6.1. Introduction

This chapter presents the top-down retrieval approach that relies on a pattern-based search. Two types of search patterns are used: query patterns and alignment patterns. The query patterns are used to assist the expression of the user query. It mainly facilitates the capture of the user queries using a goal and a context. The alignment patterns constitute business need artifacts that support the matching of the user need with the MI resources through a goal refinement mechanism and semantic matching. Thus, we present in this chapter the meta-model of search patterns, we explain their role in the MI-resource retrieval and in the alignment of these resources with the business needs.

6.2. The Meta-model of Search Patterns

In the S3 approach, the search patterns are used in a standard keyword search to contextualize the MI-resource retrieval and enhance the relevance of the search results to the user need. Indeed, the keyword search is the search technique the most intuitive and requested by the users of retrieval systems. However, such a technique could be inefficient because of the huge quantity of results that can be obtained. Moreover, in a business context like the manufacturing process control, the business needs of the end-users are often complex and business-context dependent. Contextualizing the keyword search in this case enables to better target the resources that meet the user need.

Figure 6.2-1 illustrates the meta-model of a search pattern.

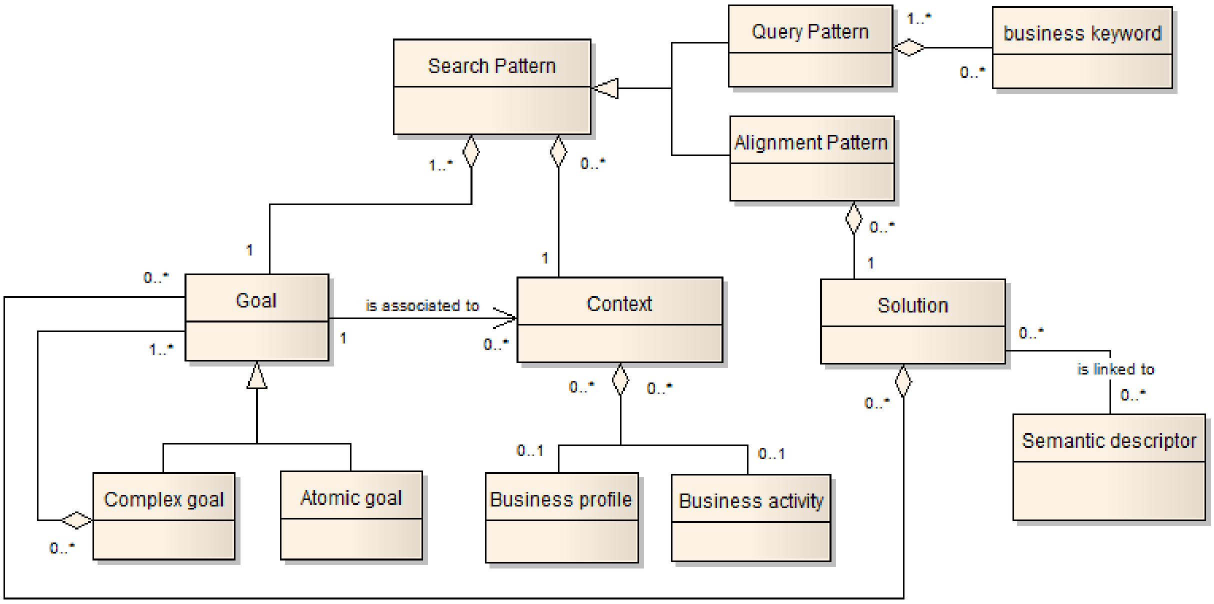
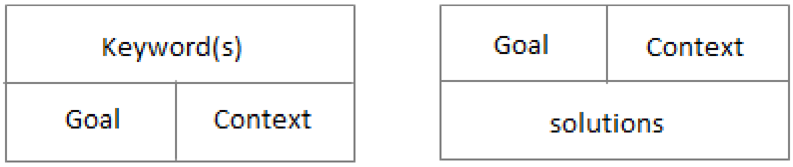


Figure 6.2-1 : The meta-model of the search pattern

We distinguish two types of search patterns: query patterns and alignment patterns. The query patterns capture the user query in a form that can be easily treated by the retrieval system. These patterns comprise a set of business concepts that allow starting the query performing. The alignment patterns capture business needs with their solutions.

A search pattern is generally composed of a ***goal*** and a ***context***, which are common to the two types of patterns. Indeed, the main difference between a query pattern and an alignment pattern consists in the fact that an alignment pattern has a solution whereas a query pattern does not (Figure 6.2-2). The alignment patterns aim at building the solution(s) of a given query pattern by aligning business needs with MI resources. Also, the alignment patterns are persistent and can be reused, comparing to the query patterns which are only created temporarily during the user-need capture.



(a) (b)

Figure 6.2-2 : Notation of search patterns: query pattern (a) versus alignment pattern (b)

The business needs are expressed with **goals**, which can be complex or atomic. A complex goal can be refined into sub goals, whereas an atomic goal cannot be refined. As example, in Figure 6.2-3 *Variability reduction* is a complex goal and *Lot control* is an atomic goal.

The **context** of a need could be the business profile of the user or the business activity inside the manufacturing process. The business profiles and activities are identified in the organization view of the MP ontology. For example, the *process engineering* function is a business profile. The business activities in the semiconductor industry are related to the main business steps of the manufacturing process (e.g. *Photo*, *Etching*).

The **solution** of an alignment pattern can be business goals or one or several semantic descriptors.

A solution can be a set of goals when the alignment pattern is composed of complex goal(s) which must be refined into atomic goals. Figure 6.2-3 shows a general example of an alignment pattern implicitly linked to other patterns. We consider that *Variability reduction* is a complex goal in the corresponding pattern, the solution would be all the sub goals (in the example, *Lot control* and *Equipment control*) that contribute to its realization. The link between a pattern solution and another pattern is not defined in the content of the patterns, but rather deduced during the goal decomposition in the search process.

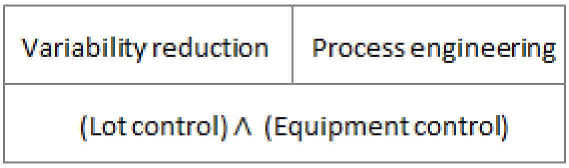


Figure 6.2-3 : Example of a complex goal having two sub goals as solution in an alignment pattern

Hence, for each complex goal cg defined in a pattern, cg can be refined into other complex goals, or into atomic goals or both. Each refinement may vary according to a business context (i.e. business profile or business activity), because a goal is associated to a context. The context leads to the alternatives in the decomposition technique. As a result, we can obtain two types of decompositions according to each business context associated to the goals (either simple or complex), as depicted in Figure 6.2-4.

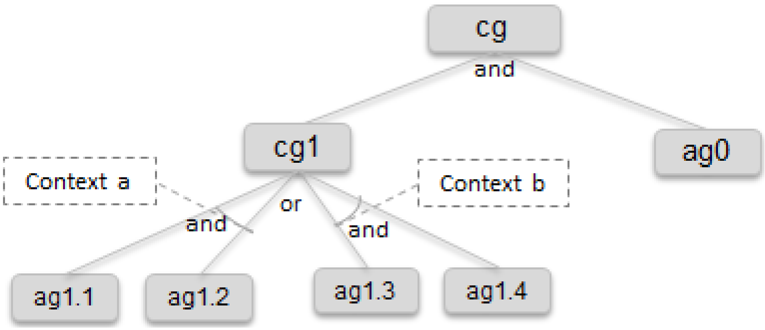


Figure 6.2-4 : Structure of an And/Or decomposition

In this way, the goal decomposition with alternatives takes the form of an AND/OR graph. An AND/OR graph is composed of a set of goals where each goal can be a parent or a sub goal of another goal. A parent goal can have a conjunctive or disjunctive achievement condition. A conjunctive condition (i.e. and), means that a parent goal is achieved when all its sub goals are achieved. A disjunctive condition (i.e. or) is considered as an alternative for the realization of a same parent goal. Goals that cannot be refined constitute the low level of the hierarchical structure of goals and are known as atomic goals.

In the S3 approach, the goal decomposition is identified according to the used goals in the manufacturing process ontology. In this ontology, a complex goal can be a manufacturing objective or a control objective. If we take the example of the control objective *variability_reduction* in Figure 6.2-5, this one can be refined in two ways depending on the business profile *process_engineering* or *defectivity*. In the MP ontology, the relations between all its views enable to identify the relation between a goal and the context specified by the user in the query pattern.

Variability reduction	Process engineering
(Lot control) ∧ (Equipment control)	

Variability reduction	Defectivity
(Equipment control) ∧ (Defect control)	

Figure 6.2-5 : Example of a complex goal having two alternatives of goal decomposition in alignment patterns

When an alignment pattern has an atomic goal, its solution is composed of the semantic descriptors that satisfy the atomic goals according to the specified context. The semantic descriptors refer here to the resources that meet the atomic goals of the alignment patterns. In Figure 6.2-6, the semantic descriptors SD1, SD2 and SD3 were identified because they match with the pattern description.

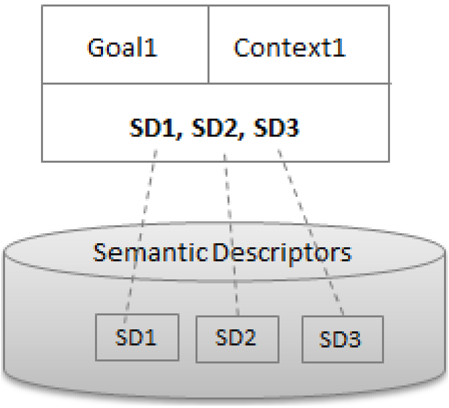


Figure 6.2-6 : An alignment pattern linked to semantic descriptors

To sum up, two types of solutions are considered in a search pattern of alignment type: goals and semantic descriptors. In case of a complex goal in a pattern, other patterns are involved to refine this goal, otherwise, semantic descriptors are involved.

6.3. Role of Search Patterns in Resource Description and Retrieval

With the advent of semantic search techniques, several approaches use the notion of context to increase the relevance of the search results to the user need. The notion of context generally refers to any knowledge that includes the user characteristics in querying [Hubert 2010]. As examples of characteristics related to the user, we can cite its profile, its domains of interest and the intention or the goal behind his need. In information-retrieval research works, other aspects are more and more considered, such as the business domain of the information needed (using ontologies) and the structure of the resources that carry information. In component retrieval approaches, software aspects are also considered, mainly related to the implementation environment of the components. On the whole, the key commonality between the retrieval approaches and systems that uses the notion of context focuses on putting the user in the heart of the search process. Indeed, the user can be included in different ways in a search process, such as:

- in the query expression: the user can be assisted in expressing his query by specifying his context of search
- during the processing of the query: this processing can take into account the user characteristics to filter the results throughout the search process
- in the visualization of the search results: the search results can be categorized according to the context of search of the user
- with resource indexation: known as semantic indexation, the resources will be indexed according to some user characteristics, to speed up the query processing during resource retrieval

We can notice that neither approach uses alignment aspects to contextualize the search process, in particular that the user need can be business-context dependent as in our case study. Indeed, the S3 approach has the ability to contextualize the users' needs in a way to align them with the resources using goal-oriented mechanisms. The top-down retrieval strategy combines a standard keyword search with a goal-based search in order to include the business needs of the users in the resource retrieval process. The pattern-based search aims then at capturing complex needs of business actors using query patterns and supports their satisfaction using alignment patterns.

Figure 6.3-1 illustrates a general architecture resulting from a pattern-based search. When the user expresses his query, it is translated after processing into a goal structure where the goal of the query pattern is refined until atomic goals according to a given context (if any). This context is identified in our approach according to the business profile or activity specified by the user in the query pattern. Each atomic goal with a context can be referred to in one or several semantic descriptors, those that are associated to the resources that address the user need.

At last, the search patterns provide kinds of business need artifacts that can be reused for same business needs. By using these artifacts, whether with a keyword search or without, the system can better identify the resources relevant to the user need and to his context of work.

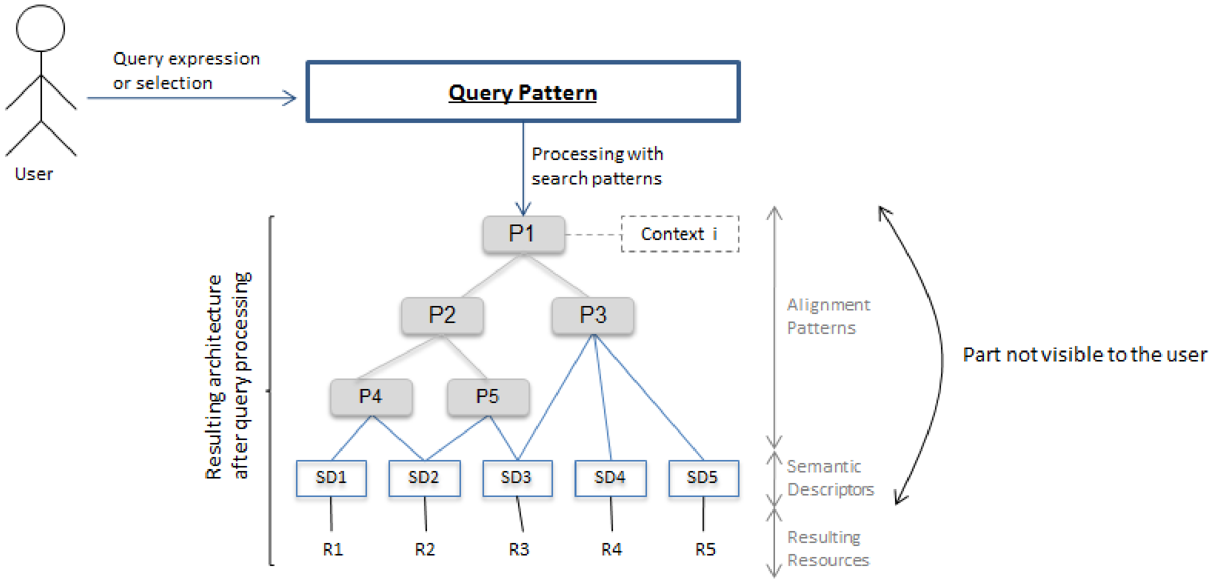


Figure 6.3-1: Role of the search patterns in aligning MI resources with business needs

Every alignment pattern provides the ability to add a business description to a resource related to the context of work of the user. As a result, the pattern-based search enables to add another level of resource description, close to business needs. The semantic alignment between the user need and the resources is achieved in this way.

Figure 6.3-2 shows the general levels obtained with the S3 approach. The retrieval strategies of the S3 approach (i.e. the top down and bottom up) are complementary. By using the bottom-up technique and the top-down one, we in fact achieve a resource description aligned with the business needs.

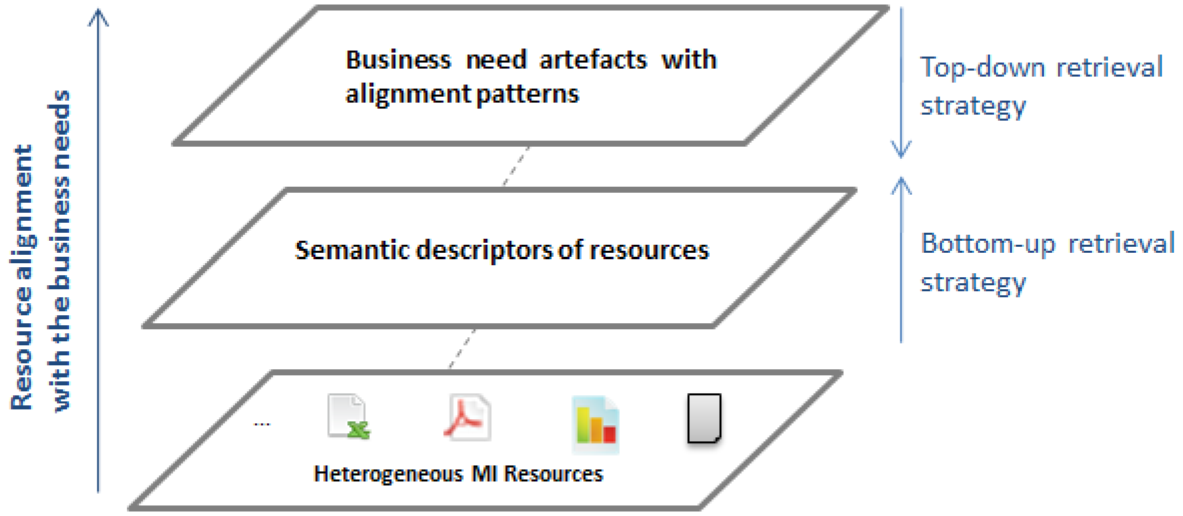

Resource alignment with the business needs
Business need artefacts with alignment patterns
Semantic descriptors of resources
...
Heterogeneous MI Resources
Top-down retrieval strategy
Bottom-up retrieval strategy

Figure 6.3-2 : The resulting levels of resource description

Finally, with respect to our purpose of resource description and retrieval, we choose a pattern-based search in the S3 approach for the following reasons:

- the search patterns contextualize the response to a user need by including his context of work (i.e. business goals, activities) in the search process
- they are suitable to any type of needs: Regardless of the level of abstraction of a captured business need, the system can, in any case, find the corresponding resources to this need
- the alignment patterns provide a modular expression and operationalization of the business needs. This modular aspect enables to better suit future needs and changes
- they foster the reuse of same solutions for same types of needs, enabling then to capitalize on business knowhow, as -in our case- for the control of the manufacturing processes

6.4. The Pattern-based Search

6.4.1. Overview

The main aim of the pattern-based search is to put the business needs of the end users in the heart of the resource retrieval process. The users formulate their query with the help of query patterns, and the system performs these queries and enriches at the same time the MI-resource description through the alignment patterns. In fact, when the user formulates his query with a query pattern, two situations, illustrated in Figure 6.4-1, can happen:

a) the alignment pattern(s) do(es) not yet exist, the system will build pattern-based solutions starting from the user query

b) the alignment patterns that meet the user need exist, the system will then take the solutions of the patterns that better meet the user query

In the first case (a), the system creates the corresponding alignment patterns basing on the goal and the context specified by the user in the query pattern. The resulting patterns are stored for reuse. Moreover, because the query of the user could contain keywords related to the resources (i.e. a process control description), a matching is applied to find correspondences with the dictionary and the semantic descriptors.

In the second case (b), the system searches in the solutions of the alignment patterns the ones that lead to the relevant resources to the user need. These patterns with their solutions are reused in several queries to guide the refinement of the user need until reaching the final solution.

N.B: *In any case, when some patterns already exist and others must be created, the created ones are implicitly linked to the existing patterns following the goal decomposition relations.*

At the end of the process, the results obtained with the atomic goals of the alignment patterns and with the keyword search are merged, filtered and ranked, so to only keep the corresponding resources to the user need. The ranking is also applied in order to display the results in a suitable way to the user.

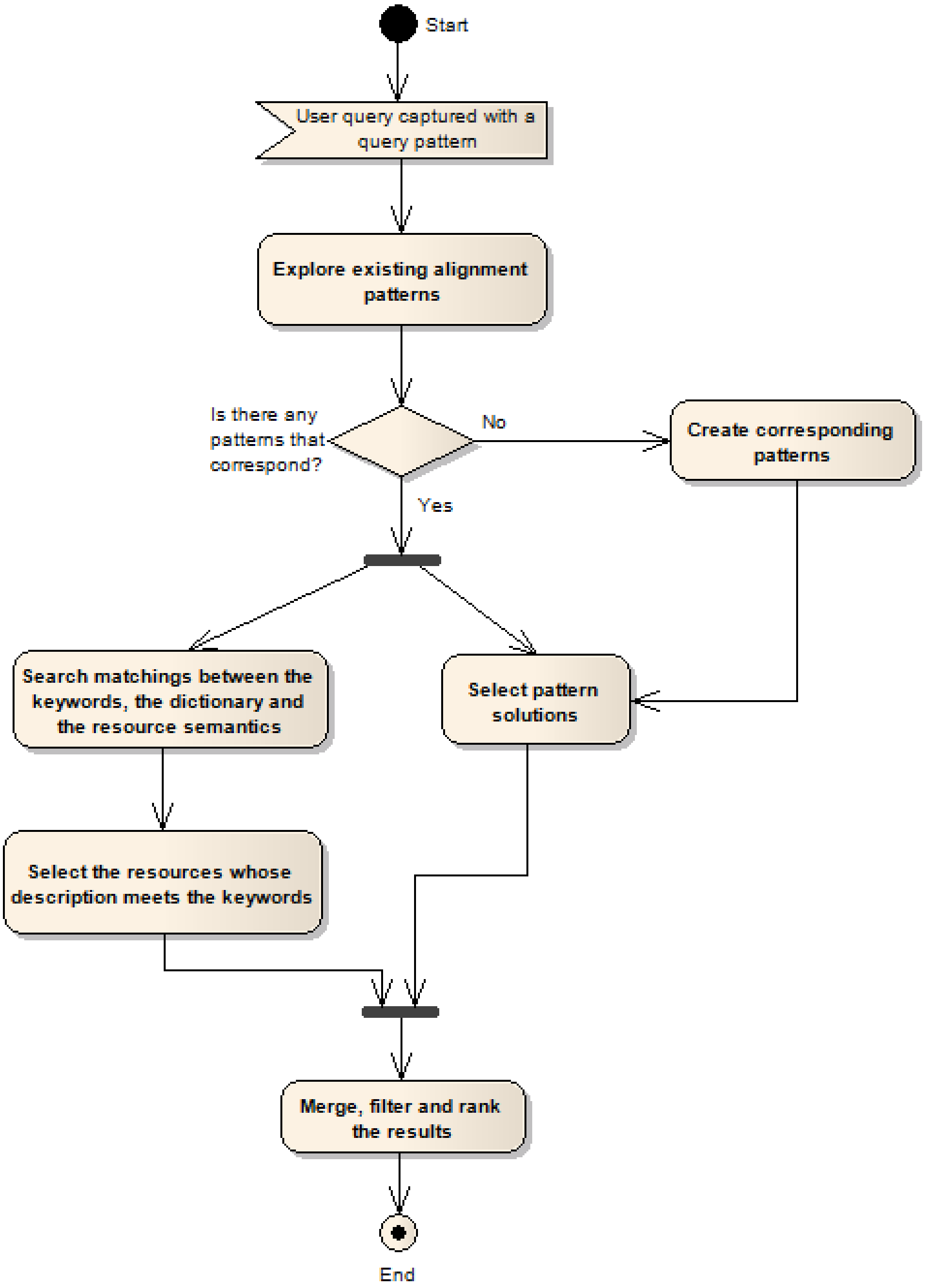

Start
User query captured with a query pattern
Explore existing alignment patterns
Is there any patterns that correspond?
No
Create corresponding patterns
Yes
Search matchings between the keywords, the dictionary and the resource semantics
Select pattern solutions
Select the resources whose description meets the keywords
Merge, filter and rank the results
End

Figure 6.4-1 : Overview of the pattern-based search process (UML activity diagram notation)

6.4.2. Creation of the Alignment Patterns

6.4.2.1. From a query pattern to alignment patterns

Each time a user formulates a query, a set of alignment patterns are created or can be created from this query pattern. Figure 6.4-2 shows how a query pattern is translated into an alignment pattern. The goal and context specified by the user in the query pattern enables starting the creation of the necessary alignment patterns for the query treatment. The captured goal G1 with the context C1 will be respectively the goal and context of a new alignment pattern (if it does not exist in the pattern base). The manufacturing process ontology and the semantic descriptors of the MI resources are used to find the corresponding descriptions according to the defined relations between concepts.

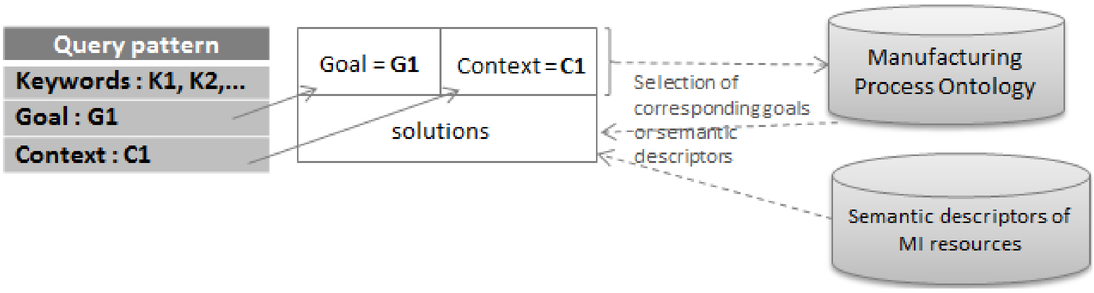

Query pattern
Keywords : K1, K2,...
Goal : G1
Context : C1
Goal = G1
Context = C1
solutions
Selection of corresponding goals or semantic descriptors
Manufacturing Process Ontology
Semantic descriptors of MI resources

Figure 6.4-2 : Creation of an alignment pattern starting from the query pattern

Goal-oriented mechanisms are used to build the alignment patterns starting from the first one captured with a query pattern. Three mechanisms are used: **goal decomposition**, **goal-sibling decomposition** and **goal abstraction**.

During the creation of the alignment patterns, each goal identified with its direct sub goals (which can be complex or atomic) in a business context is considered as a business need artifact and is stored in an alignment pattern. The atomic goals of each alignment pattern are linked to the semantic descriptors, making in this way the link with the resources.

6.4.2.2. Goal-oriented mechanisms

Goal-oriented approaches have proved their ability to efficiently support the expression of business requirements in software engineering. A goal is generally viewed as an objective or a task the system under consideration should achieve [Lamsweerde 2001]. A goal may be formulated at different levels of abstraction so to cover different types of concerns (business needs, software needs, high-level needs, low-level needs, etc.). Goal-oriented mechanisms are usually used to drive the capture and the structuring of the goals that a system should achieve.

In the context of MI-resource retrieval, the informational need of a user can be complex, especially that it is related to the business needs of the company. In this case, it is worthwhile to use goal-oriented mechanisms to capture the business needs of the end users that led them to searching for MI resources, because the expression of goals at different levels here enable to fill the gap between the high-level needs of the users and the MI resources.

In the S3 approach, three goal-oriented mechanisms are used to capture and encapsulate business-need artifacts in alignment patterns: goal decomposition, goal-sibling decomposition and goal abstraction. These mechanisms are complementary and are used repeatedly as much time as they are needed for the construction of the necessary alignment patterns related to a given high-level business need.

- Goal Decomposition:

The goal decomposition is expressed when, considering a complex goal CG, all the sub-goals of CG must be achieved in order to achieve this goal (cf. section 6.2). Also, CG can have alternatives of decomposition according to a given context. Figure 6.4-3 depicts an example of a goal decomposition with two alternatives. *Variability reduction* is a goal that can be decomposed into *Lot control* and *Equipment control* in the business context *process engineering*, or into *Defect control* and *Equipment control* in the business context *Defectivity*.

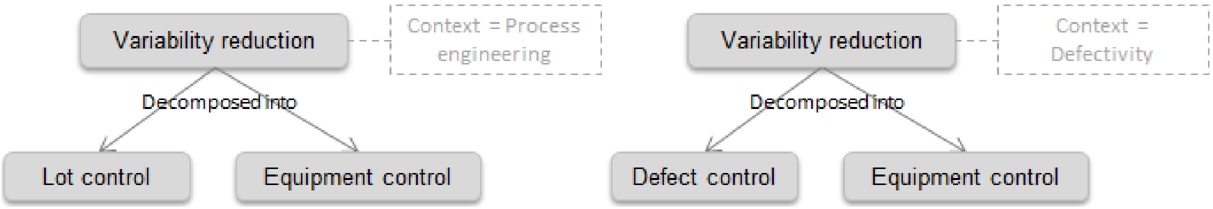


Figure 6.4-3 : Illustration of goal decomposition with alternatives

Three types of decompositions can be identified in a same context c, as follows:

- a complex goal CG can be refined into complex sub goals SG. Each sub goal SG_i must be refined in turn until achieving atomic goals

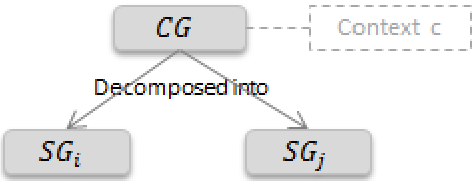


Figure 6.4-4 : Decomposition of a complex goal into other complex goals

- a complex goal CG can be refined into complex goals SG and into atomic goals AG. The sub goals that are complex must be refined in turn until achieving atomic goals

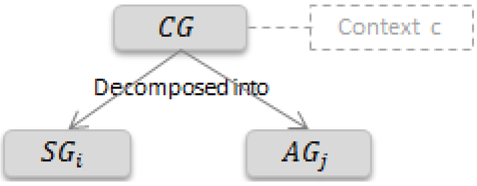


Figure 6.4-5 : Decomposition of a complex goal into complex goals and atomic goals

- a complex goal SG can be refined into atomic goals AG only. The refinement in this case is achieved

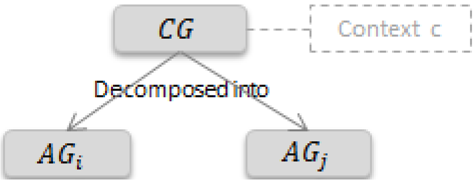


Figure 6.4-6 : Decomposition of a complex goal into atomic goals

Regarding the alignment-patterns' creation, a decompose-goal function is defined to create alignment patterns starting from a goal CG and a context c.

Function:	***decompose-goal***
Input:	complex-Goal CG, context c, ontology $\mathcal{O}$
Output:	alignment pattern P_i, such that:

	$P_i = \begin{cases} \boldsymbol{Goal} = \ CG \\ \boldsymbol{Context} = c \\ \boldsymbol{Solution} = \ \{G_n\} \ where\ \forall n, G_n\ subgoal\ of\ CG\ in\ \mathcal{O} \end{cases}$
Description:	This function identifies the alignment patterns used as solution to refine the goal of a given pattern. The MP ontology (denoted $\mathcal{O}$) is used to identify the sub goals of the complex goal CG associated to the context c. Thus, a pattern P_i will be created as output with the goal CG, having c as context. The solution to this pattern would be each goal G_n identified in the ontology as sub goal of CG in the same context c.

- Goal-Sibling Decomposition:

The goal-sibling decomposition is a special case of the goal decomposition. The decomposition of goals is identified starting from a sibling goal. Considering two goals *CG1.1* and *CG1.2* (Figure 6.4-7), if *CG1.1* is sibling of *CG1.2*, and *CG1.2* is used in the same context as *CG1.1*, thus *CG1.2* will be also captured in an alignment pattern and decomposed in turn if it is a complex goal.

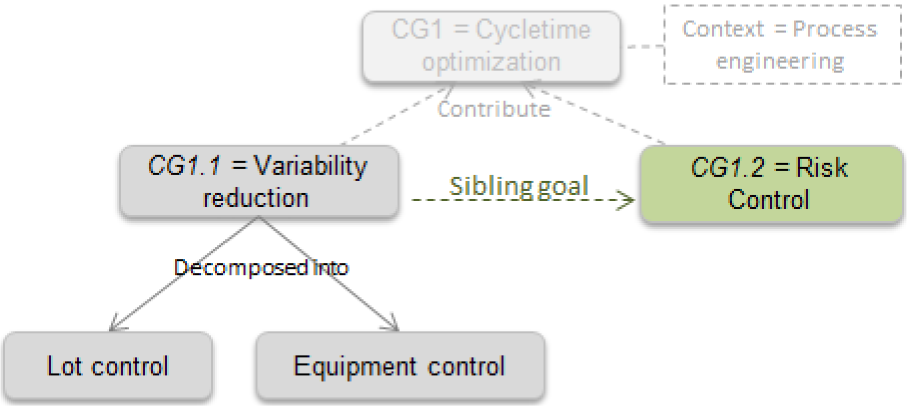

CG1 = Cycletime optimization
Context = Process engineering
Contribute
CG1.1 = Variability reduction
Sibling goal
CG1.2 = Risk Control
Decomposed into
Lot control
Equipment control

Figure 6.4-7 : Illustration of goal-sibling decomposition

Each sibling goal identified can be complex or atomic. In case of complex goals, the goal decomposition types mentioned before are the same for the sibling-goal decomposition.

Regarding the function for pattern creation, the *decompose-goal* function is also used here to refine the complex goals of each sibling goal of CG_n. The sibling goals of CG_n must be first identified before applying the *decompose-goal* function.

- Goal Abstraction:

The goal abstraction is expressed when, considering two goals *SG1.1* and *SG1.2*, they contribute to the realization of a goal *CG1*. The abstraction of goals can also be associated to a given context c. In Figure 6.4-8, the goals *Cycle time optimization* and *Procedure optimization* contribute to the realization of the same goal *Cost optimization* in the same context. Thus the goal *CG1* will be considered as a complex goal in a new alignment pattern with *SG1.1* and *SG1.2* as solutions.

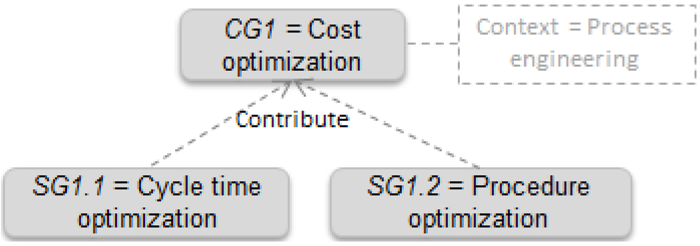

CG1 = Cost optimization
Context = Process engineering
Contribute
SG1.1 = Cycle time optimization
SG1.2 = Procedure optimization

Figure 6.4-8 : Example of a goal abstraction

The function defined for the abstraction mechanism takes as input a goal SG_k which was identified as sub goal through its relations in the ontology, a context c and the MP ontology (denoted $\mathcal{O}$). The created alignment pattern in this case necessarily contains the goal SG_k used as input.

Function:	***abstract-goal***
Input:	sub-goal SG_k, context c, ontology $\mathcal{O}$
Output:	alignment pattern P_i, such that: $P_i = \begin{cases} \textbf{Goal} = G \textit{ parent goal of } SG_k \\ \textbf{Context} = c \\ \textbf{Solution} = \{SG_k\} \cup \{SG_n\} \textit{ where } \forall n, SG_n \textit{ subgoal of } G \textit{ in } \mathcal{O} \end{cases}$
Description:	This function identifies the alignment patterns whose solution is used in the definition of other patterns. Using the defined relations between concepts in the ontology $\mathcal{O}$, the parent goal G of SG_k will be identified, as well as its sibling goals that also contribute to the realization of the goal G. As a result, starting from the goal SG_k and using the context c, a pattern P_i will be created, having G as goal and c as context. The solution to this given pattern is all the identified sub goals SG_k and SG_n that contribute to the realization of G.

6.4.2.3. Algorithm of patterns' creation:

Algorithm 8 sketches the general steps for the creation of alignment patterns starting from a goal cg and a context c. The functions related to the goal-oriented mechanisms are called depending on the type of goal cg used as input. If the goal cg is a complex goal, the function *decompose-goal()* is applied. If the goal cg is a sub goal of another goal g, the function *abstract-goal()* is applied. If the goal cg has sibling goals in the context c (identified through the goal abstraction mechanism), the function *decompose-goal()* is used with, as input, each sibling goal cg' of the goal cg. The algorithm is applied recursively as many times as there are complex goals in the solutions of cg or in the solutions of their sibling goals cg'.

When cg is an atomic goal, the semantic descriptors SD_n that match with the goal cg and the context c are then identified as solution (note that the semantic descriptors are not stored in the pattern base, only their identifiers are used to refer to them).

Algorithm 8 : Creation of alignment patterns

```
Input: goal $cg$, context $c$, ontology $\mathcal{O}$,  Pattern Base $PBase$, semantic-descriptors' base $SDBase$
Output: $PBase$ updated
/*A pattern in $PBase$ has the form: $P$[goal, context, solution]*/
begin
if ($cg$ with $c$) do not belong to any pattern of $PBase$ then
    if ($cg$ is a complex goal) then
        $P_i[cg, c, S] \leftarrow$ decompose-goal($cg, c, \mathcal{O}, SDBase$);
        Store $P_i$ in $PBase$;
        Recall algorithm 8 while each $sg$ in the solution S of $P_i$ is a complex goal;
    else
        Select the semantic descriptors $SD_n$ related to $cg$ and $c$;
        Create $P_i$ with the goal cg, the context c and the solutions $SD_n$;
    end if
    if $cg$ is a sub goal of a goal $g$ in $\mathcal{O}$ and ($g$ with c) do not belong to $PBase$ then
        $P_j[g, c, S] \leftarrow$ abstract-goal($cg, c, \mathcal{O}$);
        Store $P_j$ in $PBase$;
        /*the solution of the pattern created with the abstract function contains the sibling goals $cg'$ of $cg$ */
        Select the sibling goals $cg'$ of $cg$ according to $P_j$;
        for all $cg'_k$ in $P_j$ do
        if ($cg'$ with c) do not belong to $PBase$ then
            $P_k[cg', c, S] \leftarrow$ decompose-goal ($cg', c, \mathcal{O}, SDBase$);
            Store $P_k$ in $PBase$;
            Recall algorithm 8 while each $sg'$ in the solution S of $P_k$ is a complex goal;
        end if
    end if
end if
return;
end
```

6.4.2.4. General illustration

To illustrate the creation of alignment patterns, we present in Figure 6.4-9 a general example of patterns created with the three goal-oriented mechanisms.

We consider in the example that the first alignment pattern created from the query pattern is the pattern P1 which has G1.2 as goal and C1 as context. By using the goal decomposition, G1.2 is first decomposed into G1.2.2 and G1.2.3. These two goals are found in the MP ontology through the relations related to the manufacturing objectives and the control objectives. These relations enable to identify the required goals and contexts for the creation of the alignment patterns. As reminder, two relations are used in the MP ontology to link the goals: *hasObjective()* and *hasTopObjective()*. The relation *hasObjective()* is used between two control objectives. The relation *hasTopObjective()* is used between a control objective and a manufacturing objective and between two manufacturing objectives. We have also the relation *hasOrg()* between a business activity or profile and an objective (either manufacturing or control). By using these relations, we deduce the links between the alignment patterns during their creation. In the example of Figure 6.4-9, all the goals related to the goal G1.2 will be selected for its decomposition and/or its abstraction. We consider that we found the following relations:

- *hasTopObjective(G1.2, G1)*: means that the goal G1.2 has as parent goal G1
- *hasTopObjective(G1.1, G1)*: means that the goal G1.1 has as parent goal G1
- *hasObjective(G1.2.2, G1.2)*: means that the goal G1.2.2 contributes to the realization of the goal G1.2, or G1.2 has G1.2.2 as sub goal.
- *hasObjective(G1.2.3, G1.2)*: means that the goal G1.2.3 contributes to the realization of the goal G1.2, or G1.2 has G1.2.3 as sub goal.
- *hasOrg (G1.2.2, C1)*: means that the goal G1.2.2 is used in the context C1
- *hasOrg (G1.2.3, C1)*: means that the goal G1.2.3 is used in the context C1
- *hasOrg (G1.2.3, C2)*: means that the goal G1.2.3 is used in the context C2
- *hasOrg (G1.2, C1)*: means that the goal G1.2 is used in the context C1
- *hasOrg (G1.1, C1)*: means that the goal G1.1 is used in the context C1
- etc.

Knowing that:

- the goal G1 is a manufacturing objective
- the goals G1.1, G1.2, G1.2.2, G1.2.3 are control objectives (according to the MP ontology)

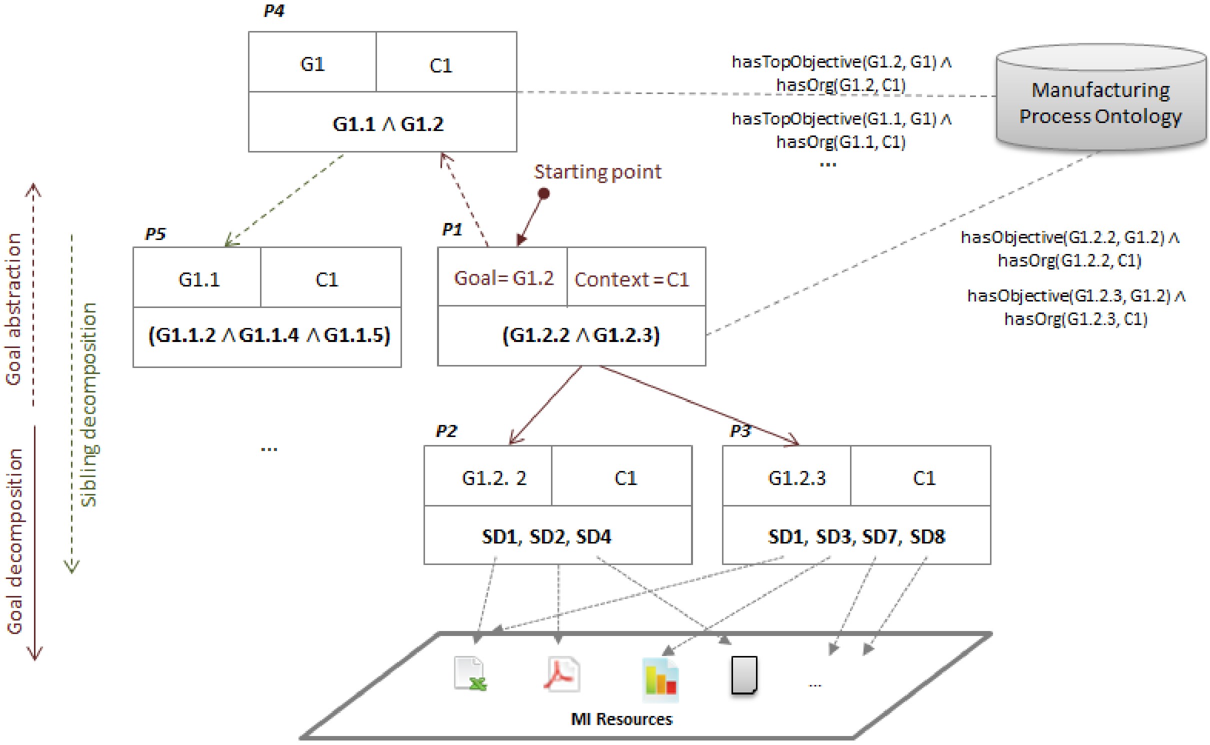


Figure 6.4-9 : General example of creation of alignment patterns

As we can notice in the ontology, the goal G1.2 has the goals G1.2.2 and G1.2.3 as sub goals and all are used in the same context C1. Thus, by using the decompose-goal function, the patterns P1, P2 and P3 will be identified and created.

Regarding the goal abstraction, the abstract-goal function is applied to all the goals to which the goal G1.2 contributes. In the example, G1.2 contributes to the realization of G1 in the context C1. Thus the pattern P4 related to G1 and C1 will be created and all the goals that refine it according to the context C1 will be selected as solution of this pattern (in the example G1 is a root goal because there are no relations in the ontology that infer that it has a parent goal).

Regarding the sibling decomposition, the decompose-goal function is applied to all the sibling goals of the goal G1.2 if they are also associated to the same context C1. Also, the sub goals of the sibling goals will be also selected if they are associated to the same context C1, until achieving the semantic descriptors. In the example of Figure 6.4-9, the goal G1.1 is a sibling goal of G1.2 because G1.1 and G1.2 contributes to the realization of the goal G1 (according to the MP ontology). Thus, the pattern P5 will be created (and successively, all the patterns linked to the solutions of P5).

Finally, the semantic descriptors are identified according to the control objectives and the control methods to which these descriptors are related. The atomic goal G1.2.2 of the alignment pattern P2 must match with the control objectives of the semantic descriptors, and the control methods related to the semantic descriptors must be used in the same context as the goal G1.2.2 (Figure 6.4-10). The relation *hasOrg()* of the MP ontology is used once again in this level. These two correspondences enables to identify the candidate semantic descriptors of resources for each given alignment pattern.

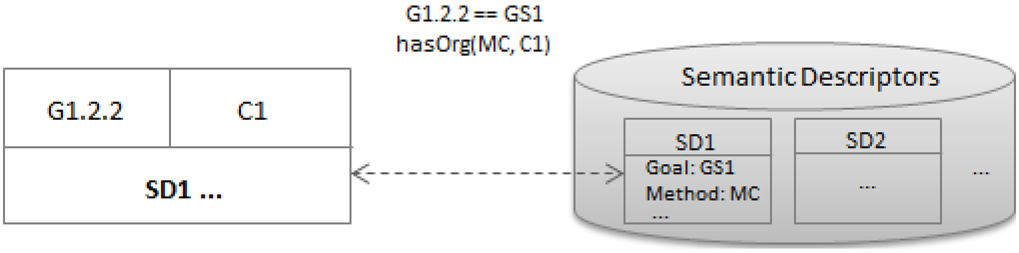


Figure 6.4-10 : Relation between an alignment pattern and a semantic descriptor

N.B: *Case of alternatives of decomposition:*

If we consider that the goal G1.2 can be also used in the context C2, and that the user specifies the context C2 for the goal G1.2 in another query pattern, Figure 6.4-11 depicts some of the resulting alignment patterns related to this case. We can denote that the goal G1.2 is captured one more time but for the context C2, because this goal is associated to two contexts in the MP ontology. As a result, two patterns are created with the same goal G1.2: P1 in Figure 6.4-9 and P6 in Figure 6.4-11. The resulting architecture infers that there are two alternatives for achieving the complex goal G1.2, because we have two decompositions of G1.2 depending on the context C1 or C2.

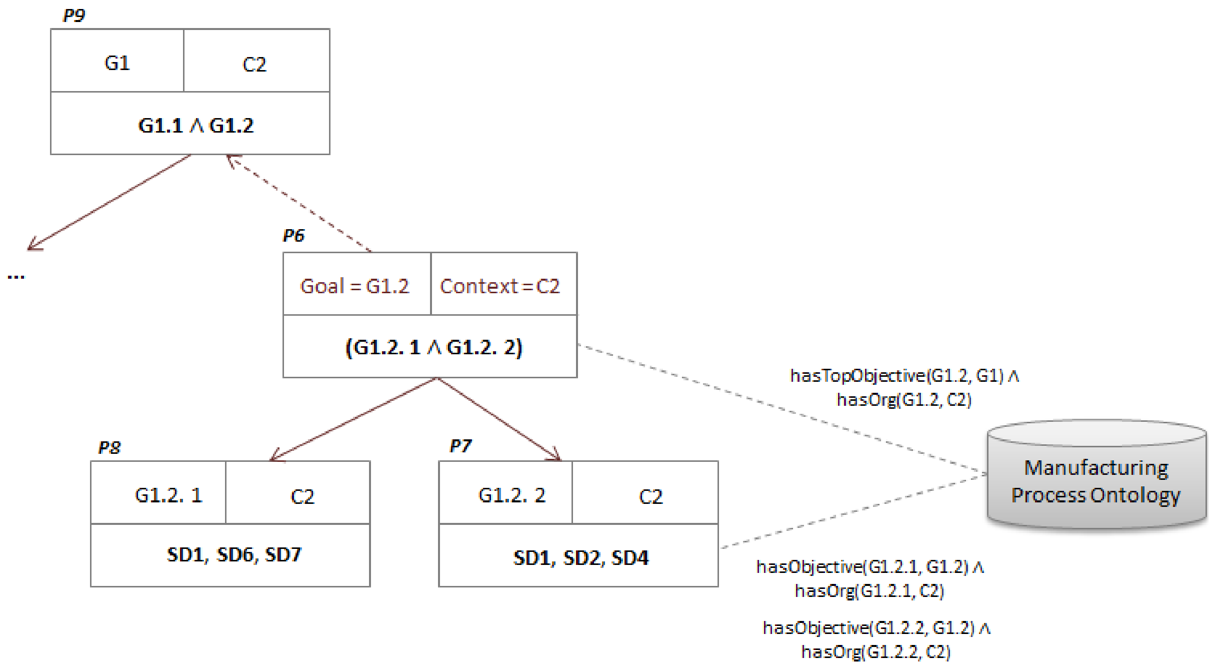


Figure 6.4-11 : Examples of alignment patterns related to the goal G1.2 and the context C2

6.4.2.5. Resulting pattern base

In summary, the creation of the alignment patterns is supported with three goal-oriented mechanisms: goal decomposition, sibling-goal decomposition and goal abstraction. The resulting patterns are stored in a persistent base, which enable to capitalize on business need solutions. The alignment patterns can then be reused to respond to recurrent business needs (Figure 6.4-12). The pattern base is progressively enriched with other alignment patterns created each time there are new needs expressed by the users with query patterns.

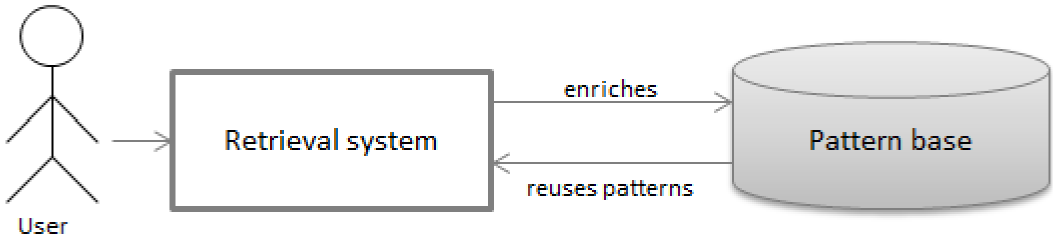


Figure 6.4-12 : Relation between the retrieval system and the pattern base

As a result, four types of situations related to the relations between patterns can be obtained at a given time in the pattern base:

Situation 1:

A goal (G1.1) of a pattern solution can be the goal of one or several other patterns regardless of the involved contexts.

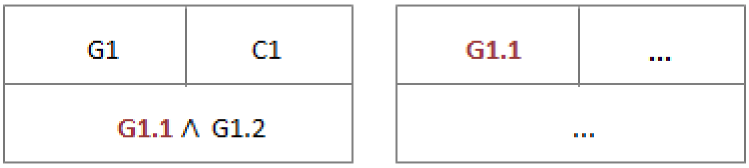


Figure 6.4-13 : Illustration of situation 1

Situation 2:

Two patterns can have the same goal (G1) in their definition and the same goals (G1.1 and G1.2) as solution but in two different contexts (c1 and c2). The patterns used as solution in this case are not the same for each context. In fact, the relation between the patterns is created dynamically when a goal of a pattern solution needs to be decomposed according to the context c1 or c2. (Example: P4 in Figure 6.4-9 and P9 in Figure 6.4-11)

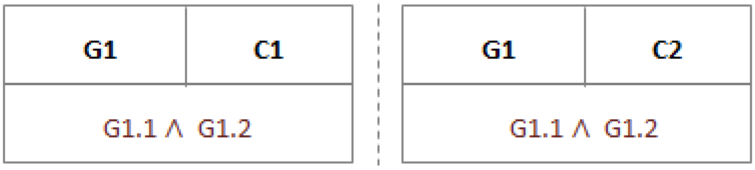


Figure 6.4-14 : Illustration of situation 2

Situation 3:

Two patterns can have the same goal in their definition (G1) but is decomposed differently according to different contexts. In the example, the first pattern has G1 as goal in a context c1 and has as solution G1.1 and G1.2. The second pattern has the same goal G1 in a context c2 but has another solution (G1.2 and G1.3). The two patterns can by the way share the same goal(s) in their solution (i.e. G1.2 in the example).

(Example: P1 in Figure 6.4-9 and P6 in Figure 6.4-11)

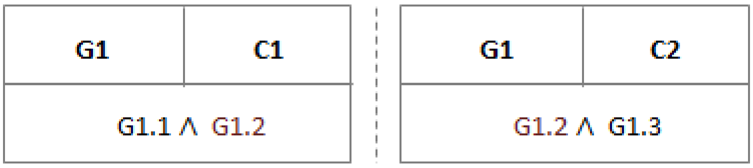


Figure 6.4-15 : Illustration of situation 3

Situation 4:

When a pattern has semantic descriptors in its solution, the latters can be in the solution of other alignment patterns.

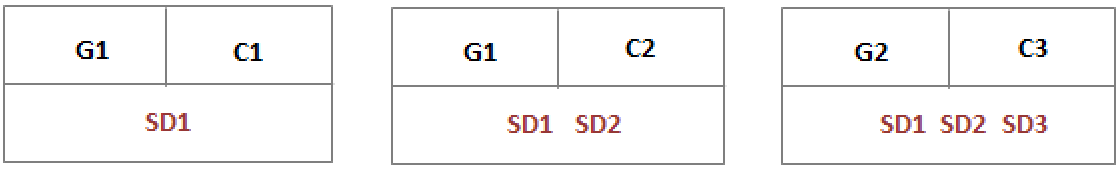


Figure 6.4-16 : Illustration of situation 4

6.4.3. Keyword Matching and Results' Ranking

We presented in the overview of the pattern-based search the general steps of the search process which forks during the query treatment to create alignment patterns. Afterwards, the system performs a standard keyword search using the keywords specified by the user in the query pattern. The results obtained here are merged with the results obtained with the alignment patterns, and are filtered and ranked in order to be displayed to the user. We present in the following these last steps of the pattern-based search. Figure 6.4-17 slightly schematizes the search process after the required alignment patterns were created (either with the current query or with other queries before).

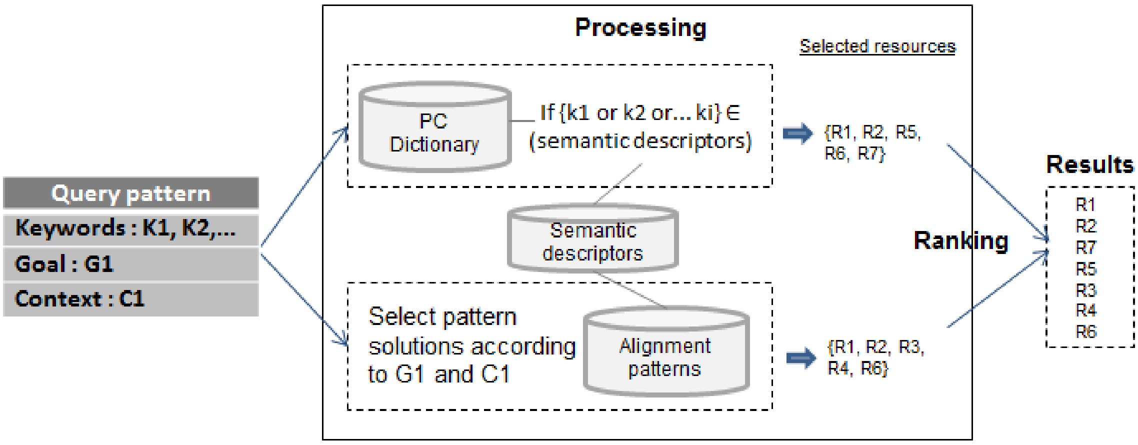


Figure 6.4-17 : Treatment of the user query when the required alignment patterns exit

The keywords specified by the user in the query pattern are matched with the semantics of the semantic descriptors. The dictionary supports this matching by providing semantics to the used concepts. In the S3 approach, a keyword can be any process control description, like a process control method, a control technique, an indicator, etc. Each semantic descriptor of a resource contains its basic process control description. The dictionary in this case deals with the variety of the used vocabulary in this business domain.

After the keyword matching, a set of resources is then selected. A step of filtering and ranking is then applied to identify the final results and display them in an order that can interest the user. In fact, the results are filtered to only keep the ones that match at least the keywords[25]. The resources are classified in four categories, denoted from 1 to 4 (Table 6.4-1). The display of the results to the user is done in the ascending order of these categories (from 1 to 4).

Goal	Context	Keywords	Ranking according to categories
match(g_i)	match(c_i)	match(k_i)	1
match(g_i)	¬match(c_i)	match(k_i)	2
¬match(g_i)	match(c_i)	match(k_i)	2
¬match(g_i)	¬match(c_i)	match(k_i)	3
match(g_i)	match(c_i)	¬match(k_i)	4
match(g_i)	¬match(c_i)	¬match(k_i)	5
¬match(g_i)	match(c_i)	¬match(k_i)	6

Table 6.4-1 : Ranking of the resources according to the matchings found

The logic of ranking is set according to the nature of the matchings found, as follows:

- at least, one matching must be found to consider the results

---

[25] We consider in this case that if the user specifies a set of keywords in his query, he must know approximately what he needs as information. The matching with the keywords is firstly considered in the ranking of the final results

- category 1: the resources that match all the concepts of the user query (keywords, goal and context) will be selected and displayed at first
- category 2: the resources that match the keywords and at least a goal or a context related to the user query will be selected in a second step
- category 3: the resources that match the keywords only will be selected, even if the goal and context don't match

N.B: In our proposed retrieval system, the user does not have to specify all the content of the query pattern. He can also choose to use the keywords only, the goals with the context or without, etc. In fact, in case of lack of keywords in the user query two other categories will be considered (Table 6.4-1):

- category 4: the resources that match both the goal and the context specified by the user in the query pattern
- category 5 : concerns the set of resources whose description meet the goal even if the context does not match
- category 6: concerns the set of resources whose description meet the context even if the goal does not match

- Example:

We consider the following pattern query as user query in Figure 6.4-18.

Query pattern	
Keywords	{'lot', 'out of control'}
Goal	{'cycle time optimization'}
Context	{'process engineering'}

Figure 6.4-18 : Query pattern example

According to the MP ontology, the goal "cycle time optimization" is in fact a complex goal. It will be refined following its relations with other goals and according to the specified business activity "process engineering". Figure 6.4-19 shows the obtained patterns for this example[26]. The resulting resources would be: R1, R2, R3, R4, R6.

---

[26] To be short in the example, the patterns' identifiers (P1, P2, etc.) are used to refer to the pattern solutions instead of the goals

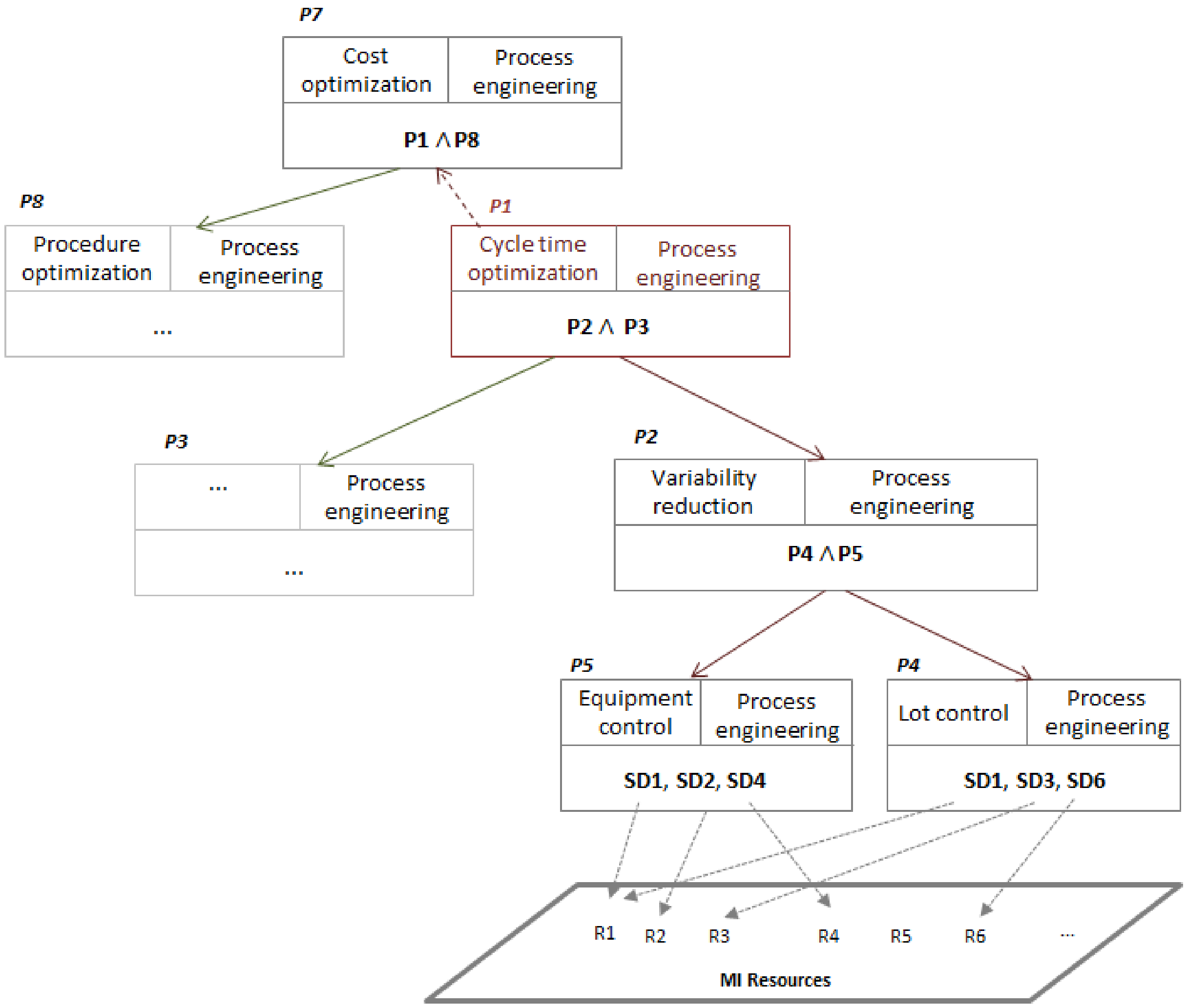


Figure 6.4-19 : The resulting alignment patterns of the example

On other hand, if we take the keywords of the user query in the example (Figure 6.4-18), the system matches the keywords '*lot*' and '*out of control*' with the semantic descriptors. The PC dictionary is used to find the semantic convergences between the keywords of the user query and the semantic descriptors of resources. In the example, the keyword *"out of control"* is a variant of the concept *"ooc"* in the dictionary. The system will then search in the semantic descriptors the descriptions that contain at least one of the two variants (i.e. "ooc" or "out of control").

Imagine we obtain the following resources according to this matching step: R1, R2, R5, R6, R7. These resources and the ones obtained with the alignment-patterns' solutions will be then classified following the ranking categories. Table 6.4-2 shows the resulting rankings.

	Goal	**Context**	**Keywords**	**Ranking**
SD1	match(g_i)	match(c_i)	match(k_i)	1
SD2	match(g_i)	match(c_i)	match(k_i)	1
SD3	match(g_i)	match(c_i)	¬match(k_i)	4
SD4	match(g_i)	match(c_i)	¬match(k_i)	4
SD5	¬match(g_i)	¬match(c_i)	match(k_i)	3

SD6	match(g_i)	match(c_i)	match(k_i)	1
SD7	match(g_i)	¬match(c_i)	match(k_i)	2

Table 6.4-2 : The resulting rankings according to the example

Finally, the results that better satisfy the user need in this example would be: R1, R2, R6, R7, R5 and will be displayed in this order.

6.4.4. Benefits of the Pattern-based Search

The pattern-based search has several benefits:

- by using the alignment patterns, users of different business profiles (process control experts and other business actors) can share their business needs
- the pattern-based search fosters the reuse of business need solutions: the creation and reuse of alignment patterns facilitates meeting users' needs and maintains the link between the resources and the business needs they address
- the pattern-based search facilitates the capture of the users' needs: the expression of user queries is assisted with a query pattern which fits into business needs in a context (the keywords reflect elementary needs, and the goals reflect complex business needs)
- the resource description is enriched and maintained progressively with the alignment patterns of the retrieval system. In fact, the pattern-based search completes the bottom-up retrieval technique in both the resource description and retrieval aspects
- the approach provides an original goal orientation to the retrieval system, where each complex business need expressed with goals can outline alternatives to its satisfaction. This goal-oriented aspect enables to better target the MI resources relevant to business-actors' needs during the search process.

6.5. Conclusion

This chapter focused on the pattern-based search, which relies on contextualizing the MI-resource retrieval by adding another level of business description to the resources. In summary, the search patterns are used in the S3 approach to:

- assist the capture of user needs according to a business context
- provide a business-need-focused search and increase the relevance of the retrieved results to the user context
- align the MI resources with the business needs by filling the gap between them through the goal decomposition principle

Basically, the search patterns are created dynamically when the user tries to retrieve a set of resources according to a business context. The search system creates alignment patterns to progressively capture business need artifacts and reuse them for further needs. The creation process is very based on how the relations between goals are defined in the business ontology. At the end of the search process, the system ranks the resulting resources according to a set of criteria. The interesting aspect of using search patterns is that a need can be addressed from any level of abstraction, either if the business need is simple or complex (abstract). Finally, by using this resource retrieval approach, each pattern identified is an artifact that contributes to the alignment of the MI resources with the business needs in the company.

Chapter 7: Approach Implementation and Experimentation

Résumé

Ce chapitre propose un prototype d'implémentation de l'approche S3 et présente quelques expérimentations de l'approche faites avec des ressources de l'entreprise STMicroelectronics. Les choix d'implémentation proposés ont été faits en fonction des besoins de l'entreprise.

Le prototype développé comprend trois fonctions de recherche : une fonction de mapping et référencement qui implémente la stratégie ascendante de recherche, une fonction de recherche à base de patterns relative à la stratégie descendante et une fonction de recherche à l'aide de cartes de concepts « Topic Maps ». Cette dernière est proposée pour faciliter la découverte et la compréhension de la sémantique associée aux ressources. Elle permet également d'avoir une visibilité sur les besoins et ressources existantes en entreprise, facilitant ainsi la maintenance du système.

L'une des spécificités de ce prototype est que la base de connaissances qui stocke les descripteurs et les patterns de recherche est basée sur le modèle « Topic Maps ». Ce modèle permet d'encapsuler différents types de connaissances quel que soit leur niveau d'abstraction grâce à la notion de « topic » et de relations entre « topics ».

Enfin, l'expérimentation de l'approche S3 dans l'entreprise STMicroelectronics a montré des résultats intéressants et prometteurs pour la suite. Les principaux traits de l'approche ont pu être améliorés et validés progressivement avec les experts métier de l'entreprise. Cependant, le prototype d'implémentation est encore en phase de test et de correction notamment pour la recherche « top-down » de l'approche.

7.1. Introduction

This chapter presents the integration of the S3 approach in a Topic-Maps-based application intended for MI-resource retrieval. The application is at a prototyping stage within STMicroelectronics. In this prototype, the results of the approach are progressively capitalized in a knowledge base that is based on the Topic-Maps' meta-model. This knowledge base stores the resource description in order to be efficiently exploited with search functions. There are three types of search functions in the prototype: the first is the mapping approach that allows exploring the MI resources in the company and referencing the relevant ones in the knowledge base. The second is the pattern-based search that provides a business-need focused search. The third type of search is a Topic-Maps-based search. It provides a whole view of the concepts around the user query in a form of a structured graph of concepts. The proposed prototype and knowledge base have been specifically tailored for the STMicroelectronics' needs.

Thus, we present in this chapter the knowledge base that stores the resource semantics, the prototype architecture and functionalities, and we present at the end some experimentations of the bottom-up retrieval approach done with STMicroelectronics.

7.2. Prototyping and Implementation

7.2.1. The Topic-Maps' Paradigm as Knowledge Base

We chose to create a knowledge base basing on the Topic-Maps model to efficiently use the resource description resulting from the top-down and bottom-up retrieval techniques of the S3 approach. The concepts of the Topic-Maps' standard (cf. Chapter 2) were chosen as the basis of the knowledge base model for its simplicity and flexibility in handling semantics. The main concepts provided by the Topic-Maps' meta-model enable to describe heterogeneous resources with high semantic abstraction and regardless of their type and their location. A Topic can therefore represent any subject from the real world with any desired level of granularity by typing topics and associations [Bouzid et al. 2012a]. According to our purpose, we use the main entities of the Topic-Maps' standard to ensure the integration and the structuring of resource semantics.

This meta-model relies on the basic concepts of the standard Topic-Maps meta-model, known as the TAO model (Topic, Association, Occurrence) [Kannan 2010]:

- **Topic**

The topic regroups all types of topics that can represent a subject of a business context. In addition, each topic can be a sub-topic of another topic. The topics of the resulting knowledge base are identified according to the concepts of the semantic descriptors of resources, as defined in the S3 approach. Examples of used topics include the objectives (either manufacturing or control objective), the control methods, data, etc. Concretely, the topic types are not implemented in the knowledge base model, all the referenced concepts are considered topics. The distinction between their types is made with the association types and is used in the programming logic only.

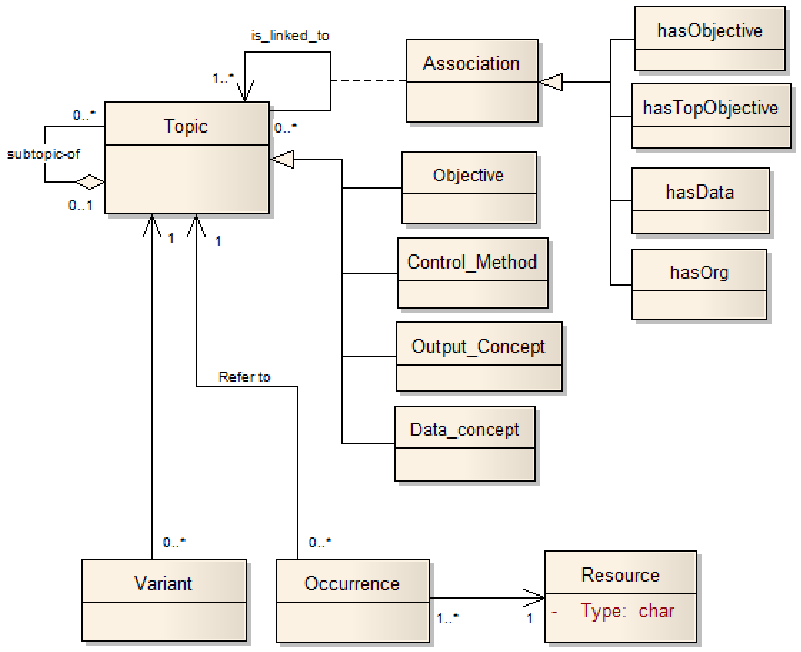


Figure 7.2-1 : The Topic-Maps' meta-model used for the knowledge base creation

- **Association**

The association refers to a semantic relation between two topics. The interesting aspect of using the associations is in typing them following the meaning we want to infer. Thus, the associations enable to expand a hierarchy of topics into complex topic networks. In our implemented model, the used associations are the relations of the MP ontology. These relations allow distinguishing between the types of topics used in the knowledge base. For example, the association *hasObjective()* links a topic *control_method* to a topic *objective*.

- **Occurrence**

The occurrence is a kind of a resource identifier. It could be a URL (Uniform Resource Locator) of a web resource, a document identifier (DOI), a DNS (Domain Name System) of a server and so forth. Example of an occurrence:

//rountace01/kla ace/rousset site/process_control/spc/control_chart_lot.rcp

We also use two other fundamental concepts of the Topic-Maps' paradigm: the resources and the variants.

- **Resource**

A resource is any addressable physical object. It is not stored physically in the knowledge base, only its location (URL or URI) is considered. We added a class *resource* in our Topic-Maps' meta-model to specify a type of a resource (example: Kla ace xp, BO, APF, etc.). A resource can be linked to one or many topics through the occurrence concept (Figure 7.2-1).

- **Variant**

A variant refers to any variant of a topic name [Pepper 2010], used to express several rewrites or meanings of the same topic. This concept responds at first to terminological and grammatical problems related to a topic. For example, *WFY* is a variant of *Wafer Fab Yield*.

Our chosen Topic-Maps' meta-model enables to handle semantic expressiveness of business contexts with flexibility in use and easiness in maintenance. Hence, when there is a new type of concept to take into account in the knowledge base model, there is no change to do, because it is considered as a topic in general. Only the association type(s) that link(s) this new type of concept to another type(s) must be defined.

Figure 7.2-2 illustrates the relation between the knowledge base and the MI resources that are usually dispatched in the company network. The resulting logical Topic-Maps' model is specifically intended to fill the gap between the resources and the business needs, by giving to the end user a semantic access to them [Bouzid et al. 2012b]. Ultimately, this knowledge base is designed in a way to continuously maintain the alignment between the MI resources in the company and the business needs of the business actors.

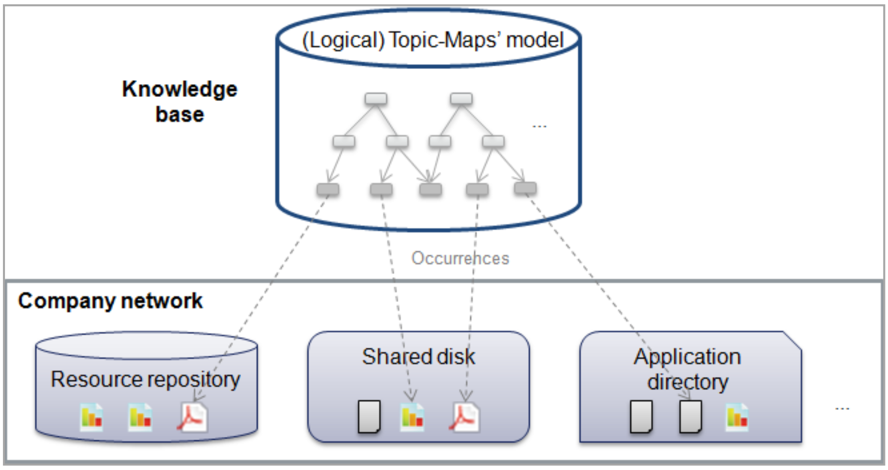


Figure 7.2-2 : Relation between the knowledge base and the resources

7.2.2. Prototype Architecture and Functionalities

Considering the general issues of this work and the requirements of the STMicroelectronics Company, the implementation choices of the S3 approach were done according to the Information-Technology policy of the company. Also, only the functionalities related to the resources retrieval will be presented.

The main functionalities of the application are illustrated with a use case diagram in Figure 7.2-3.

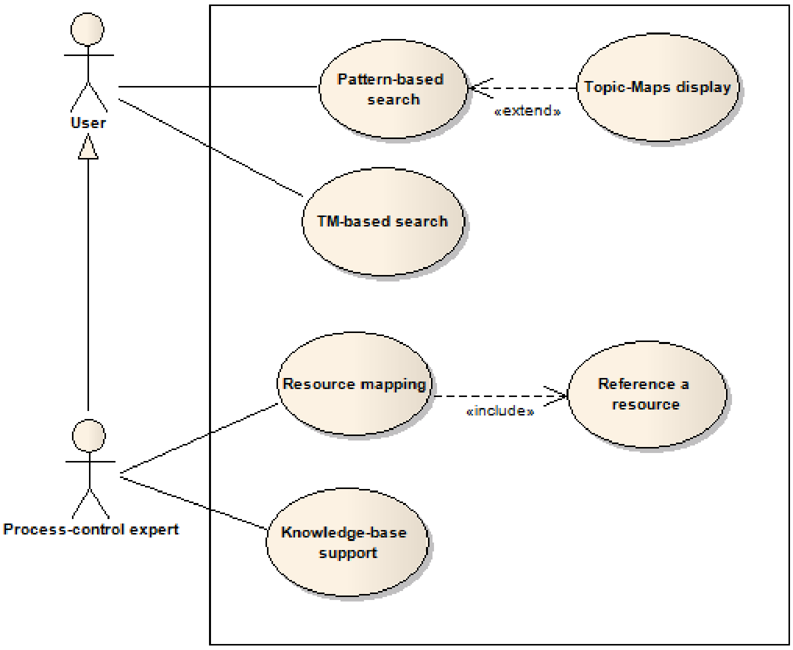


Figure 7.2-3 : Use case diagram of the prototype

A user of the retrieval system could be a manufacturing process engineer, a process-control expert, a technician, etc. Two search functions are proposed to the user: the *pattern-based search* and the *TM-based search*:

- *Pattern-based search*: the user expresses his need with the query pattern which uses search patterns for the query processing
- *TM-based search*: the user can explore a set of resources and how they are linked to each other with a simple keyword search. This last search function is proposed for informational purposes to the end users of the system.

Among the users, the process control experts have access to some advanced functionalities in the application. These functionalities are:

- *Resource mapping*: the semantic mapping approach is used in this case. The user can select a resource location (e.g. shared disk, application directory, etc.) and the system provides the MI resources found with an estimated business description
- *Resource referencing*: this functionality enables the user to reference the resources that are requested or may be requested by other users. The referencing can be done by mapping the MI resources (because this functionality is included in the resource mapping)
- *Knowledge-base support*: this functionality includes some functions for managing the MI resources and their associations in the knowledge base (i.e. *update/delete* of the resource description and the search patterns).

<u>N.B</u>**:** *The maintaining of the TM-application including the knowledge base is out of scope of the S3 approach. Thereby, this part of work will not be reported in this manuscript.*

Otherwise, the proposed functionalities (mapping, referencing and management) are limited to some process control experts to avoid errors in indexing the MI resources. In fact, *in the case of the STMicroelectronics' context, the process control experts are the most qualified to* *determine if a resource can be shared and if its description is reliable and can be understandable by all the users.*

As we can see, the semantic mapping approach is integrated in the application as a mean for exploring the MI resources in the company and referencing the ones relevant to the manufacturing-process control purposes. This task is precisely done by the process control experts. The resource retrieval process can be afterwards progressively enhanced with the top-down search functions (pattern-based search and TM-based search). These functions exploit the business semantics associated to the resources with the mapping approach to help the users in searching for their needs.

We chose to implement the prototype with a standard three-tier architecture [IBM 2009] to enable a modular design of the application with an easy support in the long term. Figure 7.2-4 depicts the used architecture.

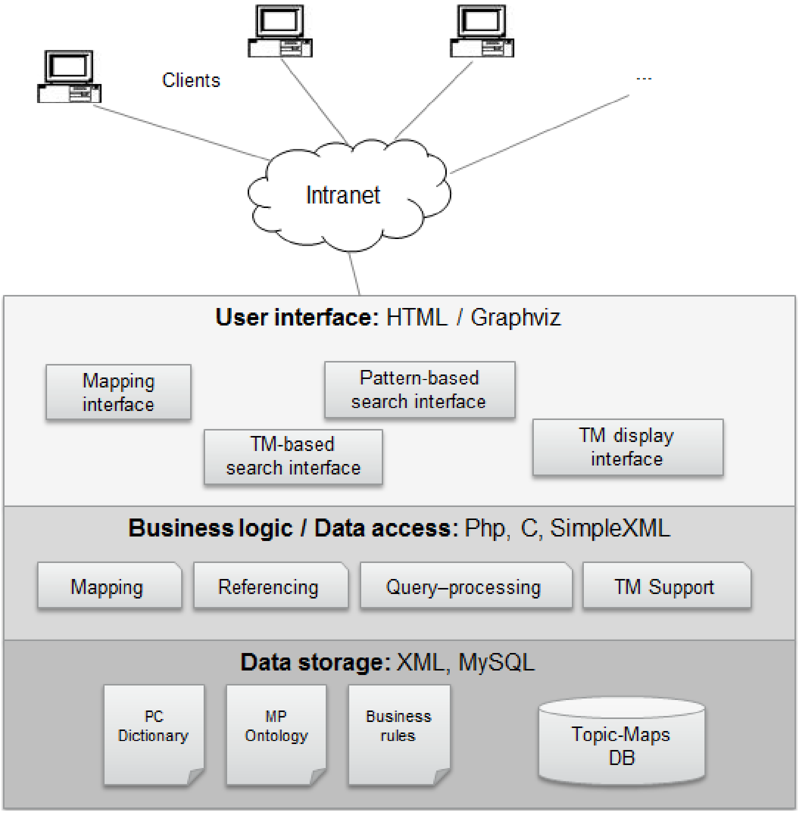


Figure 7.2-4 : Three-tier architecture of the prototype

7.2.2.1. User Interfaces

The presentation tier is composed of the main interfaces of the retrieval system, i.e. the resource mapping, the pattern-based search and the TM-based search.

These interfaces are developed with the HTML language supported with CSS sheets for presentation design. An external application called Graphviz is also used for the display of Topic Maps in a graphic form.

7.2.2.2. Business-Logic Processing

We separated the processing of the business logic in four main modules: a *mapping module*, a *referencing module*, a *query processing module* and a *TM support module*. Each module is composed of a set of functions required to carry out the user queries. These functions have been developed in a way to be reused.

The ***mapping module*** is composed of the main functions presented in the semantic mapping approach, in particular, the string-similarity computation and the analysis of the ontology-relations to find target descriptions (scenario 1, scenario 2, etc.).

The ***referencing module*** is related to the storage process of the resources' description with their location (i.e. URL identifier) in the Topic-Maps database.

The ***query processing module*** comprises the functions for processing the user queries related to the pattern-based search and the TM-based search. The creation of search patterns and their reuse with the goal-oriented mechanisms are also included here.

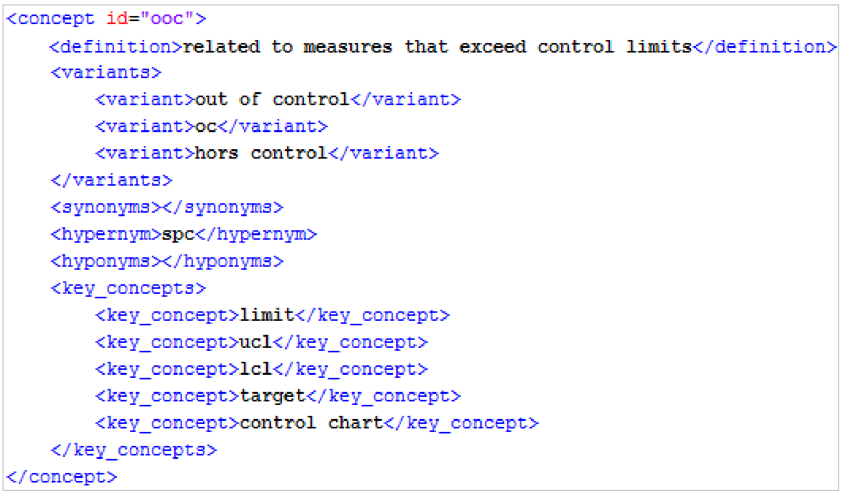
The ***TM support module*** comprises the functions used for the manual configuration and maintaining of the knowledge base.

The Php technology is mainly used for the programming of the business logic with the C language for the string-similarity computation (used to speed up the mapping process for large amounts of data). SimpleXML (an xml parser) is also used for xml data access.

7.2.2.3. Data Storage

Two types of data storage formats are used in our implementation architecture: xml format and MySQL database. The xml format is used to implement the PC dictionary, the MP ontology and the business rules used for the inference purpose. The MySQL database is used to reference the resources with their description. All these implementation choices fit into the company requirements.

The PC dictionary contains up to now 26 entries, with 232 concepts in whole (including variants, synonyms, hypernyms, hyponyms and key concepts). An extract is given in Figure 7.2-5.

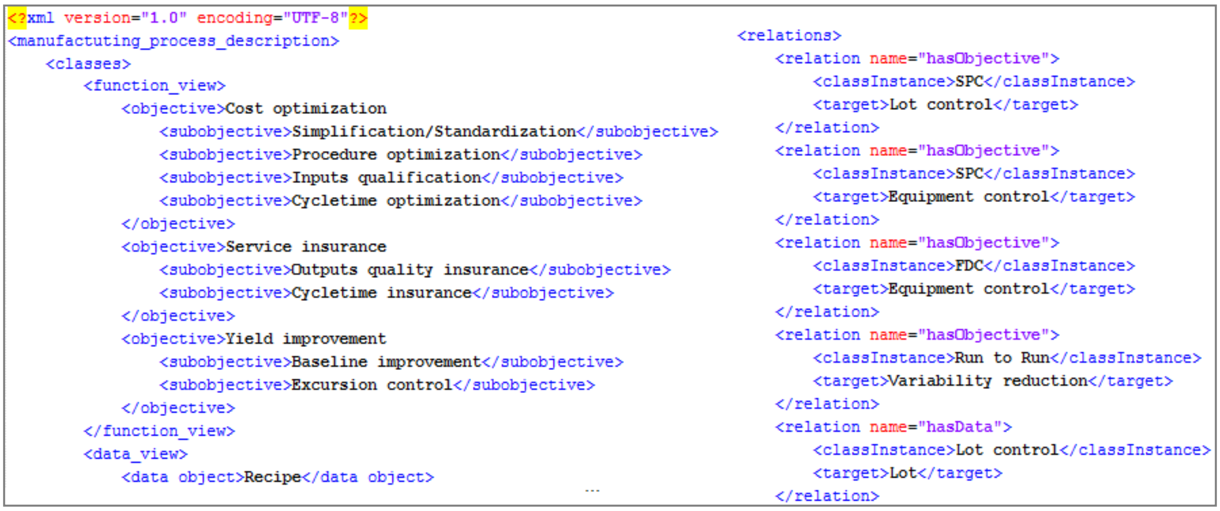


```
<concept id="ooc">
    <definition>related to measures that exceed control limits</definition>
    <variants>
        <variant>out of control</variant>
        <variant>oc</variant>
        <variant>hors control</variant>
    </variants>
    <synonyms></synonyms>
    <hypernym>spc</hypernym>
    <hyponyms></hyponyms>
    <key_concepts>
        <key_concept>limit</key_concept>
        <key_concept>ucl</key_concept>
        <key_concept>lcl</key_concept>
        <key_concept>target</key_concept>
        <key_concept>control chart</key_concept>
    </key_concepts>
</concept>
```

Figure 7.2-5 : Extract of the PC dictionary

The MP ontology is composed of a set of *classes* and *relations* (elements of the xml document). Each class represents a view of the manufacturing description. As reminder, the function view is composed of business objectives, the data view regroups the business entities related to the manufacturing activity, the organization view regroups the set of business activities and profiles related to the manufacturing process, and finally the control view is described with objectives, statistical technique and control methods. The relations in the ontology establish the link between the instances of the classes. Fives type of relations were defined during the construction of the MP ontology.

Currently, the ontology implemented within STMicroelectronics contains 86 element instances. An extract of the MP ontology is in Figure 7.2-6.

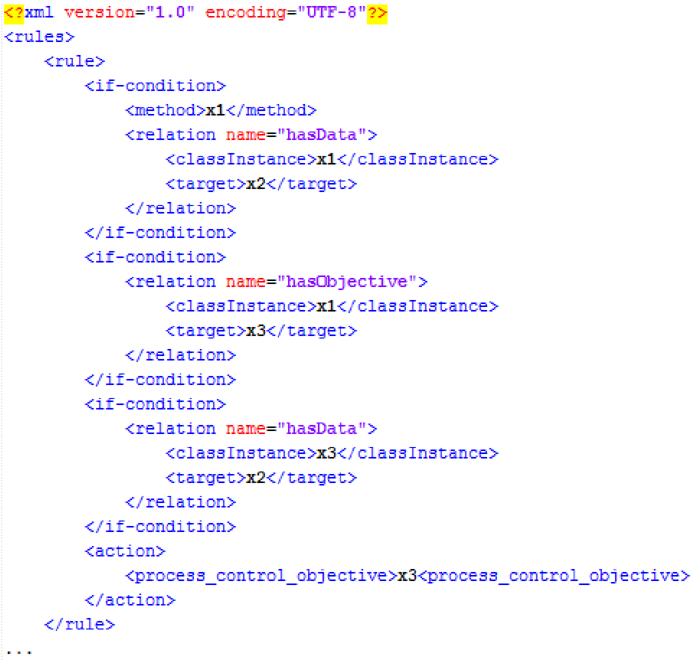


```
<?xml version="1.0" encoding="UTF-8"?>
<manufactuting_process_description>
    <classes>
        <function_view>
            <objective>Cost optimization
                <subobjective>Simplification/Standardization</subobjective>
                <subobjective>Procedure optimization</subobjective>
                <subobjective>Inputs qualification</subobjective>
                <subobjective>Cycletime optimization</subobjective>
            </objective>
            <objective>Service insurance
                <subobjective>Outputs quality insurance</subobjective>
                <subobjective>Cycletime insurance</subobjective>
            </objective>
            <objective>Yield improvement
                <subobjective>Baseline improvement</subobjective>
                <subobjective>Excursion control</subobjective>
            </objective>
        </function_view>
        <data_view>
            <data object>Recipe</data object>
                                                  ...
```

```
    <relations>
        <relation name="hasObjective">
            <classInstance>SPC</classInstance>
            <target>Lot control</target>
        </relation>
        <relation name="hasObjective">
            <classInstance>SPC</classInstance>
            <target>Equipment control</target>
        </relation>
        <relation name="hasObjective">
            <classInstance>FDC</classInstance>
            <target>Equipment control</target>
        </relation>
        <relation name="hasObjective">
            <classInstance>Run to Run</classInstance>
            <target>Variability reduction</target>
        </relation>
        <relation name="hasData">
            <classInstance>Lot control</classInstance>
            <target>Lot</target>
        </relation>
```

Figure 7.2-6 : Extract of the MP ontology

The business-rule file (Figure 7.2-7) is used to support the mapping of resources using the ontology relations in the bottom-up technique. These rules are translated in a way to be tested with a set of facts in scenario 2 of the mapping process (cf. Algorithm 6 and Algorithm 7). These rules are written in the form of *If/Then* statements and are analyzed and checked by the system for the processing of the forward chaining.



```
<?xml version="1.0" encoding="UTF-8"?>
<rules>
    <rule>
        <if-condition>
            <method>x1</method>
            <relation name="hasData">
                <classInstance>x1</classInstance>
                <target>x2</target>
            </relation>
        </if-condition>
        <if-condition>
            <relation name="hasObjective">
                <classInstance>x1</classInstance>
                <target>x3</target>
            </relation>
        </if-condition>
        <if-condition>
            <relation name="hasData">
                <classInstance>x3</classInstance>
                <target>x2</target>
            </relation>
        </if-condition>
        <action>
            <process_control_objective>x3<process_control_objective>
        </action>
    </rule>
...
```

Figure 7.2-7 : Extract of the business rules

For example, the rule illustrated in Figure 7.2-7 corresponds to:

$$control\_method\ (x1) \wedge hasData(x1, x2) \wedge hasObjective(x1, x3) \wedge hasData(x3, x2)$$
$$\Rightarrow control\_objective\ (x3)$$

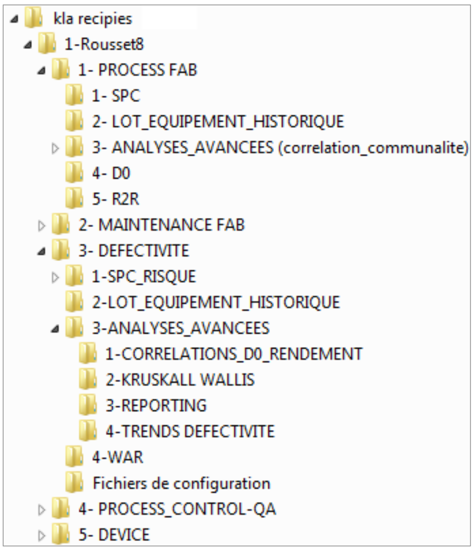
We chose to set up a file of rules with *If/Then* statements in the presentation form, to simplify the maintaining process to the business experts when new rules need to be added.

Finally, we use a MySQL database to implement the knowledge base that capitalizes on the resource description. A relational database was precisely chosen to respond to performance issues in case of large amount of data. The knowledge base is used to store both the semantic descriptors and the search patterns of the approach, in particular because the Topic-Maps' model can encapsulate all these types of knowledge.

7.2.3. Resource Retrieval

7.2.3.1. Bottom-up Resource Retrieval

We developed a mapping interface that takes as input any location of a resource repository in the company network. Examples of resource repositories include shared disks, software directories and document storage servers of the company. The main types of the mapped resources are pdf files containing charts of manufacturing data, csv files containing raw or transformed data extracted from manufacturing systems, and specific files to statistical tools (.rcp formats). The approach is based on mapping the resources' names as it is the only common and readable semantic information in the existing MI resources in the company. We also take into account the concepts of the directories' taxonomies (i.e. the directory path) where the selected resources are stored (Figure 7.2-8). The vocabulary used in these taxonomies is usually defined by the business experts.



- kla recipies
  - 1-Rousset8
    - 1- PROCESS FAB
      - 1- SPC
      - 2- LOT_EQUIPEMENT_HISTORIQUE
      - 3- ANALYSES_AVANCEES (correlation_communalite)
      - 4- D0
      - 5- R2R
    - 2- MAINTENANCE FAB
    - 3- DEFECTIVITE
      - 1-SPC_RISQUE
      - 2-LOT_EQUIPEMENT_HISTORIQUE
      - 3-ANALYSES_AVANCEES
        - 1-CORRELATIONS_D0_RENDEMENT
        - 2-KRUSKALL WALLIS
        - 3-REPORTING
        - 4-TRENDS DEFECTIVITE
      - 4-WAR
      - Fichiers de configuration
    - 4- PROCESS_CONTROL-QA
    - 5- DEVICE

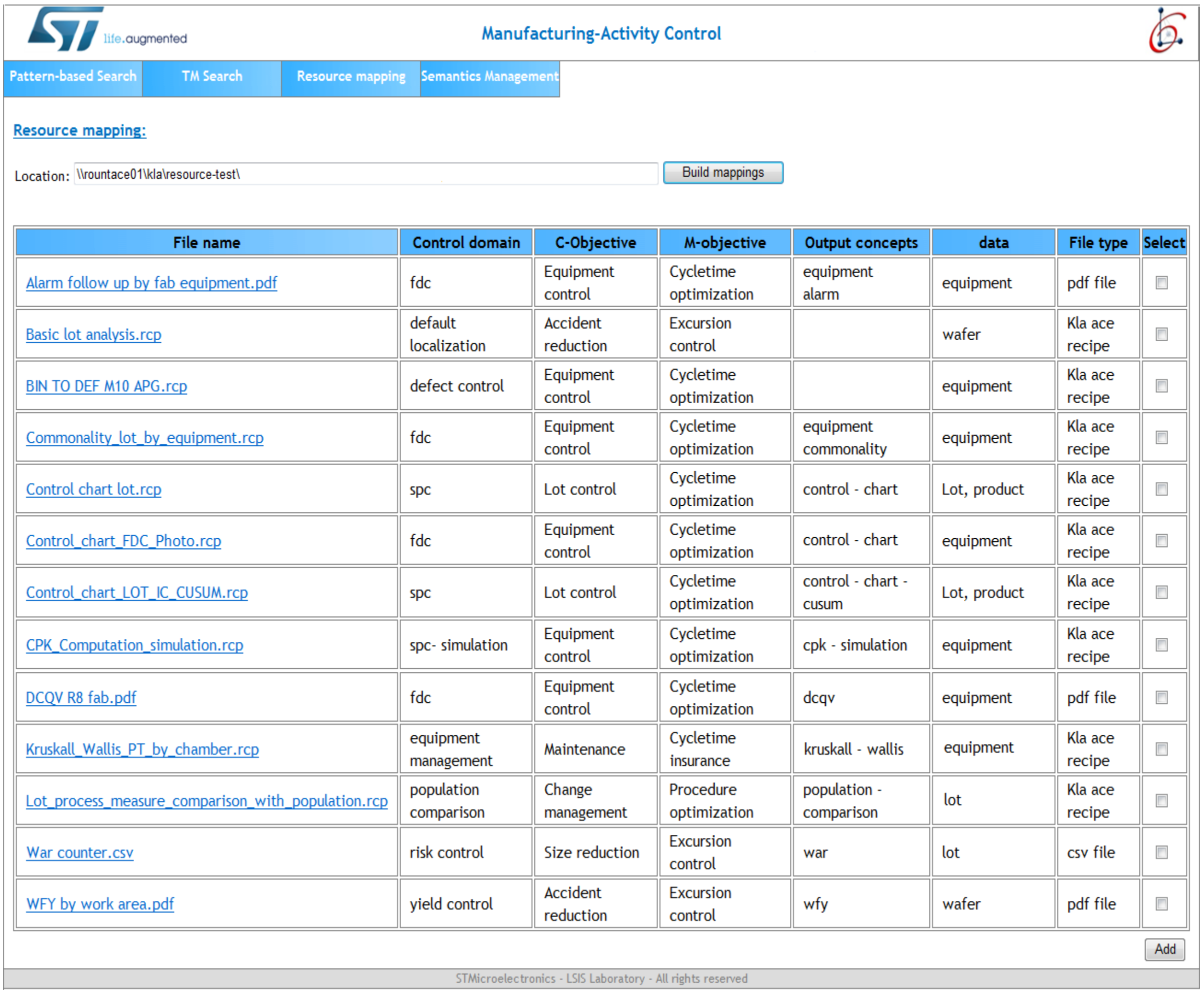
Figure 7.2-8 : Example of directory taxonomy

The produced output is a semantic descriptor for each resource as depicted in Figure 7.2-9.

The mapping interface enables a process control expert to explore the existing MI resources that they created or created by other business actors (mainly the engineers involved in the manufacturing process). Each resource can be selected with its description and added in the knowledge base (in Figure 7.2-9, this functionality is available in the last column of the results' table of the mapping interface). If the resource is already referenced, the system will check if there are differences in the referenced description in order to determine if there are updates to make (in this case the system asks the user before updating).

The referencing function takes all the types of concepts obtained with the resource mapping interface and stores them as topics (if they are not already stored). The system then checks the used types of concepts (which were specified in the semantic mapping process), and constructs the adequate associations between topics in the knowledge base.

Note that storing each MI resource (identifier) with its semantic descriptor was required by the company because the resulting knowledge base enables to capitalize on process-control know-how and can be reused for several other process control purposes (e.g. detecting obsolete and unusable resources, identifying missing needs and redundant needs).



Location: \\rountace01\kla\resource-test\

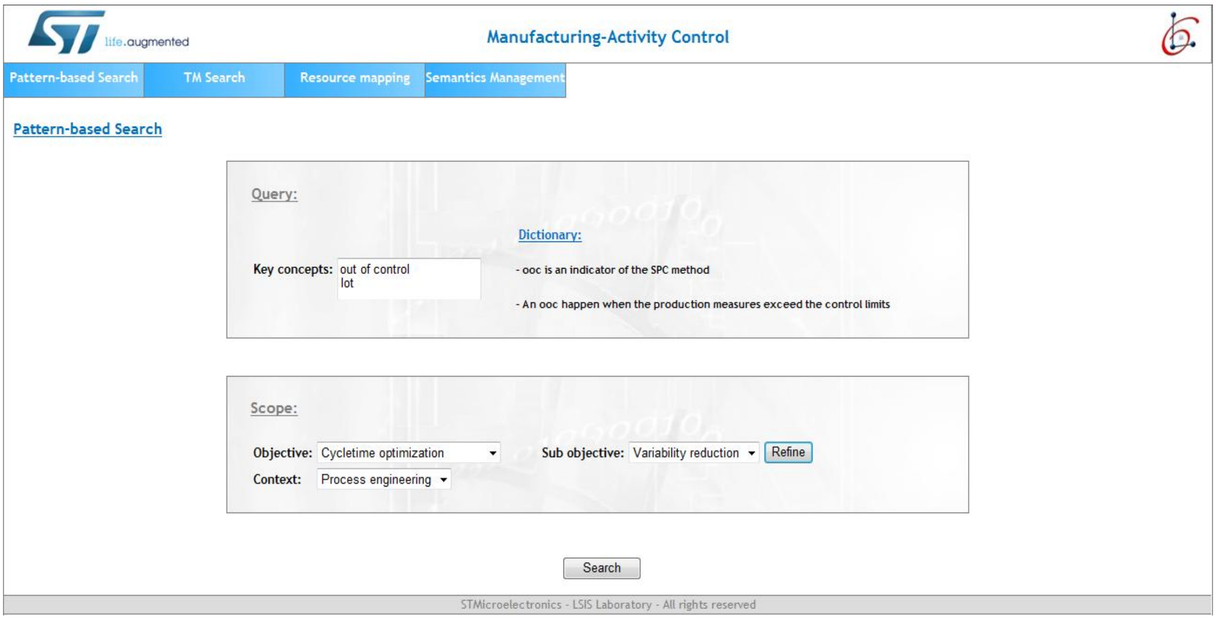
File name	Control domain	C-Objective	M-objective	Output concepts	data	File type	Select
Alarm follow up by fab equipment.pdf	fdc	Equipment control	Cycletime optimization	equipment alarm	equipment	pdf file	☐
Basic lot analysis.rcp	default localization	Accident reduction	Excursion control		wafer	Kla ace recipe	☐
BIN TO DEF M10 APG.rcp	defect control	Equipment control	Cycletime optimization		equipment	Kla ace recipe	☐
Commonality_lot_by_equipment.rcp	fdc	Equipment control	Cycletime optimization	equipment commonality	equipment	Kla ace recipe	☐
Control chart lot.rcp	spc	Lot control	Cycletime optimization	control - chart	Lot, product	Kla ace recipe	☐
Control_chart_FDC_Photo.rcp	fdc	Equipment control	Cycletime optimization	control - chart	equipment	Kla ace recipe	☐
Control_chart_LOT_IC_CUSUM.rcp	spc	Lot control	Cycletime optimization	control - chart - cusum	Lot, product	Kla ace recipe	☐
CPK_Computation_simulation.rcp	spc- simulation	Equipment control	Cycletime optimization	cpk - simulation	equipment	Kla ace recipe	☐
DCQV R8 fab.pdf	fdc	Equipment control	Cycletime optimization	dcqv	equipment	pdf file	☐
Kruskal_Wallis_PT_by_chamber.rcp	equipment management	Maintenance	Cycletime insurance	kruskall - wallis	equipment	Kla ace recipe	☐
Lot_process_measure_comparison_with_population.rcp	population comparison	Change management	Procedure optimization	population - comparison	lot	Kla ace recipe	☐
War counter.csv	risk control	Size reduction	Excursion control	war	lot	csv file	☐
WFY by work area.pdf	yield control	Accident reduction	Excursion control	wfy	wafer	pdf file	☐

Figure 7.2-9 : The resource mapping interface

7.2.3.2. Top-down Resource Retrieval

- <u>The Pattern-based search</u>

Figure 7.2-10 shows the pattern-based search interface with an example of need. The system takes in the user query one or a set of keywords and a business scope. A keyword can be any concept of the process control or manufacturing description. When the user writes keywords related to the process control, the system automatically displays meta-information about each concept if it is defined in the dictionary, so to help the user in choosing the right concepts to express his need. Regarding the scope, the concepts of the MP ontology are used to assist the user in defining the scope of his need. In case of selecting a goal, the system can propose to the user sub goals related to the selected goal and to the business activity that the user specifies. In fact, for each selected goal, the user can refine it, whenever possible. When a manufacturing objective has sub objectives defined in the MP ontology, the system automatically proposes to refine it (with the button “Refine”). When a sub objective is selected and has, in turn, sub objectives, the system will, one more time, propose the refinement option to the user (optional function). In any case, the system processes the goal refinement task with the alignment patterns regardless of the level of abstraction of the selected goal by the user. In the example of Figure 7.2-10, we chose *Cycletime optimization* as manufacturing objective and we chose to refine it by selecting as sub objective *variability reduction*. The system will then start the goal refinement from this last goal. Figure 7.2-11 shows the resulting resources with some of their semantics.



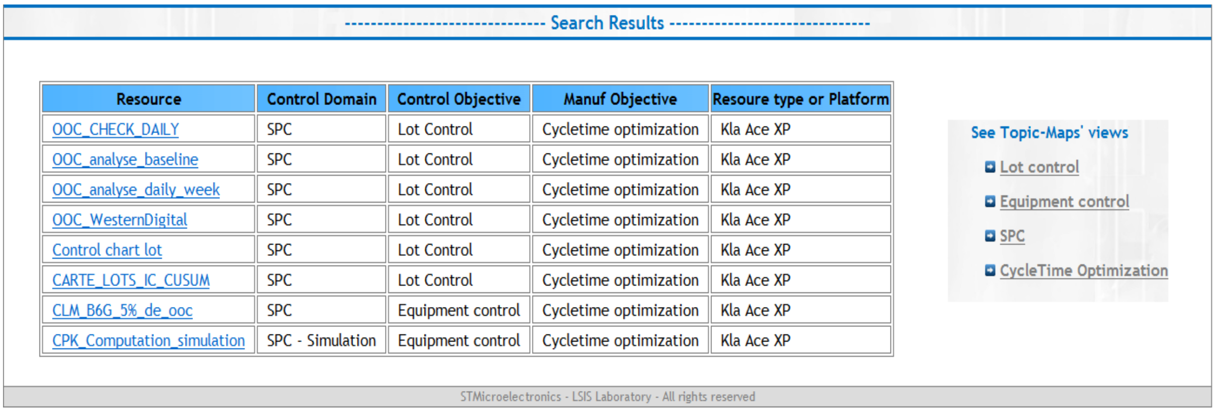
Key concepts: out of control
lot

- ooc is an indicator of the SPC method
- An ooc happen when the production measures exceed the control limits

Objective: Cycletime optimization
Sub objective: Variability reduction
Context: Process engineering

Figure 7.2-10 : Interface of a pattern-based search

In the interface of the search results (Figure 7.2-11), we propose two types of results to enable the user to explore the resources in different ways. The search results that match with the user query are simply displayed in a table of results. This table recapitulates the selected criteria by the user and displays the corresponding resources. We propose in the same interface an access to Topic-Maps’ views. The proposed Topic Maps are proposed according to the business semantics of the resources displayed in the search results. As we can see in Figure 7.2-11, the system proposes four types of Topic Maps corresponding to the cited types of topics. For example, in Figure 7.2-12, the Topic Map “Equipment control” is composed of all the resources related to this control objective.


------------------------------ Search Results ------------------------------
See Topic-Maps' views
Lot control
Equipment control
SPC
CycleTime Optimization
STMicroelectronics - LSIS Laboratory - All rights reserved

Resource	Control Domain	Control Objective	Manuf Objective	Resoure type or Platform
OOC_CHECK_DAILY	SPC	Lot Control	Cycletime optimization	Kla Ace XP
OOC_analyse_baseline	SPC	Lot Control	Cycletime optimization	Kla Ace XP
OOC_analyse_daily_week	SPC	Lot Control	Cycletime optimization	Kla Ace XP
OOC_WesternDigital	SPC	Lot Control	Cycletime optimization	Kla Ace XP
Control chart lot	SPC	Lot Control	Cycletime optimization	Kla Ace XP
CARTE_LOTS_IC_CUSUM	SPC	Lot Control	Cycletime optimization	Kla Ace XP
CLM_B6G_5%_de_ooc	SPC	Equipment control	Cycletime optimization	Kla Ace XP
CPK_Computation_simulation	SPC - Simulation	Equipment control	Cycletime optimization	Kla Ace XP

Figure 7.2-11 : Results of the assisted search

The aim of the Topic-Maps' display is to help the user understanding the resource description and to explore, by the way, some other resources closely related to his expressed need.

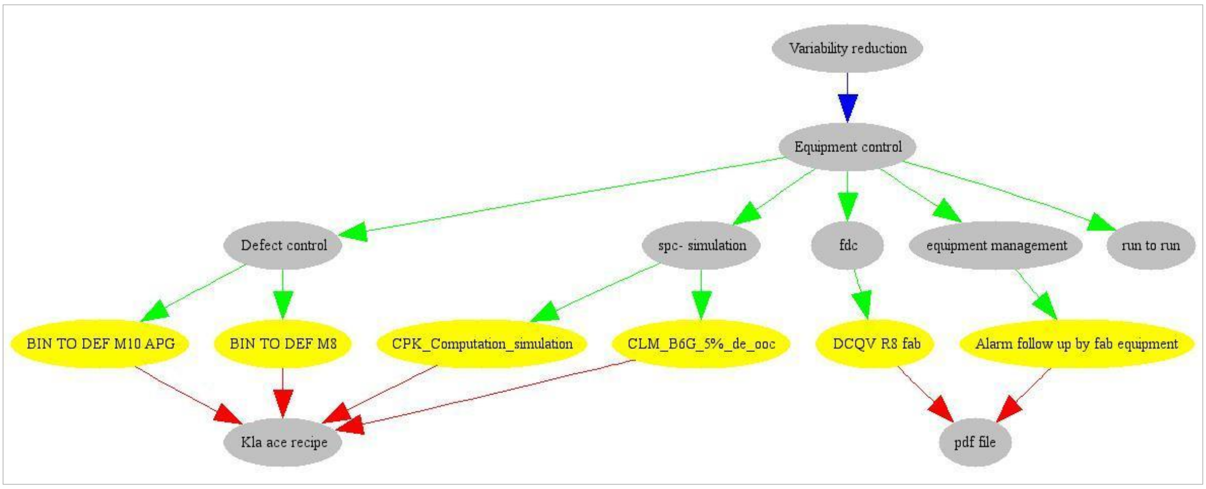


Figure 7.2-12 : The Topic Map of equipment control

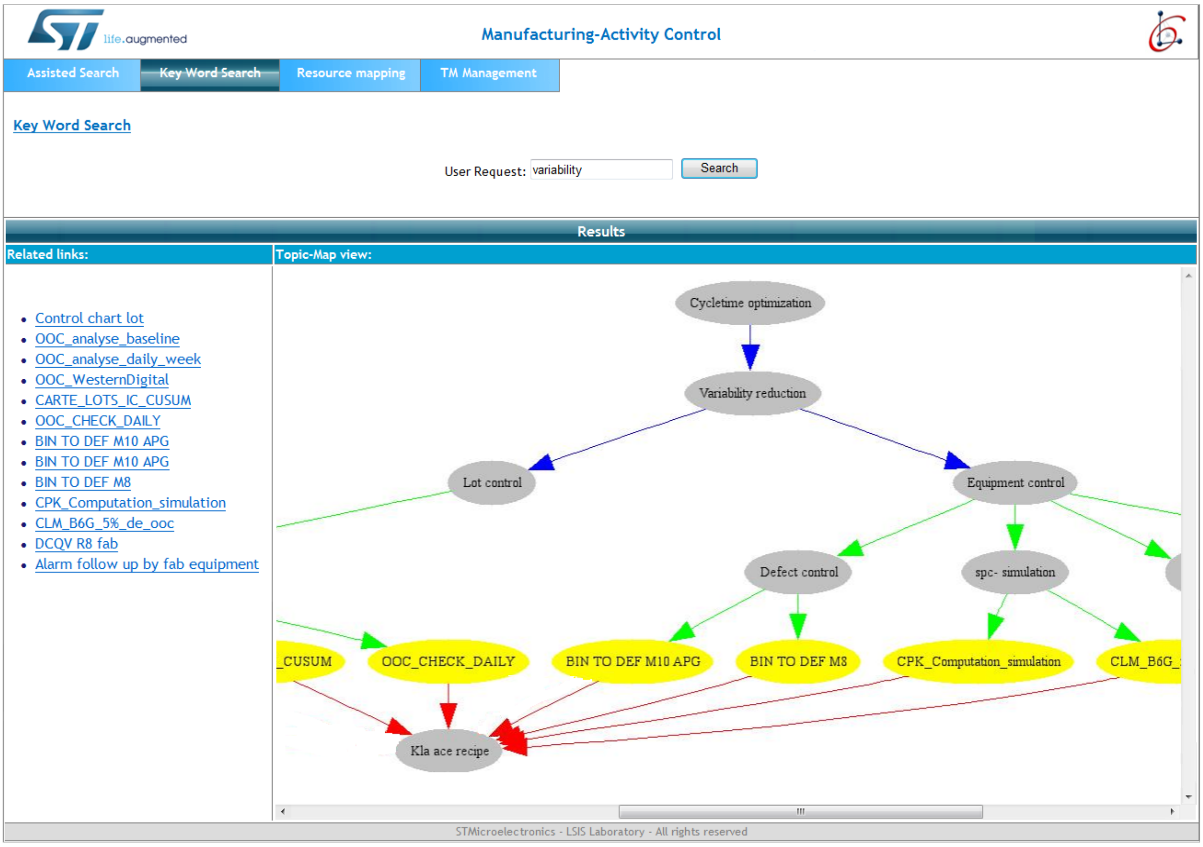
- <u>The TM-based Search</u>

The third search function of the retrieval system is the semantic navigation on the resources using Topic-Maps [Bouzid et al. 2012b]. This kind of search provides the business actors an overview of the resources linked to a given business concept. In the proposed search, the user expresses his query with one or a set of keywords, and the system builds the search results in a form of a graph of concepts. This one is, in fact, a graphical presentation of a Topic Map that matches with at least one of the keywords (also, several Topic Maps can be obtained). This Topic Map is built from the Topic-Maps' model of the knowledge base. Figure 7.2-13 depicts the TM-based search interface.

If we take the example of a user that needs the resources that deal with the *variability* notion, as we can see in the example, by typing *variability* as a keyword in the search interface, the system returns all the topics that contain this concept with their associated topics and occurrences. The resources are accessible from their occurrences in the left frame of the interface and their context can be explored with the resulting Topic Map. This Topic Map seeks guiding the end users in exploring the link between the MI resources and their business description. In the example, all the MI resources found under the topic "variability" are displayed. A detailed view of the resulting Topic Map of the example is depicted in Figure 7.2-14.

To simplify the display of the Topic Map, only some concepts related to the business usage of the resources are displayed. The exposed concepts are:

- the goals: the topics related to the manufacturing objectives and control objectives. In the Topic-Map example, these topics are linked to each other with blue edges
- the control methods or domains of control: in the example, these concepts are linked to each other with green edges
- the *resources' names* and the used COSTS platforms. In Figure 7.2-14, the topics of yellow color represent the resources and they point with red edges to the resource types.



User Request: variability

- Control chart lot
- OOC_analyse_baseline
- OOC_analyse_daily_week
- OOC_WesternDigital
- CARTE_LOTS_IC_CUSUM
- OOC_CHECK_DAILY
- BIN TO DEF M10 APG
- BIN TO DEF M10 APG
- BIN TO DEF M8
- CPK_Computation_simulation
- CLM_B6G_5%_de_ooc
- DCQV R8 fab
- Alarm follow up by fab equipment

STMicroelectronics - LSIS Laboratory - All rights reserved

Figure 7.2-13 : The TM-based search interface

Regarding the query processing of the TM-based search, the basic idea of the algorithm relies on nested recursive selection of topics according to the first topic found with the user query. When a user expresses his need, the system checks at first in the *topics* of the knowledge base model if there are concepts that match (at least partially). Otherwise it will checks in the *variants*. According to this step, each topic that matches with the user query represents a starting point for the creation of a Topic Map. Hence, for each topic found, the system selects its sub topics and its related associations. This process is repeated while each topic found in the hierarchy has sub topics and associations. The occurrences and the resource types are selected at the end to complete the Topic Map.

The Graphviz[27] application has been integrated in the prototype to allow the display of the Topic Maps in the TM-based search interface. Graphviz is an open source software for graph visualization. It takes as input a specific graph description in a text format and makes types of diagrams as output in several formats (image formats, PDF, SVG, etc.).

We can notice that the Topic-Maps’ paradigm has a specific usage in our application. Topic Maps are usually used to improve information retrieval by exposing the subjects contained in information resources (mainly text documents). In our context, Topic Maps are not content oriented, but rather, business-knowledge oriented.

Finally, the TM-based search functionality is proposed in this prototype for not only resource exploration, but also for facilitating the maintaining of the retrieval system in the company. Indeed, this functionality allows the business experts in the company to explore the impact of any change related to the MI resources on their description and on the whole knowledge model. Thus, any gaps between the resources and the business needs can be detected easily.

---

[27] http://www.graphviz.org/

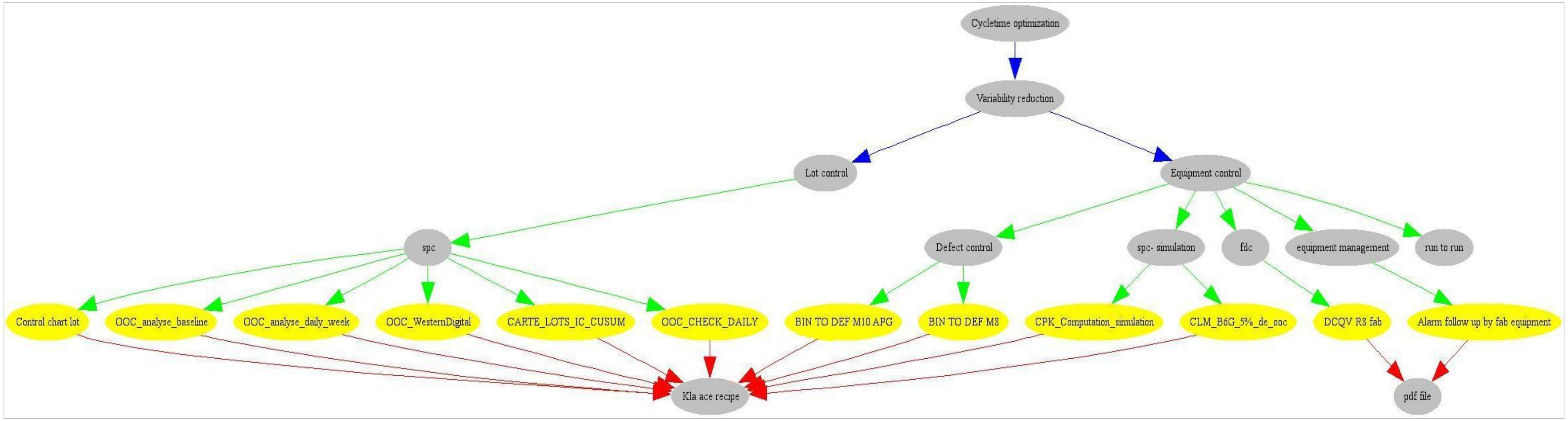
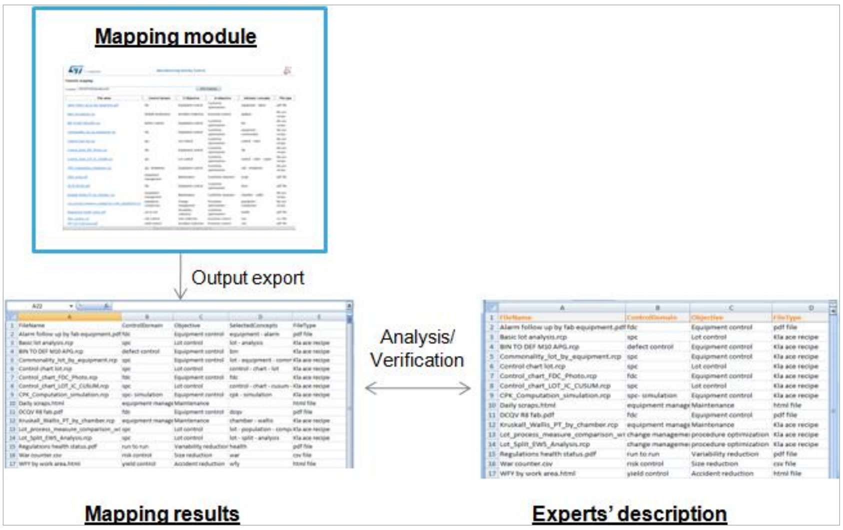


Figure 7.2-14 : Detailed Topic Map

7.3. Experimentations within STMicroelectronics

The bottom-up semantic-descriptors' construction approach has been experimented within STMicroelectronics on a sample of MI resources, properly selected for this purpose. The semantics of this sample and its directory taxonomy have been reviewed slightly by the business experts before the experimentation, so to ensure a fairly representative sample of the control activity. The sample was composed of 384 resources and the expected results were pre-defined with the help of expert engineers in order to assess the results of this bottom-up approach. The validation of the mapping techniques and the semantic structures was done progressively in this way.

The verification technique to assess the results was done manually in csv files. The system automatically exports the results in csv files each time a mapping is done. This output is then thoroughly analyzed and compared with the experts' description (Figure 7.3-1). The aim of the verification step is to evaluate the effectiveness of the mapping techniques, and to check the coherence of the dictionary and the ontology, up to stabilizing their content. The limit of this technique is that it is time consuming. However, in such a context, only the business experts have the necessary knowledge to validate the results.



	A	B	C	D	E
1	FileName	ControlDomain	Objective	SelectedConcepts	FileType
2	Alarm follow up by fab equipment.pdf	fdc	Equipment control	equipment - alarm	pdf file
3	Basic lot analysis.rcp	spc	Lot control	lot - analysis	Kla ace recipe
4	BIN TO DEF M10 APG.rcp	defect control	Equipment control	bin	Kla ace recipe
5	Commonality_lot_by_equipment.rcp	spc	Equipment control	lot - equipment - comm	Kla ace recipe
6	Control chart lot.rcp	spc	Lot control	control - chart - lot	Kla ace recipe
7	Control_chart_FDC_Photo.rcp	fdc	Equipment control	fdc	Kla ace recipe
8	Control_chart_LOT_IC_CUSUM.rcp	spc	Lot control	control - chart - cusum -	Kla ace recipe
9	CPK_Computation_simulation.rcp	spc- simulation	Equipment control	cpk - simulation	Kla ace recipe
10	Daily scraps.html	equipment manage	Maintenance		html file
11	DCQV R8 fab.pdf	fdc	Equipment control	dcqv	pdf file
12	Kruskall_Wallis_PT_by_chamber.rcp	equipment manage	Maintenance	chamber - wallis	Kla ace recipe
13	Lot_process_measure_comparison_wi	spc	Lot control	lot - population - comp	Kla ace recipe
14	Lot_Split_EWS_Analysis.rcp	spc	Lot control	lot - split - analysis	Kla ace recipe
15	Regulations health status.pdf	run to run	Variability reduction	health	pdf file
16	War counter.csv	risk control	Size reduction	war	csv file
17	WFY by work area.html	yield control	Accident reduction	wfy	html file

	A	B	C	D
1	FileName	ControlDomain	Objective	FileType
2	Alarm follow up by fab equipment.pdf	fdc	Equipment control	pdf file
3	Basic lot analysis.rcp	spc	Lot control	Kla ace recipe
4	BIN TO DEF M10 APG.rcp	defect control	Equipment control	Kla ace recipe
5	Commonality_lot_by_equipment.rcp	spc	Equipment control	Kla ace recipe
6	Control chart lot.rcp	spc	Lot control	Kla ace recipe
7	Control_chart_FDC_Photo.rcp	fdc	Equipment control	Kla ace recipe
8	Control_chart_LOT_IC_CUSUM.rcp	spc	Lot control	Kla ace recipe
9	CPK_Computation_simulation.rcp	spc- simulation	Equipment control	Kla ace recipe
10	Daily scraps.html	equipment manage	Maintenance	html file
11	DCQV R8 fab.pdf	fdc	Equipment control	pdf file
12	Kruskall_Wallis_PT_by_chamber.rcp	equipment manage	Maintenance	Kla ace recipe
13	Lot_process_measure_comparison_wi	change manageme	procedure optimization	Kla ace recipe
14	Lot_Split_EWS_Analysis.rcp	change manageme	procedure optimization	Kla ace recipe
15	Regulations health status.pdf	run to run	Variability reduction	pdf file
16	War counter.csv	risk control	Size reduction	csv file
17	WFY by work area.html	yield control	Accident reduction	html file

Figure 7.3-1 : Verification technique

The experimentation of the approach showed good results on the sample but it required some improvements to make on the semantic support (the PC dictionary and the MP ontology), as well as on the mapping process. In fact, to ensure that the mapping techniques work well, it is necessary to have well defined concepts and error free in the MI resources. The semantic support must also be well defined and sufficiently exhaustive. For this reason, we did a set of experiments on the selected sample in order to improve the MP ontology and the PC dictionary while testing the mapping techniques. These experiments were also a way to deduce guidelines for maintaining the MI resources, in order to allow good functioning of the system. The whole experimentation was done in five steps. At each step, the resources and the semantic support were corrected and improved. The five steps are:

- **Step 1:** initial step. The MI resources were used as found within the company
- **Step 2:** grammatical errors and misspellings were removed in this step from the raw vocabulary of the resources

- **Step 3:** a threshold has been set for the similarity measure in order to select a significant similarity ratio between two concepts with the edit-distance measure
- **Step 4:** missed concepts were added in the dictionary and the ontology and others were corrected, according to the results of the steps 1, 2 and 3
- **Step 5:** the redundancy of the key concepts was avoided whenever possible in the dictionary and their number was balanced. Different thresholds were re-tested again.

The findings of the experimentation are presented with the standard Information-Retrieval metrics [Lin 2007]: *Precision*, *Recall* and *F-measure*. The *precision* represents the ratio of the correct results on the retrieved results (total found), and the *recall* represents the ratio of the correct retrieved results on the correct retrieved and non retrieved results (missed).

$$Precision = \frac{\#correct\_results}{\#totalfound}$$

$$Recall = \frac{\#correct\_results}{\#correct + \#missed}$$

The harmonic mean of precision and recall metrics is known as *F-measure*:

$$F - measure = 2\frac{Precision * Recall}{Precision + Recall}$$

Figure 7.3-2 shows the results of the experimentation.

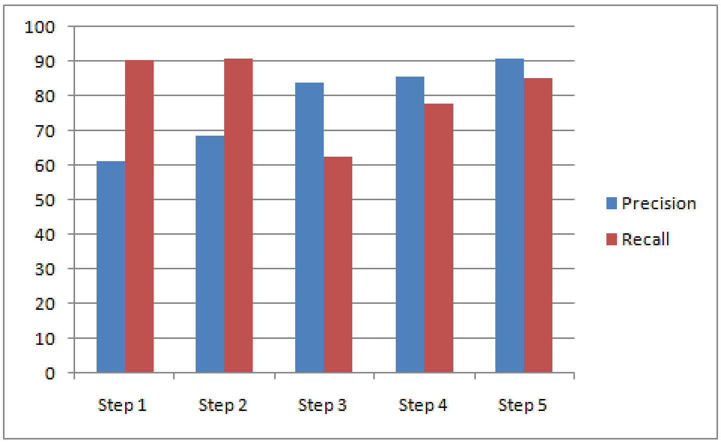


Figure 7.3-2 : Experimentation results with the precision and recall metrics

We can notice a significant improvement of the precision and recall metrics during the five steps, ensuring at the end, high performances with 90% of precision and 85% of recall. The results that must be rather obtained would be to have close ratios between the precision and recall metrics (i.e. an optimal F-Measure).

The result of the first steps was a bit low in term of precision (61%) and many irrelevant results were obtained. The correction of errors in the MI resources increases a little the correct results in step 2 with 68% of precision. On the contrary, the recall values were very high (90%) in step 1 and 2, in particular because there was no threshold set. In fact, the string similarity was considered for all the concepts found, even when it is very low.

The set of the threshold is step 3 inversed the percentage of precision and recall in a positive way. In fact, the setting of an appropriate threshold reduced the number of irrelevant mappings from the results, increasing in this way the value of precision (83%). Inevitably, the number of missed concepts increases, which decreases the value of recall (62%). The threshold was set to 0.7 after several tests. We used different thresholds ranging from 0.5 to 1. The chosen value 0.7 corresponds to the best F-Measure found. Furthermore, in this step, we assessed individually the similarity measures used in the approach, i.e. the dice coefficient, the edit distance, the combined similarity without setting the threshold and the combined similarity with the chosen threshold. Figure 7.3-3 shows the results of this assessment. At first glance, the dice coefficient gives better results (64% of precision) than the edit distance[28] (41% of precision). The edit distance calculates the similarity of characters regardless of their order in the word, and this technique can return many errors comparing the dice coefficient technique. Nonetheless, by combining the two techniques we can obtain better results, and preferably using a given threshold, to reduce the errors. Hence, these findings confirmed our choice of using the combined similarity measure with the threshold.

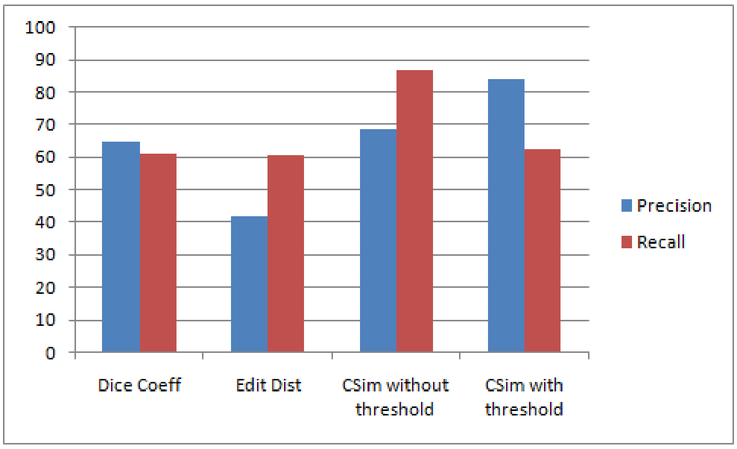


Figure 7.3-3 : Assessment of mapping techniques

We also noticed at step 3 some missed concepts in the dictionary and the ontology, leading to several missed mappings in the results. As we can see in Figure 7.3-2, the recall attains 62% only. By adding missed concepts in the semantic support in step 4, the recall rate increased up to 77%, and the precision rate was maintained at a good rate (85%).

Finally, in step 5, the improvement of the dictionary entries, in particular the key concepts, has enhanced one more time the overall results, because these concepts are used to calculate the similarity measures and they highly orient the results. As reminder, the key concepts in the dictionary enable to set the scope of use of a concept. We noticed that only the key concepts relevant to each defined concept in the dictionary must be used, otherwise there will be too many concepts that could be useless or source of errors (e.g., close syntaxes but different meanings). Hence, by only keeping the relevant key concepts to each concept defined in the dictionary, we were able to achieve 90% of precision and 85% of recall.

According to the last modifications, we chose in the last step to re-tested again different thresholds to confirm our choice. The used thresholds also range from 0.5 to 1, as shown in Table 7.3-1. In fact, we consider that the similarity between two concepts must be considered when it at least exceeds the half of the matching rate (0.5). Otherwise, the similarity could be low. The optimal threshold corresponds to the best ratio between the precision and recall, corresponding to the best F-Measure. The final experimentation showed that the threshold 0.7 must be maintained.

---

[28] The edit distance here were tested alone with the threshold

Threshold	Precision	Recall	F-Measure
0.5	0.702	0.926	0.798
0.6	0.829	0.891	0.858
0.7	0.909	0.853	0.880
0.8	0.910	0.716	0.801
0.9	0.927	0.684	0.787
1.00	0.953	0.652	0.774

Table 7.3-1 : Experimentation results using different thresholds in step 5

Finally, these first experimentations were done to test the mapping techniques and assess the consistency of the semantic structures. The used panel of resources was selected with the help of the process control experts. Hence, we tried again to experiment the mapping technique on other resources selected as found in the company, in order to compare the results. Three different samples of resources were used:

- the first comprises 232 resources related to a statistical analysis tool called Kla Ace XP. The vocabulary used in the resources and the used taxonomies were well defined by the business experts
- the second sample comprises 400 resources produced with the Business-Object (BO) reporting tool. The used vocabulary was controlled for some resources only
- the third sample was composed of 610 resources from a shared disk (called PC-System) where various MI resources are usually stored. The used vocabulary was spontaneous and not error free.

Figure 7.3-4 depicts the results of the experimentation. The first sample (Kla Ace) that has a controlled vocabulary got good results with 88% of precision and 89% of recall. The second sample (BO) got interesting results with average of 76% of precision and 70% of recall, but less interesting than the first sample. Finally, the third sample got the lower results with 56% of precision and 60% of recall. We can estimate that such results are quite effective, considering the variety of the used vocabulary (controlled, spontaneous). Nonetheless, it is preferable to have a controlled vocabulary (as best as possible) in order to ensure reliable results.

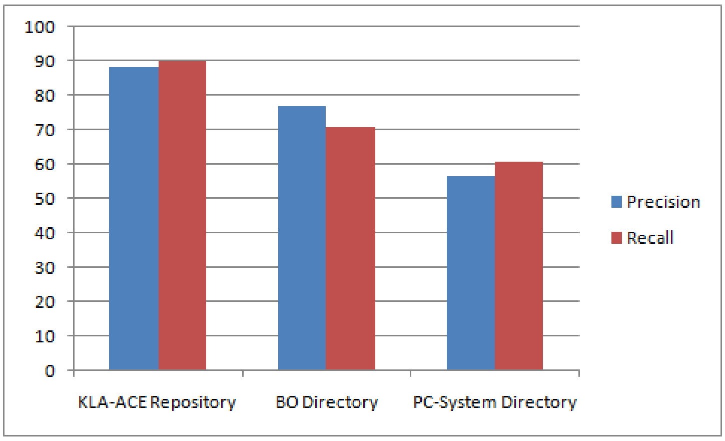


Figure 7.3-4 : Experimentation on various STMicroelectronics' resources

These experimentations give an idea about how the system must be supported to continuously obtain significant results. Basically, we can say that avoiding errors in the MI resources and well defining the required vocabulary in the semantic support are the key points to take care of.

Finally, regarding the pattern-based search, this functionality is still under test today within the company. The main difficulty in testing the top-down approach is to set up a set of pre-defined scenarios of needs with their solutions. This task requires not only the implication of several business actors in order to identify a set of representative needs, but also to know in advance the solutions to these needs, in order to assess the precision of the results.

7.1. Conclusion

We presented in this chapter an overview of the prototype developed for the STMicroelectronics Company to improve the description and retrieval of the resources used to support the control of its manufacturing process. We seek through this application offering an easy and common semantic access to these resources, which are often dispatched in the company network. Three search functions are proposed. With the mapping function the users (process control experts and business engineers) can explore existing MI resources in the company and associate to them semantic description. Using the pattern-based search function, the users can be assisted in expressing their business needs and in retrieving the relevant resources to their high-level business needs. They can also explore the MI resources of the company with the help of knowledge maps dynamically created with the resource description.

We chose a knowledge base model based on the Topic-Maps' paradigm for its suitability in handling any type of knowledge with high expressiveness. It also gives a generic aspect to the knowledge base and to the type of semantics that the S3 approach handles. The proposed Topic-Maps-based application can then be easily maintained and adapted for eventual changes in the company. However, the implementation choices of the S3 approach were dependent on the STMicroelectronics' IT requirements. Other implementation choices could be interesting to consider, such as the OWL language for the implementation of the ontology which is more suitable for the formalization of the concepts of a domain and for expressing inferences. A service-oriented architecture could be also worthwhile to consider for the implementation of the alignment patterns with the pattern-based search.

Finally, the experimentation of the retrieval approaches showed promising findings within STMicroelectronics. The main outlines of the S3 approach were globally agreed by the company, but the prototype is still in the test phase carried by the process control experts.

General Conclusion and Future Work

1. **Contribution Synthesis:**

This thesis proposed a semantic approach, named S3, for resource description and retrieval for manufacturing-process control needs. Many research works assert the importance of semantics nowadays for the exchange of information in industrial-related applications. However, the most difficult task in using semantics in industrial contexts is, on one hand, to define the semantic structures that will suitably meet the purpose of work, and on other hand, to consider the business needs of the business actors during the search process. Comparing to existing information retrieval and component retrieval approaches, the S3 approach presents three original assets:

- the first particular asset resides in the business consistency of the ontology carried by four standard views of description, where the control view is semantically consolidated with the process control dictionary. These views semantically converge to business needs in a process, and rationalize, by the way, the description of the process control
- the second asset of the approach relies on the retrieval strategies in which the semantic structures are used. The approach proposes two novel retrieval strategies which address complementary needs: associating semantics to MI resources and enhancing their retrieval. While the bottom-up strategy provides the usage of the MI resources in a business process, the top-down strategy completes this elementary description with the business needs of the process actors through the goal-oriented aspect of the pattern-based search. Moreover, by using the search patterns as proposed, we achieve in fact an alignment between the users' needs and the resources, because a pattern carries the expression of a business need up to its satisfaction
- The third particular asset relies on the originality of the alignment pattern base. By storing business need artifacts associated to MI-resource description, we progressively capitalize on business know-how related to business data and to the process control experience in the company.

The first experimentations of the approach within STMicroelectronics enabled to stabilize the content of the proposed semantic structures and the techniques of the bottom-up strategy. The findings of this work also showed that the retrieval strategies of the S3 approach offer different means for information retrieval that can better correspond to the industrial needs.

Regarding the STMicroelectronics' needs, this approach is intended to help the company engineers in exploring means of manufacturing-process control by exploring which resource is more adequate to a given business need. Actually, the company expects to capitalize on the used resources in the manufacturing-process control through this work, because the resulting retrieval system offers to the engineers a single platform for the exchange of manufacturing information independently of the used COTS platforms in the company.

2. **Future work:**

The main purpose of this thesis was to address the research issues taking into account the requirements of the STMicroelectronics Company for which this work should apply. However, some aspects of the S3 approach can be improved and some other tracks should be investigated, in particular because the approach as proposed presents some limits.

- The main limit of this work is the lack of automatic adaptation of the resource description to changes. In fact, the top-down retrieval strategy is dependent on the bottom-up retrieval strategy, mainly because of the lack of semantics in the resources. The bottom-up technique must be applied each time there are changes in the resources, so to update the semantic descriptors. Otherwise the results of the top-down retrieval approach will not be satisfying. One of the solutions to this problem would be to set up a checking system in the company to automatically detect changes in the resource description to update the semantic descriptors afterwards. Nevertheless, such solution remains specific to the company software environment. The optimal solution should tackle the source of the problem, which is the lack of semantics in the resources. In fact, up to now, the MI resources are created without description in the COTS platforms of the company. Thus, adding an annotation functionality in the prototype where each new MI resource can be properly described and annotated following a resource description meta-model would be a better solution in this case, to manage the descriptions and changes of all company resources. This track was studied during this research work and is considered today in future work.
- The general characteristics of the approach rely on the consistency of the MP ontology which plays a key role in consolidating the link between the users' needs and the MI resources. We can denote that this ontology has two particularities: it coherently combines a goal-oriented description with the description of a business activity, and it embeds two business knowledge domains which are the manufacturing process and the process control. The importance of the MP ontology in our approach shows us how much such kind of business ontologies must be well defined and built. The construction of the MP ontology was one of the main difficult tasks in this research work, because of the particularity of the studied industrial context with respect to our purposes of resource description and retrieval. Also, main existing ontology construction approaches are not yet quite mature and conclusive. We estimate that it would be worthwhile to investigate new approaches for the construction of such specific business ontologies that need to embed business needs with two interrelated business domains.
- The meta-model of the semantic descriptors of resources can be a bit more enriched with software description. In the approach, this meta-model contains business concepts related to the usage of the resources. This model could be extended to include some software aspects such as the used COTS platforms with which the resources were created, the used data sources to produce manufacturing information, the resource formats, etc. A software description can be helpful in this case in discriminating some resources during their retrieval by the process control actors. However, a software ontology will be necessary in this case to facilitate the inclusion of a software description in the approach.

- The pattern-based search needs several experimentations with the process control actors in the company to assess the effectiveness of the top-down retrieval approach in targeting the resources relevant to the business needs. Up to now, the prototype is still under test. Furthermore, it is also planned to experiment the *CSim* measure for matching the keywords of the users' queries with the semantic descriptors of the resources. This measure proved its efficiency in the bottom-up strategy, so it would be worth to reuse it in the top-down retrieval. Ultimately, the overall improvement of the MI-resource retrieval in the company can be effectively observed and evaluated over the long term.

Finally, we can notice through the study of the process control of the semi-conductor industry that the process control domain encompasses several standard and interesting methods that can be applied to any business process or activity that needs to be controlled. The general structure of the business ontology and the domain-specific dictionary of the S3 approach can be easily reused for other process control purposes. Also, the business-need-oriented description of the resources constitutes an important aspect that needs to be ever more considered, not only for resource retrieval, but also for the process of MI-resource creation in industries.

Bibliography

ALI, R., DALPIAZ, F., AND GIORGINI, P. 2010. A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering 15*, 4, 439–458.

ALNUSAIR, A. AND ZHAO, T. 2010. Component Search and Reuse : An Ontology-based Approach. *Knowledge Creation Diffusion Utilization*, 258–261.

ARDELT, M. 2004. Wisdom as Expert Knowledge System: A Critical Review of a Contemporary Operationalization of an Ancient Concept. *Human Development 47*, 5, 257–285.

ARSANJANI, A. 2005. *Service-oriented modeling and architecture*.

BAADER, F., MCGUINNESS, D., NARDI, D., AND PATEL-SCHNEIDER, P. 2003. *THE DESCRIPTION LOGIC HANDBOOK : Theory , implementation , and applications*. Cambridge university press.

BAAZAOUI, H., AUFAURE, M.-A., AND BEN MUSTAPHA, N. 2007. A model-driven approach of ontological components for on- line semantic web information retrieval. *Journal of Web Engineering 6*, 4, 303–329.

BELKIN, N.J. AND CROFT, W.B. 1987. Retrieval Techniques. In: M.E. Williams, ed., *Annual Review of Information Science and Technoloy (ARIST)*. Elsevier Science Publishers B.V., 109 – 145.

BERNSTEIN, A. AND KLEIN, M. 2004. Towards High-Precision Service Retrieval The Challenge : High Precision Service Retrieval The State of the Art. *Internet Computing 1*, IEEE 8, 30 – 36.

BIANCHINI, D., CAPPIELLO, C., ANTONELLIS, V. DE, AND PERNICI, B. P2S : a methodology to enable inter-organizational process design through Web Services.

BLEISTEIN, S.J., COX, K., AND RAY, P.K. 2004a. Strategy-Oriented Alignment in Requirements Engineering : Linking Business Strategy to Requirements of e-Business Systems using the SOARE Approach. *Practice 36*, 4, 259–276.

BLEISTEIN, S.J., COX, K., AND VERNER, J. 2004b. Integrating Jackson Problem Diagrams with Goal Modeling and Business Process Modeling in e-Business Systems Requirements Analysis. *In Proceedings of the 5th International Workshop on Conceptual Modeling Approaches for e-Business (eCOMO 2004),* Springer.

BLEISTEIN, S.J., COX, K., AND VERNER, J. 2005. Strategic Alignment in Requirements Analysis for Organizational IT : an Integrated Approach. *ACM Symposium on Applied Computing*, 1300–1307.

BLEISTEIN, S.J., COX, K., VERNER, J., AND PHALP, K.T. 2006. B-SCP : a requirements analysis framework for validating strategic alignment of organizational IT based on strategy , context , and process 1 Introduction. *Information and Software Technology 48*, 9, 846–868.

BOUZID, S., CAUVET, C., FRYDMAN, C., AND PINATON, J. 2013a. A Semantic Support to Improve the Collaborative Control of Manufacturing Processes in Industries. *17th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2013),* IEEE.

BOUZID, S., CAUVET, C., FRYDMAN, C., AND PINATON, J. 2013b. A Semantic Mapping Approach to Retrieve Manufacturing Information Resources. *The IEEE IFAC-MIM Conference*, IEEE Xplore.

BOUZID, S., CAUVET, C., FRYDMAN, C., AND PINATON, J. 2013c. A Bottom up Search Technique of Manufacturing Indicators. *24th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC 2013),* IEEE.

BOUZID, S., CAUVET, C., AND PINATON, J. 2012a. A Survey of Semantic Web Standards to Representing Knowledge in Problem Solving Situations. *CAMP12*, IEEE, 121–125.

BOUZID, S., CAUVET, C., AND PINATON, J. 2012b. A Topic-Map-based Framework for Resource Retrieval in an Industrial Context: STMicroelectronics ' Case Study. *Proceedings of the International Conference on Systems Engineering and Engineering Management (ICSEEM'12)*.

BRACHMAN, R.J. AND LEVESQUE, H.J. 2003. *Knowledge Representation and Reasoning*.

BRG. 2010. *The Business Motivation Model*.

BRISCOE, T. AND CARROLL, J. 2002. Robust Accurate Statistical Annotation of General Text. *In Proceedings of the 3rd International Conference on Language Resources and Evaluation*, 1499–1504.

BUDAK ARPINAR, I., ALEMAN-MEZA, B., ZHANG, R., AND MADUKO, A. 2004. Ontology-driven Web services composition platform. *Proceedings. IEEE International Conference on e-Commerce Technology, CEC 2004. 3*, 2, 146–152.

BURGOS, J.L.M. 2011. Semantic Web Standards. *SNET Computer Engineering*.

CAHIER, J. 2005. Ontologies sémiotiques pour le web socio-sémantique.

CAUSSANEL, J. 2000. Contribution à l ' étude des Systèmes de Capitalisation des Connaissances : SMOKC , un système dédié aux PME-PMI Remerciements. *Sciences-New York*, 1–249.

CHAPMAN, S. 2008. Sam ' s String Metrics. *String Similarity Metrics for Information Integration*.

CHARDONNET, A. AND THIBAUDON, D. 2003. Le guide du PDCA de Deming. In: *Progrès Continu et Management*. 52–72.

CHEN, H.-M. 2008. Towards Service Engineering : Service Orientation and Business-IT Alignment. *Proceedings of the 41st Hawaii International Conference on System Sciences*, IEEE, 1530–1605.

CHU, H. 2003. Information Representation and Retrieval : An Overview. In: I. Information Today, ed., *Information Representation and Retrieval: An Overview*. 1–25.

COHEN, W.W. AND FIENBERG, S.E. 2003. A Comparison of String Distance Metrics for Name-Matching Tasks. *the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, 73–78.

CORBY, O., DIENG-KUNTZ, R., AND FARON-ZUCKER, C. 2004. Querying the Semantic Web with Corese Search Engine. *In the International Conference on Electronics, Computers and Artificial Intelligence (ECAI) proceedings*, 705.

CRUZ, I.F. 2002. *DAML-S: Semantic Markup for Web Services*.

DELGADO, A., RUIZ, F., AND GUZMÁN, I.G. DE. 2010. A Model-driven and Service-oriented framework for the business process improvement. *Journal of Systems Integration* Mdd, 45–55.

DENAYER, M. 2004. Decouverte de Service : Etude d' Approche Syntaxique et Semantique. *Technical report, Bioinformatics Grid Ressources and Environments*.

DICHEV, C., DICHEVA, D., AND AROYO, L. 2004. Using Topic Maps for Web-based Education. *Advanced Technology for Learning 1*, 1.

DING, Y., STOLLBERG, M., AND FENSEL, D. 2001. SEMANTIC WEB LANGUAGES – STRENGTHS AND WEAKNESS. *Language*.

DOU, D. AND MCDERMOTT, D. 2006. Deriving Axioms Across Ontologies ∗. *AAMAS'06*, ACM.

ELLOUZE, N. 2010. Approche de recherche intelligente fondée sur le modèle des Topic Maps Application au domaine de la construction durable. *Architecture*.

ETIEN, A. 2006. Ingénierie de l ' alignement : Concepts , Modèles et Processus. *Structure*.

EUZENAT, J. 2007a. Ontology matching.

EUZENAT, J. 2007b. *Ontology matching*. Springer-Verlag, Berlin Heidelberg.

FERDIAN. 2001. *A Comparison of Event-driven Process Chains and UML Activity Diagram for Denoting Business Processes*.

FERNANDEZ, M., GOMEZ-PEREZ, A., AND JURISTO, N. 1997. *METHONTOLOGY : From Ontological Art Towards Ontological Engineering*.

GOLO, G. 2011. *Advanced Process Control : Nice to have or vital to have*.

GOMEZ, J.M., RICO, M., TOMA, I., AND HAN, S. 2006. GODO : Goal Oriented Discovery for Semantic Web. *5th International Semantic Web Conference*, Springer.

GRAUBMANN, P. AND ROSHCHIN, M. 2006. Semantic Annotation of Software Components. *32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO'06) 2*, 46–53.

GRUBER, T.R. 1993. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Knowledge Creation Diffusion Utilization*, 907–928.

GRUNINGER, M. AND FOX, M.S. 1995. Methodology for the Design and Evaluation of Ontologies. *Proc. Int'l Joint Conf. AI Workshop on Basic Ontological Issues in Knowledge Sharing*.

GUARINO, N. AND WELTY, C.A. 2009. *An Overview of OntoClean*. Springer.

GUITTARD, C., ZAHER, L.H., AND CAHIER, J. 2005. Experimentation of a socially constructed " Topic Map " by the OSS community. *In proceedings of the IJCAI-05 workshop on Knowledge Management and Ontology Management (KMOM)*.

HAI, D.H. 2005. SCHEMA MATCHING AND MAPPING-BASED DATA INTEGRATION.

HAKI, M.K. AND FORTE, M.W. 2010. Service-Oriented Business-IT Alignment: a SOA Governance Model. *INTERNATIONAL JOURNAL ON Advances in Information Sciences and Service Sciences 2*, 2, 51–60.

HENDERSON, J.C. AND VENKATRAMAN, N. 1999. Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal 38*, 2 & 3, 472–484.

HUBERT, G. 2010. *Recherche d'Information et Contexte (HDR)*. Toulouse, France.

HYMPHREYS, B.L. AND LINDBERG, D.A.B. 1993. The UMLS ® project : making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association 81*, 2, 170.

IBM. 2009. Architecture a trois niveaux. *WebSphere*. http://pic.dhe.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/covr_3-tier.html.

JACKSON, M.A. 1999. Problem Analysis Using Small Problem Frames. *South African Computer Journal - Special Issue on WOFACS'98 22*, 47–60.

KALFOGLOU, Y. AND SCHORLEMMER, M. 2003. Ontology mapping: the state of the art. *The Knowledge Engineering Review 18*, 1, 1–31.

KANNAN, R. 2010. Topic Map : An Ontology Framework for Information Retrieval. *Knowledge Management*.

KHELIF, K., DIENG-KUNTZ, R., AND BARBRY, P. 2007. An Ontology-based Approach to Support Text Mining and Information Retrieval in the Biological Domain. *Journal of universal computer science 13*, 12, 1881–1907.

KÁSLER, L., VENCZEL, Z., AND VARGA, L.Z. 2006. Framework for Semi Automatically Generating Topic Maps. *Proceedings of the 3rd international workshop on text-based information retrieval*, 24 – 30.

LAMSWEERDE, A. VAN. 2001. Goal-Oriented Requirements Engineering : A Guided Tour. *Proceedings Fifth IEEE International Symposium on requirements Engineering*, IEEE, 249–263.

LAVIK, S. AND NORDENG, T.W. 2004. Brainbank learning – building topic maps-based e-portfolios. *XML 2004 Proceedings*.

LI, C. 2012. A Holistic Semantic Based Approach to Component Specification and Retrieval.

LI, S. AND QIAO, L. 2012. Ontology-based Modeling of Manufacturing Information and its Semantic Retrieval. *Proceedings of the 16th International Conference on Computer Supported Cooperative Work in Design*, 540–545.

LI, W., GUO, Y., LIAO, W., AND HANG, R. 2008a. Research on Ontology Component Description Model Based on the Semantic Web. *2008 IEEE Asia-Pacific Services Computing Conference* 0626120, 697–702.

LI, Z., RASKIN, V., AND RAMANI, K. 2007. A Methodology of Engineering Ontology Development for Information Retrieval. *International Conference on Engineering Design, ICED'07*, 1–12.

LI, Z., RASKIN, V., AND RAMANI, K. 2008b. Developing Engineering Ontology for Information Retrieval. *Journal of Computing and Information Science in Engineering 8*, 1, 1–13.

LI, Z., YANG, M.C., AND RAMANI, K. 2008c. A methodology for engineering ontology acquisition and validation. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 23*, 01, 37.

LIN, D. 1998. An Information-Theoretic Definition of Similarity. *The 15th international conference on Machine Learning*, 296–304.

LIN, F. 2007. *State of the Art : Automatic Ontology Matching*. Jonkoping, Sweden.

LIU, W., HE, K., AND LIU, W. 2005. Design and realization of ebXML registry classification model based on ontology. *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, Ieee, 809–814 Vol. 2.

LUCRÉDITO, D., DO PRADO, A.F., AND SANTANA DE ALMEIDA, E. 2005. A Survey on Software Components Search and Retrieval. *The 30th EUROMICRO Conference*.

MAS, S. AND MARLEAU, Y. 2009. Proposition of a Faceted Classification Model to Support Corporate Information Organization and Digital Records Management. *The 42nd Hawaii International Conference on System Sciences*, 1–10.

MAS, S. AND ZACKLAD, M. 2008. Classification à facettes et modèles à base de points de vue : Différences et complémentarité. *36e congrès annuel de l'Association canadienne des sciences de l'information (ACSI)*, 1–10.

MCMAHON, C., LOWE, A., CULLEY, S., ET AL. 2004. Waypoint: An Integrated Search and Retrieval System for Engineering Documents. *Journal of Computing and Information Science in Engineering 4*, 4, 329.

MELLAL, N. 2007. Réalisation de l ' interopérabilité sémantique des systèmes , basée sur les ontologies et les flux d ' information. *Flux*.

MENDLING, J. AND NÜTTGENS, M. 2005. *EPC Markup Language (EPML) An XML-Based Interchange Format for Event-Driven Process Chains (EPC)*. Vienna, Austria.

MIRBEL, I. AND CRESCENZO, P. 2009. Des besoins des utilisateurs à la recherche des service web: une approche sémantique guidée par les intentions. *Revue Ingénierie des systèmes d'information, RTSI Série ISI (Hermes) 15(4)*, 0.

MIRBEL, I. AND CRESCENZO, P. 2010. From End-User ' s Requirements to Web Services Retrieval : a Semantic and Intention-Driven Approach. *First International Conference on Exploring Services Sciences*.

MOORE, A. 2000. Semantics - meanings, etymology and the lexicon. http://www.universalteacher.org.uk/lang/semantics.htm.

MORISIO, M., SEAMAN, C.B., BASILI, V.R., ET AL. 2002. COTS-Based Software Development : Processes and Open Issues. *Journal of Systems and Software (elsevier) 61*, 189–199.

NIANFANG, X., XIAOHUI, Y., AND XINKE, L. 2010. Software Components Description Based on Ontology. *2010 Second International Conference on Computer Modeling and Simulation*, 423–426.

NOY, N.F. 2004. Semantic Integration : A Survey Of Ontology-Based Approaches. *33*, 4, 65–70.

OUZIRI, M. 2006. Semantic integration of Web-based learning resources : A Topic Maps-based approach. *Learning*, 0–4.

PACONTROL. 2006. *Instrumentation & Control*.

PARSIA, B. AND SIRIN, E. 2000. Pellet : An OWL DL Reasoner.

PENG, X. AND ZHAO, W. 2007. An Incremental and FCA-Based Ontology Construction Method for Semantics-Based Component Retrieval. *Seventh International Conference on Quality Software (QSIC 2007)* Qsic, 309–315.

PENG, Y., PENG, C., HUANG, J., AND HUANG, K. 2009. An Ontology-Driven Paradigm for Component Representation and Retrieval. *Ninth IEEE International Conference on Computer and Information Technology*, Ieee, 187–192.

PEPPER, S. 2008. Topic Maps and All That. http://topicmaps.wordpress.com/2008/05/11/topic-maps-and-the-semantic-web/.

PEPPER, S. 2010. Topic Maps. *Encyclopedia of Library and Information sciences*.

PRAPHAMONTRIPONG, U. AND HU, G. 2004. XML-Based Software Component Retrieval with Partial and Reference Matching. *October*.

PRIETO-DIAZ, R. 1991. Implementing Faceted Classification for Software Reuse. *Communications of the ACM 34(5)*, 88–97.

QUAN, L., XINJUAN, J., AND YIHONG, L. 2007. Research on Ontology-based Representation and Retrieval of Components. *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, Ieee, 494–499.

REGEV, G. AND WEGMANN, A. 2004. Remaining Fit : On the Creation and Maintenance of Fit. *Proceedings of BPMDS 4*.

RISTAD, E.S. AND YIANILOS, P.N. 1998. Learning String Edit Distance. *Learning string-edit distance." Pattern Analysis and Machine Intelligence, IEEE Transactions 5*, 20, 522–532.

ROLLAND, C. 2003. Fitting Business Models to System Functionality : Alignment issues and challenges. *Proceedings of the 15th Conference on Advanced Information Systems Engineering (CAiSE'03)*.

SCHECHTER, J. AND ENOCH, D. 2006. *Meaning and Justification: The Case of Modus Ponens*.

SCHEER, A. AND NÜTTGENS, M. 2000. ARIS Architecture and Reference Models for Business Process Management. *Business Process Management*, LNCS 1806, 376–389.

SCHIPPERS, W.A.J. 2000. Structure and applicability of quality tools.

SCHNEIDER, D.K. AND SYNTETA, V. 2005. *Introduction aux Topic Maps*.

SHAO, Y. AND ZHANG, M. 2010. Research on Decision Tree in Component Retrieval. *Science And Technology* Fskd, 2290–2293.

SIMONIN, J., BERTIN, E., TRAON, Y. LE, ET AL. 2010. Business and Information System Alignment : a Formal Solution for Telecom Services. *Fifth International Conference on Software Engineering Advances (ICSEA)*, IEEE, 278–283.

SIRIN, E., PARSIA, B., AND HENDLER, J. 2004. Composition-driven Filtering and Selection of Semantic Web Services. *In AAAI Spring Symposium on Semantic Web Services*, 129–138.

SMULLYAN, R.M. 1995a. First-Order Logic. In: *First-order logic*. Dover Publications, 146 – 255.

SMULLYAN, R.M. 1995b. Propositional logic. In: Dover Publications, 28 – 113.

SRIDHARAN, B., DENG, H., CORBITT, B., AND INFORMATION, B. 2009. AN ONTOLOGY-DRIVEN TOPIC MAPPING APPROACH TO MULTI-LEVEL MANAGEMENT OF E-LEARNING RESOURCES. *Europeen Conference on Information System (ECIS)*.

SUBHASHINI, R. AND AKILANDESWARI, J. 2011. A SURVEY ON ONTOLOGY CONSTRUCTION METHODOLOGIES. *International Journal of Enterprise Computing and Business Systems 1*, 1.

SURE, Y., STAAB, S., AND STUDER, R. 2004. On-To-Knowledge Methodology (OTKM).

SYCARA, K., WIDOFF, S., KLUSCH, M., AND LU, J. 2002. Larks : Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace ∗. *Autonomous Agents and Multi-Agent systems*, 173–203.

THEVENET, L.-H. 2009. Proposition d ' une modélisation conceptuelle d ' alignement stratégique : La méthode INSTAL.

TIXIER, B. 2001. *La problématique de la gestion des connaissances Le cas d ' une entreprise de développement informatique bancaire*. Nantes, France.

TSAI, W.T., XIAO, B., PAUL, R.A., AND CHEN, Y. 2006. Consumer-Centric Service-Oriented Architecture: A New Approach. *The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06)*, IEEE Computer Society, 175–180.

USCHOLD, M. AND GRUNINGER, M. 1996. Ontologies : Principles , Methods and Applications. *Knowledge Engineering Review 11*, 2, 1 – 69.

VAZEY, M. AND RICHARDS, D. 2006. Evaluation of the FastFIX Prototype 5Cs CARD System. *Advances in Knowledge Acquisition and Management*, Springer, 108–119.

VERES, C., SAMPSON, J., BLEISTEIN, S.J., COX, K., AND VERNER, J. 2009. Using Semantic Technologies to Enhance a Requirements Engineering Approach for Alignment of IT with Business Strategy. *2009 International Conference on Complex, Intelligent and Software Intensive Systems*, 469–474.

VISSER, U., STUCKENSCHMIDT, H., SCHUSTER, G., NEUMANN, H., AND H, S. 2001. Ontology-Based Integration of Information — A Survey of Existing Approaches. *In IJCAI-01 workshop: ontologies and information sharing*, 108–117.

WEGMANN, A., BALABKO, P., LÊ, L.-S., REGEV, G., AND RYCHKOVA, I. 2005a. A Method and Tool for Business-IT Alignment in Enterprise Architecture. *In Proceedings of the CAiSE - International Conference on Advanced Information Systems Engineering*, 113–118.

WEGMANN, A., REGEV, G., AND LOISON, B. 2005b. Business and IT Alignment with SEAM. *REBINTA Worshop - 14th International Requirement Engineering Conference*, IEEE.

WEGMANN, A., REGEV, G., RYCHKOVA, I., JULIA, P., AND PERROUD, O. 2007. Early Requirements and Business-IT Alignment with SEAM for Business. *15th IEEE International Requirements Engineering Conference (RE 2007)* 3, 111–114.

WHEELER, D.. AND CHAMBERS, D.. 1992. *Understanding Statistical Process Control*.

WIERINGA, R.J., BLANKEN, H.M., FOKKINGA, M.M., AND GREFEN, P.W.P.. 2003. Aligning Application Architecture to the Business Context. *In Advanced Information Systems Engineering, CAiSE'2003*, Springer Berlin/Heidelberg, 1028–1029.

YAN, W. 2010. Component Retrieval Based on Ontology and Graph. *Journal of Information & Computational Science 4*, 893–900.

YANG, X. 2001. Ontologies and How to Build Them 1 Introduction 2 What is an Ontology ? *World*, 1–21.

YAO, Y., LIN, L., AND DONG, J. 2009. Research on Ontology-Based Multi-source Engineering Information Retrieval in Integrated Environment of Enterprise. *International Conference on Interoperability for Enterprise Software and Applications*, Ieee, 277–282.

ZACHMAN, J.A. 2003. *The Zachman Framework For Enterprise Architecture*.

ZACKLAD, M. 2007. Classification, Thésaurus, ontologies, folksonomies: comparaisons du point de vue de la recherche ouverte d ' information (ROI). *35eme Congrès annuel de l'association canadienne des sciences de l'information*, 1–15.

ZACKLAD, M., CAUSSANEL, J., AND CAHIER, J. 2002. Proposition d ' un méta-modèle basé sur les Topic Map pour la structuration et la recherche d ' information. *Actes des Journées du Web Sémantique*.

ZAHER, L.H., CAHIER, J., AND GUITTARD, C. 2008. *Cooperative Building of Multiple Point-of-View Topic Maps with Hypertopic*.

ZAHER, L.H., CAHIER, J., ZACKLAD, M., AND DELAUNAY, I.C. 2006. The Agoræ / Hypertopic approach. *International Workshop IKHS - Indexing and knowledge in Human Sciences (Sdc)*.

ZAHER, L.H., CAHIER, J., ZACKLAD, M., AND DELAUNAY, T.I.C. 2007. De la recherche d ' information à une recherche ouverte d ' information. *Recherche*.

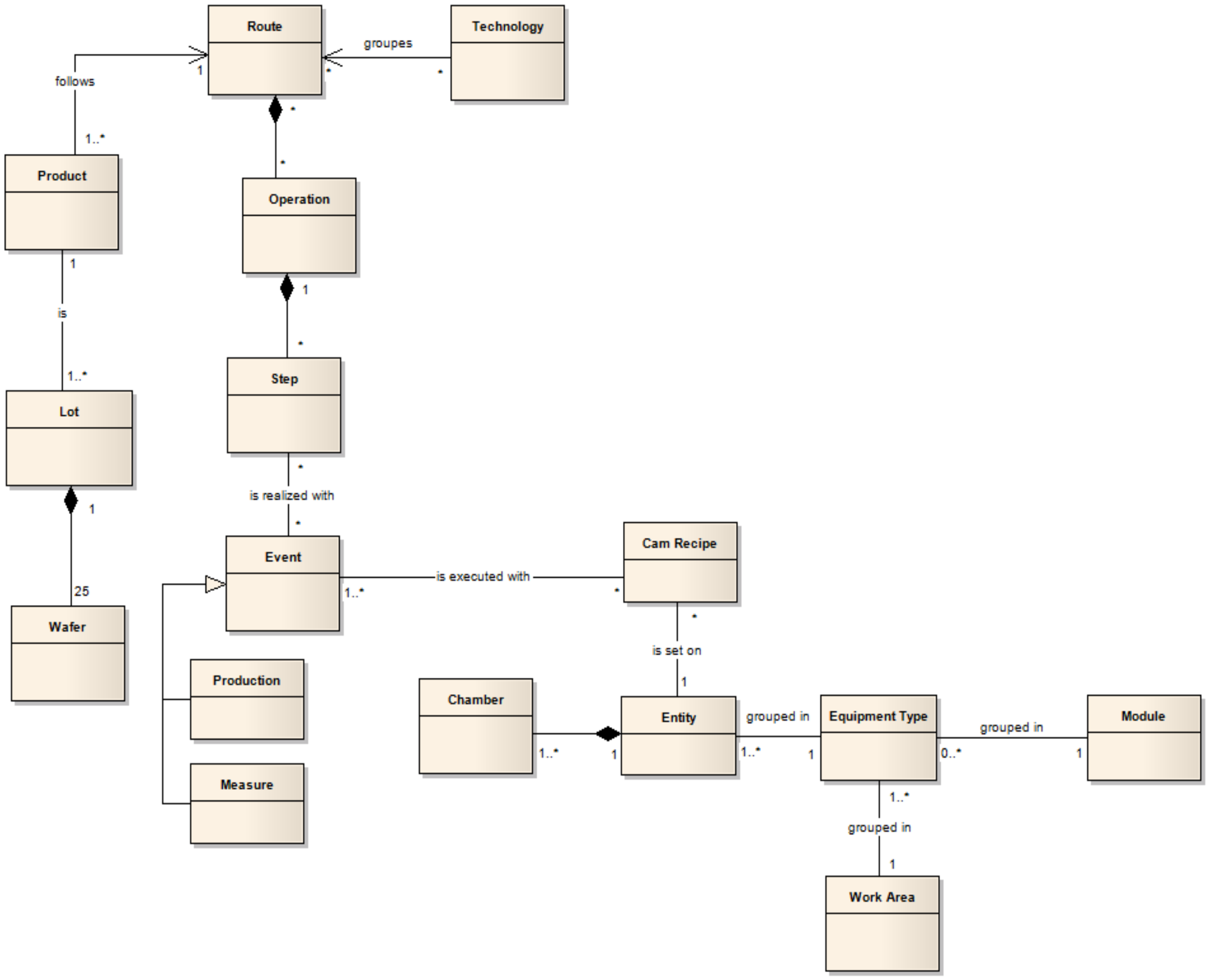
ZHANG, X. AND LI, W. 2008. Ontology-Based Semantic Retrieval System. *2008 4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, Ieee, 1–4.

ZHONG, Q., LI, H., LI, J., ET AL. 2009. A gauss function based approach for unbalanced ontology matching. *Proceedings of the 35th SIGMOD international conference on Management of data - SIGMOD '09*, ACM Press.

Appendices

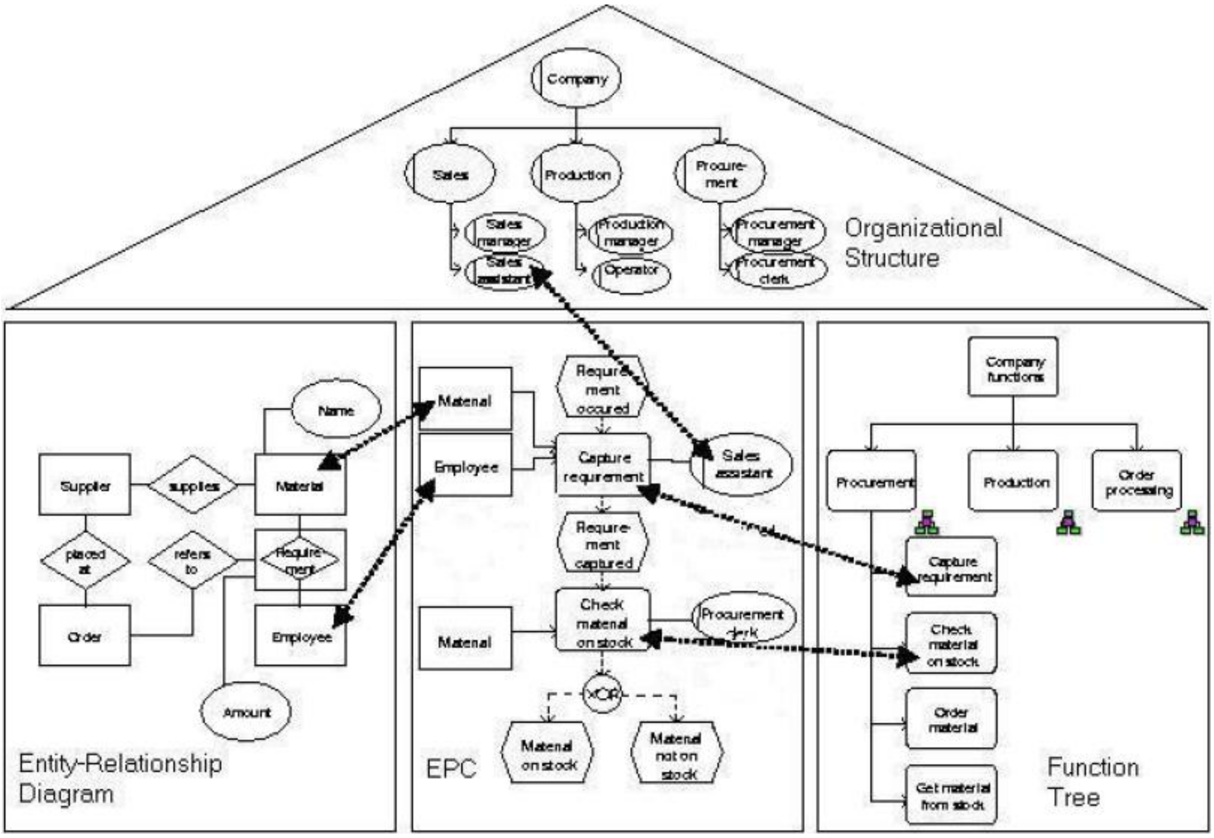
Appendix A : The UML Class Diagram of STMicroelectronics' Manufacturing-Entities

This appendix presents the UML class diagram of the entities related to the manufacturing activity of STMicroelectronics. This diagram shows the main entities used in the data view of the MP ontology (cf. section 4.2.2.2).



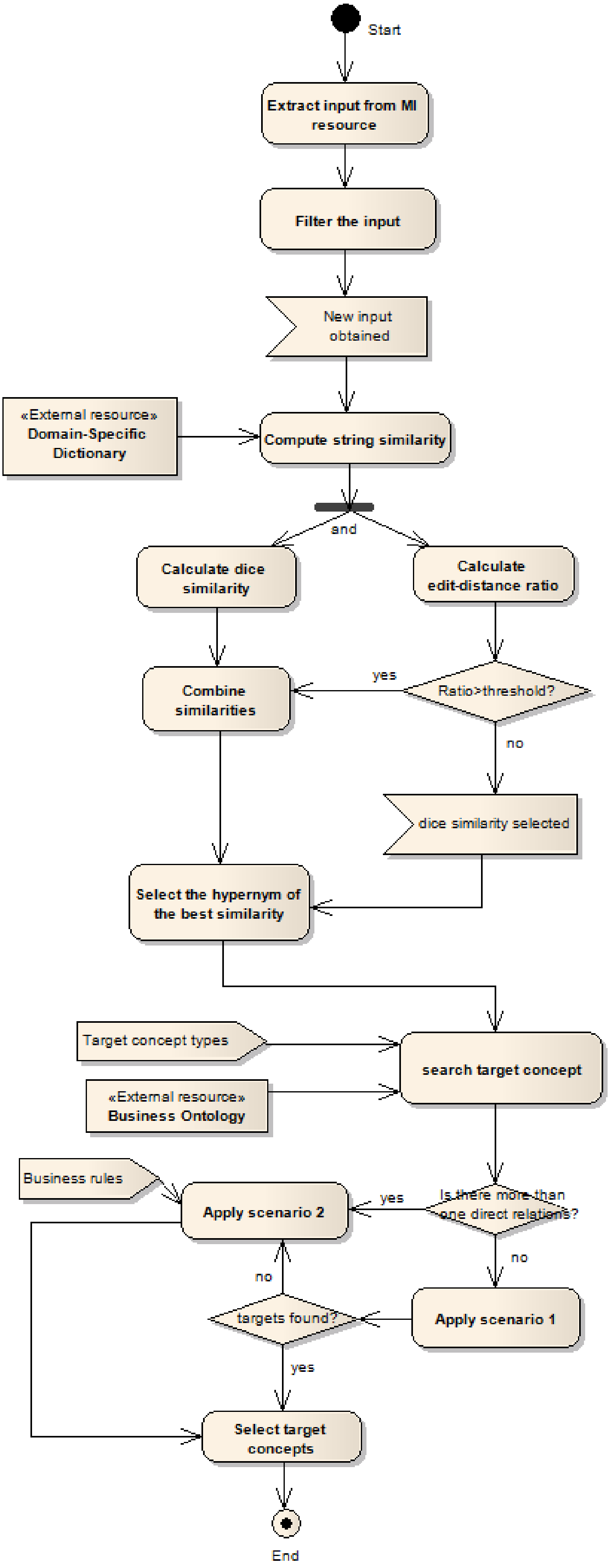
<u>Appendix B :</u> Example of Requirement Modeling with the ARIS Approach [Ferdian 2001]

Below an example taken from [Ferdian 2001] related to the four views of the ARIS Approach. Different techniques of requirement engineering are used to model the description of each view. In the example, an organizational structure is used for modeling the organization view. A function tree is used to model the function view. An entity-relationship diagram is used for modeling the data view. Finally, the EPC modeling approach [Mendling and Nüttgens 2005] *is used to express the process flow that involves the business concepts captured in the other views.*



Appendix C: Overview of the Mapping Process for One Set of Concepts

This appendix depicts the general process of the semantic mapping technique used to build semantic descriptors of resources. The mapping technique is presented in Chapter 5 (cf. section 5.3.2).


Start
Extract input from MI resource
Filter the input
New input obtained
«External resource» Domain-Specific Dictionary
Compute string similarity
and
Calculate dice similarity
Calculate edit-distance ratio
Combine similarities
yes
Ratio>threshold?
no
dice similarity selected
Select the hypernym of the best similarity
Target concept types
search target concept
«External resource» Business Ontology
Business rules
Apply scenario 2
yes
Is there more than one direct relations?
no
no
targets found?
Apply scenario 1
yes
Select target concepts
End

Appendix D: Detailed Description of Engineering-Document Retrieval Approaches

In this appendix, four approaches related to engineering-document retrieval are presented: the ***Waypoint framework [McMahon et al. 2004]*** *related to engineering-document annotation and retrieval with faceted-classification technique, the* ***EO search approach of [Li et al. 2007]*** *which is an engineering-ontology development method with a resource retrieval tool, the* ***EIR Framework [Yao et al. 2009]*** *for engineering-document indexing and search, and finally the* ***approach of [Li and Qiao 2012]*** *which proposed a semantic retrieval framework with a manufacturing-information ontology. These approaches were introduced in the state of the art (cf. section 2.2.2). Their detailed description is presented here.*

- The Waypoint framework [McMahon et al. 2004]:

The authors developed an integrated retrieval system for engineering documents. This one provides a uniform access to heterogeneous information collections and multiple document sources. In the proposed framework, the engineering documents can be annotated using pre-identified concepts and retrieved using a faceted-classification mechanism [Mas and Zacklad 2008]. Moreover, the access mechanism allows both keyword searching and browsing of classification schemes of the document collections. This system can be used in a stand-alone mode, as a document management system, or can be incorporated into other software systems with some adaptations. Fig. 1 shows the Waypoint architecture. It includes two main modules: a classifier module and a search module.

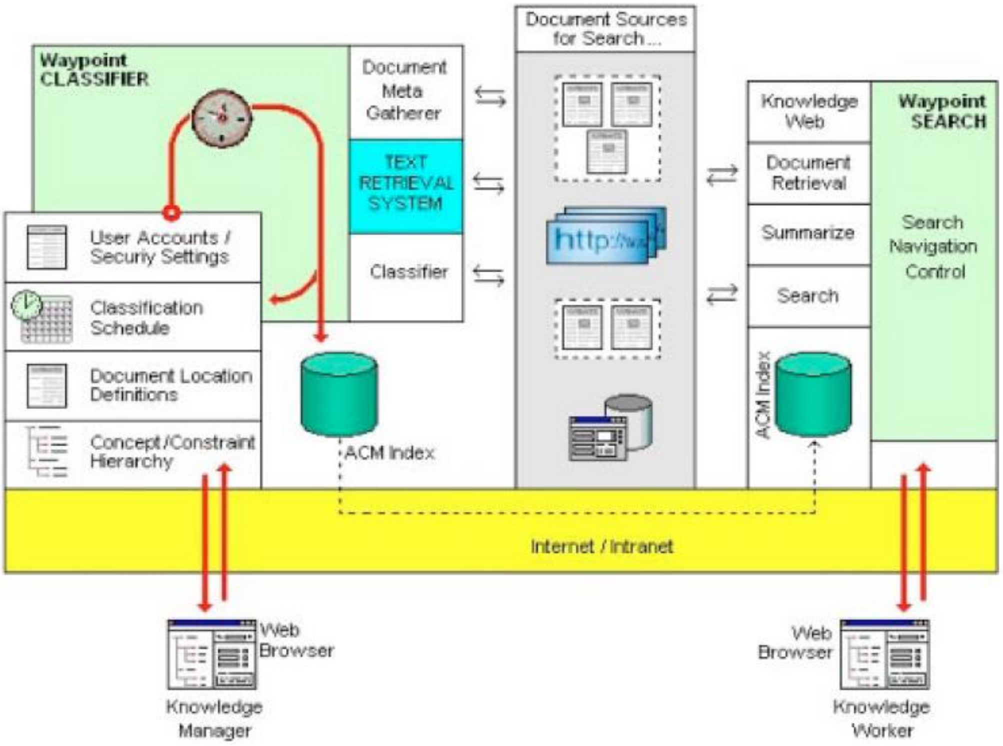


Fig. 1 : The Waypoint architecture [McMahon et al. 2004]

The classifier module regroups the "back-end" functions. They seek defining the classification schemes, identifying and locating the document collections to classify and configuring the automatic classifier. The classification scheme and document references

(URL, disk address, etc.) are maintained in a relational database. Engineering documents are considered here distributed through a number of systems —e.g. document management systems, company intranets, shared disk archives, virtual project rooms, etc. The Classifier uses a faceted-classification mechanism, where a facet regroups characteristics of a particular subject domain. The classification of information objects related to documents is carried out by associating each object to multiple properties from each facet. The eXchangeable Faceted Markup Language (XFML) is used, in a first step, for handling the classification of each document collection. These classification schemes and the relations between the documents and nodes in the schemes are stored and maintained in a database. The whole Waypoint system is managed by a scheduling system, which automates the classification process and builds index files used to support the browsing of the classification schemes.

The search module is dedicated to end-users. It provides, at first, a mean for document search and retrieval. The search function incorporates free-text search with the browsing technique. In addition, the user can save and reuse the results of the browsing operations. The system also provides a summarizer tool, which gives a short description of each document and incorporates it in the browsing interface. The summarization process uses document name and/or its meta-data (depending on the document type) and the text in the introductory paragraphs of the document.

The Waypoint framework was initially developed by the former Chrysler Corporation and has been taken up by Airbus Deutschland. The system has been applied to provide means for organizing and facilitating the access to electronic documents stored on shared network drives. An experimentation of the framework has been done in the Airbus UK site. The development of the faceted classifications was mostly based on the taxonomies of the directories that already existed within the company site, so to easily capture the concepts with which the engineers were familiar. The experimentation of the approach showed that engineers were able to more easily retrieve relevant documents using the Waypoint-hybrid-browsing approach, compared to the keyword search or conventional browsing of existing folder structures.

- The Engineering-Ontology (EO) search [Li et al. 2007]:

The authors proposed a semi-automatic ontology development methodology and integrated it in an engineering-information retrieval system, called EO-Search. The general approach has been developed and experimented in the context of the automotive sector.

The framework comprises six parts: pre-processing, ontology basis, ontology acquisition and maintenance, concept tagging, concept indexing and query processing. Fig. 2 depicts the overall architecture of the framework:

- ***The pre-processing*** task aims at converting engineering documents (catalog descriptions, CAD[29] drawings, technical reports, etc.) into a unified format which can be processed by the system (e.g. .txt format).
- ***The ontology basis*** contains an engineering ontology linked to an engineering lexicon. They assist the system in recognizing technical terms in documents and queries.

---

[29] Computer-Aided Design

- ***The ontology acquisition and maintenance*** part is integrated in the system to build and update the engineering ontology and lexicon.
- ***The concept tagging*** part aims at tagging the engineering documents with the concepts of the ontology and lexicon. The tagging process is transformed into an XML-based representation.
- ***The concept indexing*** process is used to index the XML documents with the tagged concepts using an inverted concept index [Li et al. 2008b]. This index is accessed when the system ranks the documents during the query processing.
- ***The query processing*** part deals with the users' queries using the engineering ontology and lexicon.

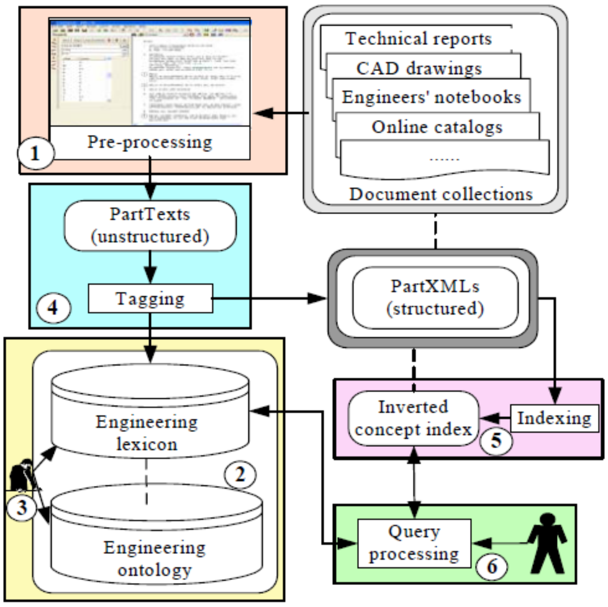


Fig. 2 : The system architecture of EO-Search

To implement the search system with the ontological approach, the authors proposed a methodology for the engineering-ontology development. This method is composed of six steps, a bit similar to Methontology [Fernandez et al. 1997], a top-down ontology-construction method. The method uses handcrafted acquisition process supported by computer-assisted tools. The six steps are summarized as follows:

Specification: the authors proposed a set of taxonomies of themes for the engineering ontology. Ten themes were identified as shown in Fig. 3. They also proposed an engineering lexicon which main role is to store a list of words and derivations of each concept identified in the taxonomy (e.g., *move* and *moving* are lexical terms of the functional concept *F-MOVE*).

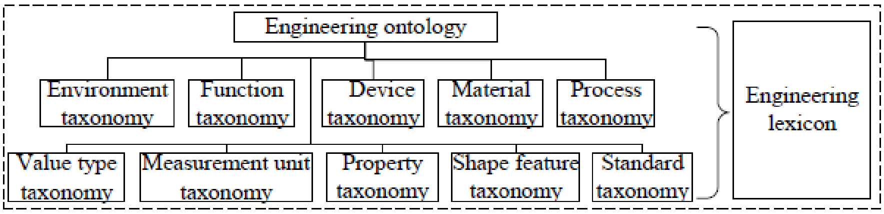


Fig. 3 : The schema of the ontology basis

Acquisition: conducts the knowledge-acquisition process by exploring and analyzing various knowledge engineering resources, where engineering concepts can be found. The concepts, relationships and lexical terms related to each taxonomy of a theme are acquired in this step.

Formalization: formatted worksheets are proposed as template to guide the structuring and formalization of the acquired knowledge [Li et al. 2008c]. These worksheets also enable automatic uploading of the acquired concepts into the Protégé editor for the population step.

Population: the authors developed a Protégé plug-in which can read the knowledge worksheets and generate the engineering ontology and lexicon. The manual method was more used for the maintenance of the ontology basis.

Evaluation: The OntoClean[30] methodology [Guarino and Welty 2009] were used to evaluate and validate the core content of the engineering ontology.

Maintenance: with the Protégé editor.

Once the ontology basis is built and integrated in the EO-Search system, the engineering documents can then be tagged using the concepts of the engineering ontology. Documents from various resources are first converted into .txt files during the pre-processing step. Afterwards, the ontology and lexicon are used to recognize concepts contained in the documents. Thus, each recognized word/phrase in a document is tagged by the corresponding concept of the ontology. XML documents (called PartXMLs in the approach) are generated as a result in this phase. In addition, these files are indexed in order to rank the relevancy of documents in query processing.

The search interface of the EO-Search system provides to end-users a keyword search and concept navigation. For each user query, the system processes it with a concept disambiguation technique and a concept abstraction metric. The disambiguation technique consists in calculating the correlations of the matched concepts between all the key words and the ontology concepts (including lexicon concepts). In fact, a concept is considered highly correlated with other concepts if they are semantically closer following their position in the taxonomy level in the ontology and if there are more words that match with the query key-words. The abstraction metric tries to exploit the structure and content of the engineering ontology in order to ferret out the true meaning, i.e. the target concept of the query. Finally, the proposed system returns for each user query a list of ranked documents categorized according to the concept categories as defined in the engineering ontology.

- Ontology-based EIR framework [Yao et al. 2009]:

The Engineering-Information-Retrieval (EIR) framework has been developed in the context of an aerospace innovation project in China. The authors proposed a unified platform for search from multi-source documents, to enable engineers to retrieve engineering information during the processes of product design, analysis and manufacturing. The framework comprises three main modules: ontology module, document analysis module and query processing module (Fig. 4).

The ontology module: this module is composed of *engineering ontology* and *application ontology*. The engineering ontology gives an abstract description (concepts and relations) of

---

[30] OntoClean is a methodology for analyzing ontologies based on meta-properties (identity, unity, rigidity and dependence)

an engineering domain, regardless of the document resources of the company. The application ontology gives the description of the document resources based on the concepts of the engineering ontology. Note that the resources here include CAD drawings, CAPP[31] sheets, design manuals, PDF documents and some images and demo videos. The engineering ontology structure has been built from an engineering-knowledge base. This one regroups a set of standard engineering-knowledge sources such as engineering lexicon, technology manuals, online resources, etc. The application ontology implements mappings between information related to the document resources and the concepts of the engineering ontology. To obtain the relevant information from the resources, the document-analysis module has been used.

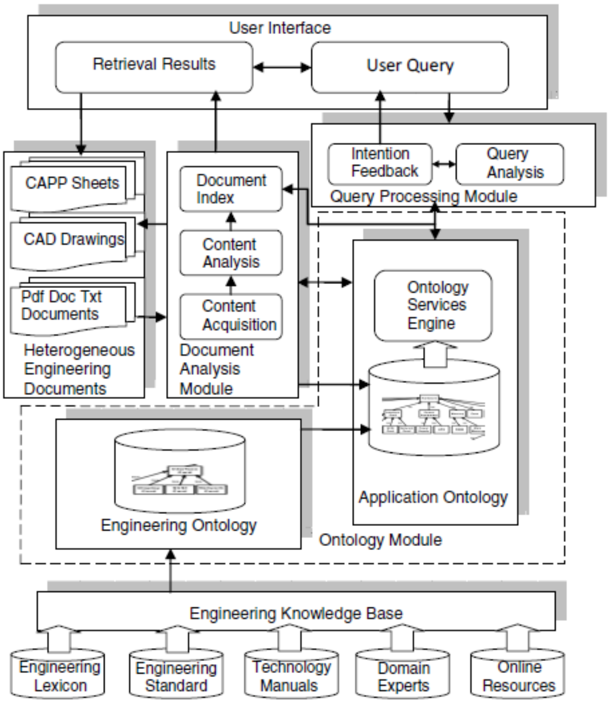


Fig. 4 : Ontology-based EIR framework

The document-analysis module: it allows exploring information in document resources and supplying the application ontology in terms of concepts instances and relations. The document-analysis process includes three steps:

- *Content acquisition*: because of the heterogeneity of the resources used in companies, the authors adopted many strategies to acquire information. For example, documents with textual content were converted into txt format. PDF extractor toolkits were used to extract information from pdf documents. Documents of manufacturing systems with specific formats were analyzed manually. For some specific resources like images and videos, the title and the meta-data of the format were captured.
- *Content analysis*: the authors did the analysis of documents' content to abstract the captured information during the acquisition process to put it in the application ontology. The free text generated during the content acquisition was converted into structured document information with meta-data, using Natural Language Processing (NLP) techniques and matching-concepts' strategy.

---

[31] Computer-Aided Process Planning

- *Document index:* document indexes are created in this step to associate the document resources with ontology concepts. The indexes are used to facilitate query processing during document search.

The query-processing module: this module includes a query-processing function and an intention-feedback function.

- *Query-processing:* the user queries are mainly composed of keywords, phrases and sometimes data value. The system maps the user query with the concepts of the application ontology and the engineering ontology using the semantic relations and semantic extension. The semantic extension includes transverse extension, which extends semantic relations in the same level coverage, and longitudinal extension which explores information and concepts with semantic relations in multiple levels.
- *Intention feedback:* this function enables interaction between the end-users and the system before retrieval in order to make the query close to user intention. The feedback is presented in the form of a tree graph where the recognized query concepts, relations and semantic depth are proposed to the user for further navigation and inputs modification, for a more precise querying.

- Ontology-based modeling of manufacturing information and its semantic retrieval [Li and Qiao 2012]:

This approach seeks to support the definition, the integration management and the retrieval of manufacturing datum in industries. A retrieval system was proposed based on manufacturing-information ontology and a semantic-similarity algorithm. The authors developed, at first, an UML conceptual model –called Manufacturing Information (MI) model– which describes and specifies information related to the manufacturing process. The main entities of this model arise from four manufacturing subjects: products, processes, resources and plants. The manufacturing-information ontology was developed from the MI model. Top-down and bottom-up modeling strategies were used to build the ontology. In the top-down strategy, the authors tried to divide the main entities of the MI model into sub-themes with the help of business experts. In the bottom-up strategy, they use several knowledge sources such as the international manufacturing-data standards, handbooks related to machining processes, the enterprises' production experiences and so on. The Protégé tool was used to implement the MI ontology before integration in the retrieval system.

The retrieval system relies on a semantic similarity algorithm which uses the concepts of the MI ontology to process user queries. The proposed algorithm supposes that two concepts have certain similarity according to the type of relations and distance between them. Three types of relations were identified here: kind-of, part-of and inference (e.g. Fig. 5). A similarity distance between each two concepts is calculated accordingly in order to determine the closeness among the keywords of a user query and the concepts of the ontology.

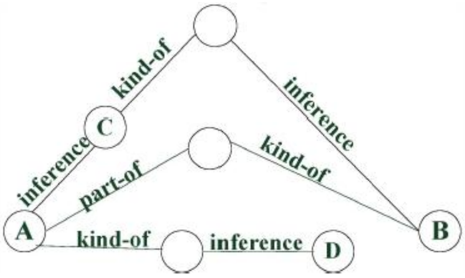


Fig. 5 : Example of relations between classes in the MI ontology

Fig. 6 illustrates the general idea of the framework and the process of semantic retrieval. The system takes as input one or some key words and returns in a web browser the top-ten relevant manufacturing-information resources ranked according to the biggest similarity results.

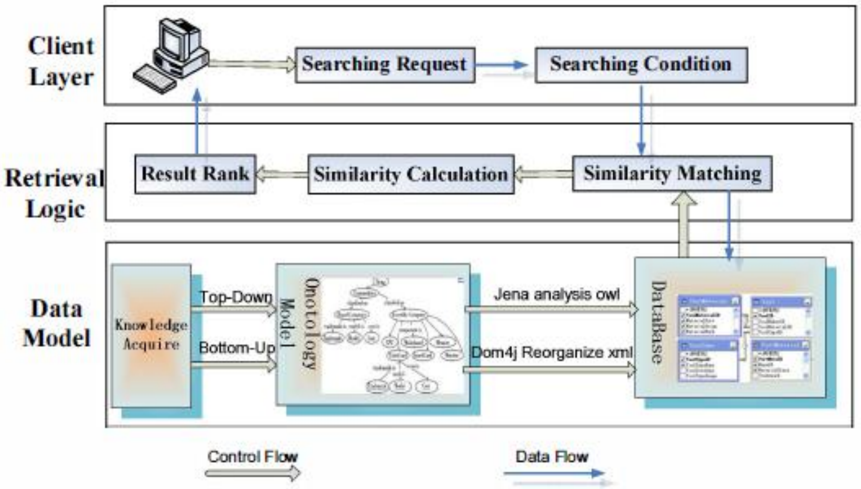


Fig. 6 : The semantic-retrieval framework

Appendix E: Detailed Description of Information-Resource Retrieval Approaches

*This appendix contains a detailed description of three approaches presented in the state of the art (cf. section 2.2.2). These approaches are: the **MEAT approach [Khelif et al. 2007]** related to the annotation and search of biological documents, the **Topic-Maps-based framework of [Kásler et al. 2006]** related to heterogeneous information-resource organization and retrieval, and the **HyperTopic approach [Zaher et al. 2006]** related to knowledge co-building and information-resource search.*

- The MEAT approach:

This approach [Khelif et al. 2007] was developed for biologists who work on DNA Microarray experiments, to support them in the validation and interpretation of their results. The authors proposed an approach and a system for the generation of ontology-based semantic annotations (called MeatAnnot), and a system for an advanced search by the biologists on the annotation base (called MeatSearch). Fig. 7 depicts the general idea of the MEAT approach.

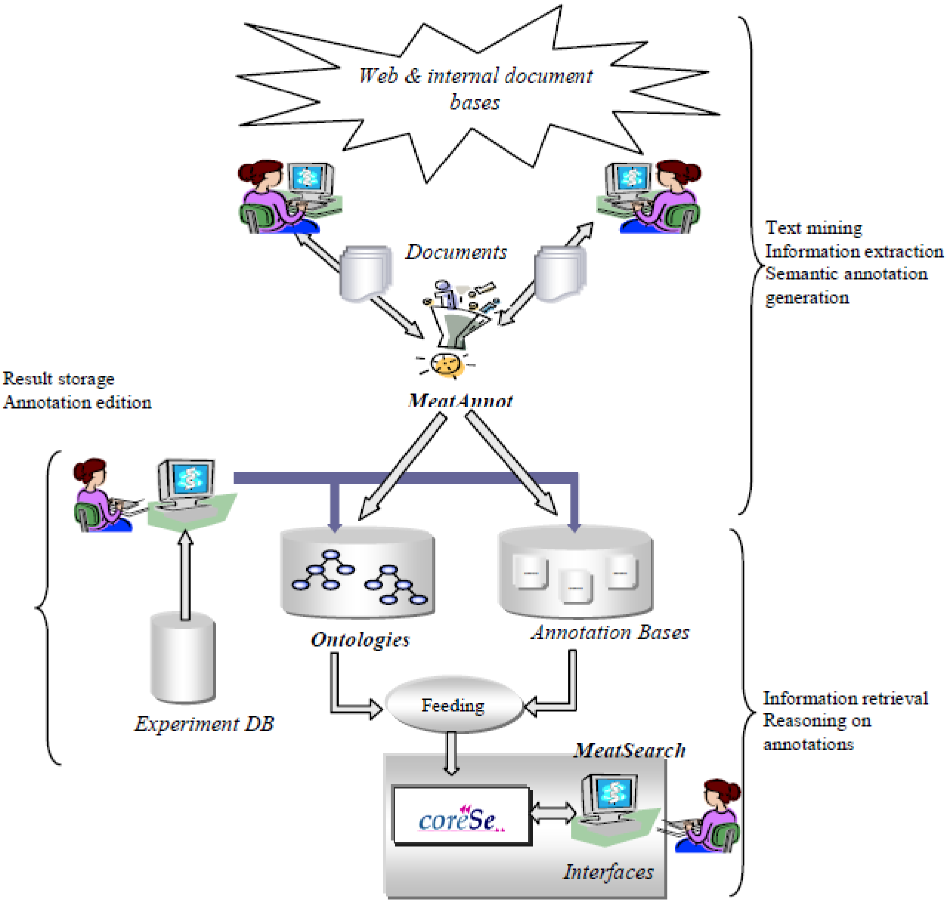


Fig. 7 : The MEAT approach

In the MeatAnnot system, the generation of ontology-based annotations on documents requires a lexicon of terms of the biological domain and an ontology describing this domain. The authors chose the semantic network of UMLS (Unified Medical Language System) [Hymphreys and Lindberg 1993] as upper-level ontology for the biomedical domain. Hence, starting from a textual document (like a scientific paper), the MeatAnnot system enables to generate a structured annotation based on the UMLS semantic network on one hand, and describes the semantic content of this document on other hand. The system processes texts and extracts interactions between genes and other UMLS concepts using the NLP technique with the Gate[32] tool. Thus, for each sentence identified in the scientific document, it tries to detect an instance of an UMLS relation and the instances of UMLS concepts linked by this relationship, so to generate an annotation describing this interaction. The generation process in summarized in three steps:

- *Relation detection*: for each UMLS relation (e.g. interacts_with, expressed_in, etc.), an extraction grammar process was manually done to extract all instances of this relation. The authors used the JAPE language, a language based on regular expressions, to write the information extraction grammar
- *Term extraction*: To extract terms, the authors used the Tokeniser module of GATE and the TreeTagger[33]. The Tokeniser splits text into tokens (e.g. numbers, punctuation, word, etc.) and the TreeTagger assigns a grammatical category (noun, verb, etc.) to each token. After the tokenizing and tagging processes, the MeatAnnot maps each candidate word with the concepts of the UMLS semantic network. If the term exists in UMLS, the answer is obtained in an XML format. This format is parsed to get information about the concept found (semantic type, synonyms, etc.) in order to generate the semantic annotation
- *Annotation generation*: the authors used the RASP module [Briscoe and Carroll 2002] which assigns a grammatical relation to sentence words. It allows discovering the concept instances that are linked by each relation, by mapping the sentence words with the UMLS concepts. Then it generates an annotation describing an instance of this relation.

After these processing steps, MeatAnnot generates an RDF annotation linked to each processed resource. This annotation base is used in the MeatSearch system. The authors developed a search system based on the semantic search engine CORESE (Conceptual Resource Search Engine)[Corby et al. 2004] to enable users to query on the annotation base. In addition, the search mechanism is enhanced with an RDFS ontology which references the concepts of the UMLS. The search inputs consists of a set of biological concepts that the user must choose as inputs, and the search system returns the related annotations to these inputs. It also returns the sentences from which the annotations were extracted and the document containing these sentences. This ability to trace the provenance of information seemed important and useful for the biologists to validate their experiments in a reliable way.

- The Topic-Maps-based approach for resource navigation [Kásler et al. 2006]:

The approach of [Kásler et al. 2006] was created to improve the access to the NetWorkshop conference proceedings. It relies on a software framework to semi-automatically generating a semantic presentation of information present in a set of text files. The resulting semantic network is used to organize the access to the involved resources in a web portal. The Topic-Maps technology has been used to handle the resulting network. This framework comprises four phases for the generation process: the data organization, the analysis, the Topic-Maps

---

[32] http://gate.ac.uk/conferences/training-modules.html

[33] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

population and the content-management phase. The whole process is incremental and requires user interaction in some steps to configure some parameters or to confirm some assumptions made by heuristic algorithms. The general approach of the framework is depicted in Fig. 8.

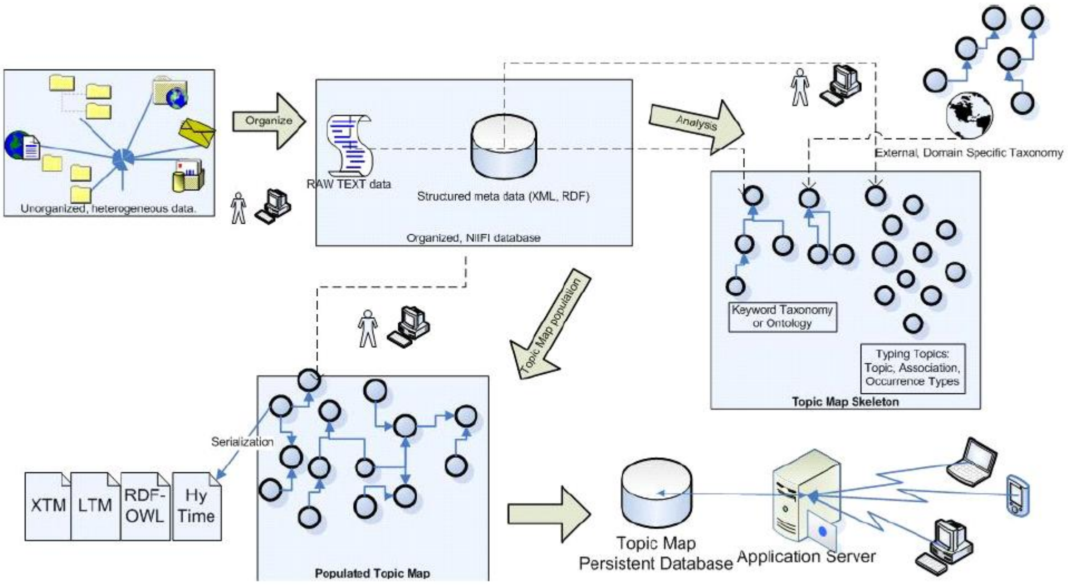

Unorganized, heterogeneous data.
Organize
RAW TEXT data
Structured meta data (XML, RDF)
Organized, NIIFI database
Analysis
External, Domain Specific Taxonomy
Keyword Taxonomy or Ontology
Typing Topics: Topic, Association, Occurrence Types
Topic Map Skeleton
Topic Map population
Serialization
XTM
LTM
RDF-OWL
Hy Time
Populated Topic Map
Topic Map Persistent Database
Application Server

Fig. 8 : The general approach of the framework

The four phases are described as below:

- ***Data-organization phase***: in this phase, all meta-data are extracted from various raw source texts and are stored in a structured way using the XML technology, so to have a uniform and formal structure. Some pattern-matching techniques (e.g. regular expressions) was used to facilitate this task. Some parsers were also used to get information from popular formats (e.g., MS Docs, pps, pdf, etc.)
- ***Analysis phase***: the goal of this phase is to obtain a Topic-Map skeleton containing typed topics and topic keywords. In fact, the typed topics and the keywords will serve respectively to create topics and instances of the final Topic Map. Two steps are accomplished for this purpose. The first consists in identifying the topics and associations from the structure of the source text. This process is manually configured by the user and is supported by the TMHarvest framework[34]. The user has to create an XML configuration file which contains some patterns like X-Path and regular expressions to extract topics from the structured meta-data. The second technique consists in using existing external ontologies and taxonomies such as the FOLDOC source[35] to assign keywords to papers. The associations between keywords are defined by these external ontologies. The assigned keywords will serve at the next step to categorize the resources.
- ***Topic-Maps population phase***: the goal of this phase is to identify the concrete instances of the topic types identified in the precedent phase. This operation is treated by configuring an XML file which contains the patterns for the mapping between the source text and the topic instances to extract. Afterwards, resources are associated to instances by an automated process of documents' classification based

---

[34] http://www.folge2.de/topicmaps/tmharvest/userdoc01/en/index.html#f

[35] FOLDOC is a free dictionary of computing: http://foldoc.org/

on the keywords identified in the previous step and supported by the WEKA framework[36]. The main techniques used are the unsupervised classification and statistical information retrieval. In fact, the authors used in a first step a full-text-search-based classification using the keywords of the FOLDOC ontology. This step gives rough estimation of the used keywords. For this reason a second step is necessary to accurate the first classification. The authors propose then to use the Vector Space Model (VSM)[37] to reduce the irrelevant classifications and keywords by calculating relative relevancy of each occurring keyword in a paper. They defined a threshold of 66% after many experiments. Thus, an association between a paper and a keyword is created if and only if the relative relevancy is in the first 66% among the other keywords present in the paper. At the end of this phase, a Topic Map of information resources is obtained and is stored, afterwards, in a persistent database (XTM file or SQL database)

- ***Access via a web portal / Content management***: the generated Topic Map is presented to the end-users with a content management system. It is based on a user-friendly web portal where the topics and resources can be viewed and edited. The user can therefore explore the referenced resources using the semantic network obtained with the approach by navigating from one topic to another.

- The HyperTopic approach [Zaher et al. 2006; Zaher et al. 2007; Zaher et al. 2008]:

An HyperTopic [Zacklad et al. 2002] is a knowledge-representation model and language, created by Tech-CICO lab[38] for cooperative building of knowledge in the context of the Socio-Semantic Web [Guittard et al. 2005]. The HyperTopic model is considered as a specialization of the Topic-Maps model, created to guide users for structuring knowledge and retrieving information in a same activity domain. This model is based on five main concepts (Fig. 9) to describe a knowledge domain:

- **Point of view**: represent a perspective vision of a set of topics and entities. It widely corresponds to a vision of an actor or a group of actors in a business domain
- **Topic**: is a symbolic representation of a subject as defined in the Topic-Maps standard
- **Entity**: is a generic object related to a subject (topic).
- **Attribute**: is the descriptor of the entity[39].
- **Resource**: corresponds to an information resource like a document, a web page, etc.

The HyperTopic model was created to build Knowledge-Based-Market Place (KBM) [Cahier 2005] by many community members. The HyperTopic concept enables to structure and share information resources between users who have common interests. This approach is supported by the Agorae tool [Zaher et al. 2006] for the co-construction of the KBM model. Three roles are predefined in this collaborative approach: the *semantic editor* for the creation and the modification of a point of view or a topic, *the contributor* for the creation of an entity for a given topic under a given point of view and finally *the reader* for the browsing of the HyperTopic Map. The construction of the KBM has been done at a first stage by the

---

[36] http://www.cs.waikato.ac.nz/ml/weka/
[37] http://en.wikipedia.org/wiki/Vector_space_model
[38] The Tech-CICO Lab is located in France, in the University of Troyes: http://techcico.utt.fr/fr/index.html
[39] An entity can be view as a table in a relational database and the attribute like a field of the table

authors of the approach for the running of the agorae tool. They did series of interviews with domain experts to capture all topics. Afterwards, they did a deep analysis of the studied domain with the help of domain experts in order to structure topics on hierarchical levels and to extract the main relevant categories that will constitute the points of view. In fact, a point of view corresponds to a vision of an actor or a group of actors for the access to information in a given activity. Thus, the HyperTopic approach had led the authors of the project to seek consensus among end-users in order to constitute a coherent and relevant knowledge map [Guittard et al. 2005].

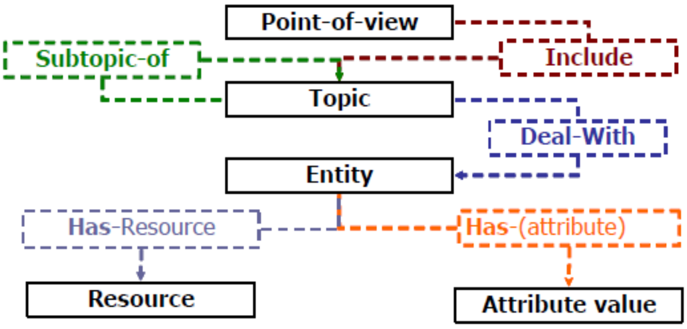


Fig. 9 : The HyperTopic basis

This approach has been experimented in France Telecom and EADS companies. The main purpose was to share knowledge about software projects. The Agorae tool was used to create different knowledge related to these projects and link them to resources (like documentations, project homepages, software links, etc.). As the concept of point of view is the starting description point in the HyperTopic model, the authors tried to identify them at first. Five points of view were identified for the software projects:

- *themes/features* (e.g., software development, system tools, multimedia, games),
- *software engineering* (e.g., methods and tools for development, integration, deployment, etc.),
- *business models* (e.g., hardware models and services, Linux distribution models, non-business model, etc.),
- *legal aspects* (e.g., legal point of view, licenses, patents, etc.)
- *actors/stakeholders* (organizational points of view dealing with software communities, companies, institutions establishment, or research project).

Afterwards, the rest of the concepts of the HyperTopic model must derive from these points of view, in order to keep a consistent description of the knowledge domain. Finally, the resulting HyperTopic knowledge map is what the authors call a “semiotic ontology”. The end-users can consult the concepts of this map through the reader module of the Agorae tool. They have a general view upon all viewpoints and can navigate among several hundred topics. For each topic selected at any level, the users can see corresponding entities and the topics transversely related to this topic.

Appendix F: Detailed Description of Component Retrieval and Service Retrieval Approaches

This appendix gives some details with illustrations about component retrieval and service retrieval approaches, which were introduced in the state of the art (section 2.2.3). The presented approaches are:

- ***the approach of [Quan et al. 2007]****: an ontology scheme for software-component description and a semantic-retrieval process*
- ***the approach of [Peng et al. 2009]****: ontology-driven paradigm for component representation and retrieval*
- ***the CompRE tool [Alnusair and Zhao 2010]*** *: an Eclipse plug-in for semantic search of components*
- ***the Larks framework [Sycara et al. 2002]:*** *a service retrieval approach*
- ***the GODO approach [Gomez et al. 2006]****: a goal-oriented approach for service discovery and search*
- ***the SATIS approach [Mirbel and Crescenzo 2010]*** *: a goal-oriented approach for web-service retrieval in the context of neuroscientist community*

- The approach of [Quan et al. 2007]:

[Quan et al. 2007] developed an ontology scheme for software-component description with the OWL language. This approach was developed to support the component-based development method, a method that relies on assembling and composing already built software components. The authors proposed an ontology scheme to describe component information, so that it can provide semantic reasoning during the component-retrieval process. They identified four facets of description:

- *Component form*: provides elementary description about the form of the component. As description, the authors identified *Basic Information* (like component name, registry date, vendor name, price, etc.), *the presentation form* (e.g., source code, graph, EXE, DLL, etc.), *the kind* of the component which is mainly the used technology (Net, Java Bean, EJB, Corba, etc.) and the hierarchy (i.e. the corresponding stage of software development process)
- *Application environment*: provides information about the implementation environment of the component including software implementation (operating system, network, data base) and hardware implementation.
- *Application functionality*: serves to describe the function of the component and its application domain (a particular business domain).
- *Semantic information*: provides the semantic retrieval for the users that need to reuse a component (e.g., inputs, outputs, pre-conditions, post-conditions, etc.)

In the retrieval process, the user query is translated into OWL representation in order to facilitate the semantic matching with the ontology concepts. The architecture of the component-retrieval system comprises three layers (Fig. 10): a user-interface layer which displays the user interface for component search; a middle layer which allows parsing user queries and matching them with the domain ontology using semantic reasoning; and a resource layer which stores the components and the owl files related to user queries. In the proposed system, the user can specify its query in natural language. The system converts it into an owl format and tries to match this query with the domain ontology using semantic associations.

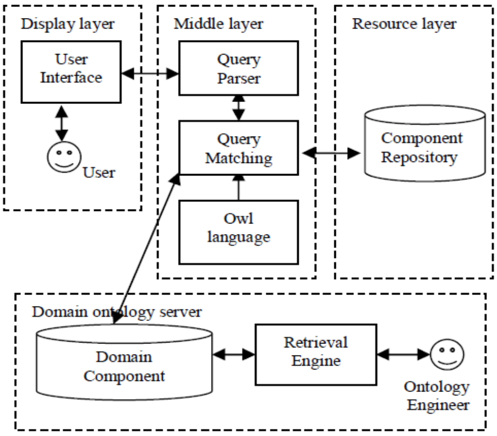


Fig. 10 : The architecture of the component-retrieval system [Quan et al. 2007]

- <u>The ontology-driven paradigm of [Peng et al. 2009]:</u>

In [Peng et al. 2009], the authors proposed an ontology-driven paradigm for component representation and retrieval. They developed a component ontology composed of five facets (Fig. 11): provider, environment, application domain, functions, interfaces. The provider class records component providers' name and point of contact. The environment refers to the implementation information, i.e. hardware or software. Each environment type can import concepts from external ontologies (hardware ontology, software ontology). The application domain describes the application scope of the component (i.e. the context of use). The function class refers to the main function of the component. It comprises three properties: operation, subject and object. The operation is the action performed by a subject and the object is the target of the operation. Finally, the interface class describes the interface properties of a component, in particular, the input/output parameters. The application domain, the functions and the interfaces import information related to them from the domain ontology of software components.

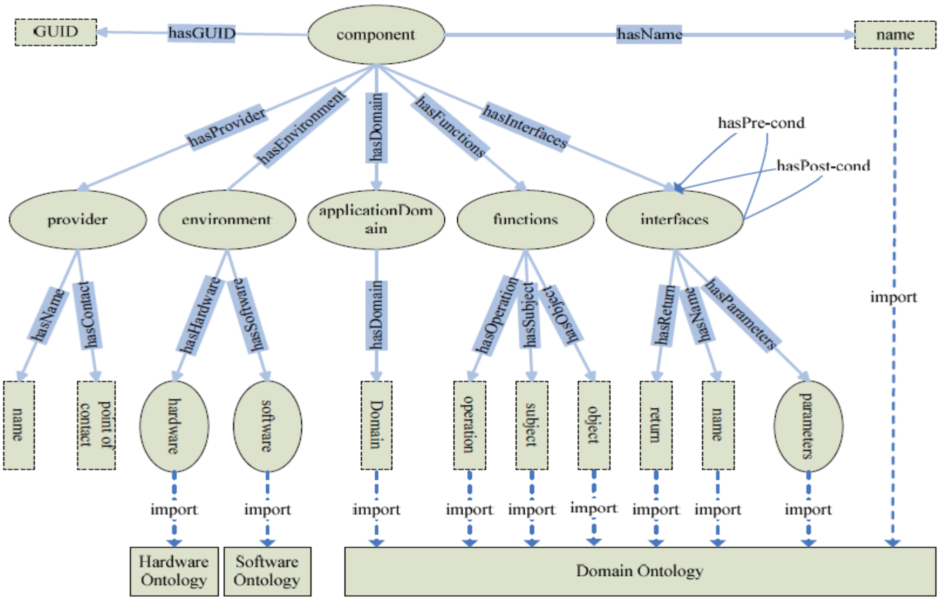


Fig. 11 : The component ontology [Peng et al. 2009]

To use the proposed component ontology in the retrieval system, the authors merged the component ontology and the domain ontology into a synthetic ontology for the implementation. They defined a retrieval algorithm based on syntactic matching and on the semantic association between the concepts of the resulting ontology and the user needs.

- The CompRE tool [Alnusair and Zhao 2010]:

In [Alnusair and Zhao 2010], the authors developed a component-search tool (called CompRE) as a plug-in for the Eclipse Environment. This tool is based on a knowledge base which gathers a source-code ontology, a component ontology and a domain-specific ontology:

- *The source-code ontology (called SCRO)* captures the structure of an object-oriented library and helps understanding the relations and dependencies among source-code artifacts. The generation of the semantic instances for the concepts and relations specified in SCRO were done automatically by parsing the source code
- *The component ontology (called COMPRE)* is an extension of the SCRO ontology. It gives additional component-specific descriptions. It also makes the link with the concepts of the domain ontology. The annotation of the component with the COMPRE ontology were done manually by the authors
- *The domain ontology (called SWONTO)* conceptualizes the software libraries, by providing a common vocabulary with the unambiguous and conceptually sound terms that can be used to annotate software components. To that aim, the authors developed a mini ontology for data retrieval in the semantic-web domain using the OWL language.

The generated semantic instances of the proposed knowledge base were serialized using the RDF syntax, so as to allow SPARQL queries using the Jena framework[40]. The ontology-based-search mechanism in the CompRE tool is performed by semantically matching user queries with the component descriptions in the populated SCRO and SWONTO ontologies. The user queries are expressed with the concepts of the domain ontology, either on a simple data-entry form (translated afterwards into SPARQL queries), or directly using a SPARQL query. After query performing, the system ranks the resulting instances and returns the results in a viewer which enables further exploration of each recommended component.

- The LARKS framework [Sycara et al. 2002]:

The LARKS framework [Sycara et al. 2002] has been developed to provide a dynamic matchmaking among software agents regarding service search and retrieval. The authors proposed a specification schema to define a service. This specification consists of a Context of specification, Type of variables, Inputs and Outputs, InConstraints and OutConstraints, ConcDescription related to the ontological description of used words and finally TextDescription related to a textual description of specification. The search request is based then on the elements of this specification schema. The search framework (Fig. 12) relies on three knowledge sources: an advertisement database for referencing the services, a domain ontology (called conceptDB) describing and referencing all the vocabulary used in the LARKS specification and an auxiliary database storing word distances and hierarchy types.

---

[40] Jena is an Eclipse plug-in for managing RDF ontologies : http://jena.apache.org/

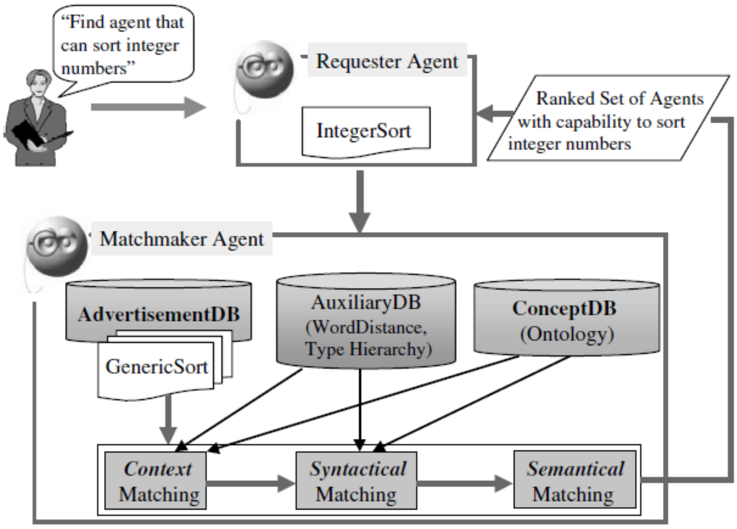

"Find agent that can sort integer numbers"
Requester Agent
IntegerSort
Ranked Set of Agents with capability to sort integer numbers
Matchmaker Agent
AdvertisementDB
GenericSort
AuxiliaryDB (WordDistance, Type Hierarchy)
ConceptDB (Ontology)
Context Matching
Syntactical Matching
Semantical Matching

Fig. 12 : The matchmaking process with the LARKS framework [Sycara et al. 2002]

The search mechanism is based on five types of filters: context, profile, similarity, signature and constraint [Denayer 2004]. The composition of these filters allows establishing different degrees of correspondences: exact match of the five filters, plug in match (matching of the signature and constraint filters) and relaxed match (encompasses exact match and plug in match). In this way, the LARKS framework tries to provide a combination of syntactic and semantic matching, according to a context of matching, to satisfy as best as possible a user request.

- The GODO approach [Gomez et al. 2006]:

In [Gomez et al. 2006], the authors developed the GODO approach using a domain ontology and a goal-oriented approach. GODO is a goal-oriented-service discovery platform, developed to ensure the achievement of user intentions by means of semantic web services.

The interface (GUI) of the GODO system captures user needs with two different input methods: the user can write his request in natural language text or can be assisted in formulating his intention. An ontology-guided input is applied in the last case (Fig. 13). It offers the users some recommendations to complete the statements expressing their intentions/goals. These recommendations are extracted from an external ontology repository. The natural-language input is treated with a Language Analyzer. This one receives a sentence as input and processes it by determining the concepts (attributes and values) and relations included using ontologies and a knowledge-acquisition technique named MCRDR [Vazey and Richards 2006]. This technique tries to obtain the relationships between concepts using a knowledge base which contains linguistic expressions representing generic conceptual relationships, and by a subsystem which infers the participants in these relationships. The semantic network stemming from the user request and the language analyzer is regarded as a lightweight-ontology. The latter is matched with goal templates of the goal-template repository, where different types of goals are stored.

As depicted in Fig. 13, the Goal Loader retrieves the goal templates from the repository and transmits them to the Control Matcher through the Control Manager. The Goal Matcher compares the lightweight-ontology to the description of the goal templates. From this matching, several goals are selected, composed by the Control Manager in order to build up the sequence of execution of the web services.

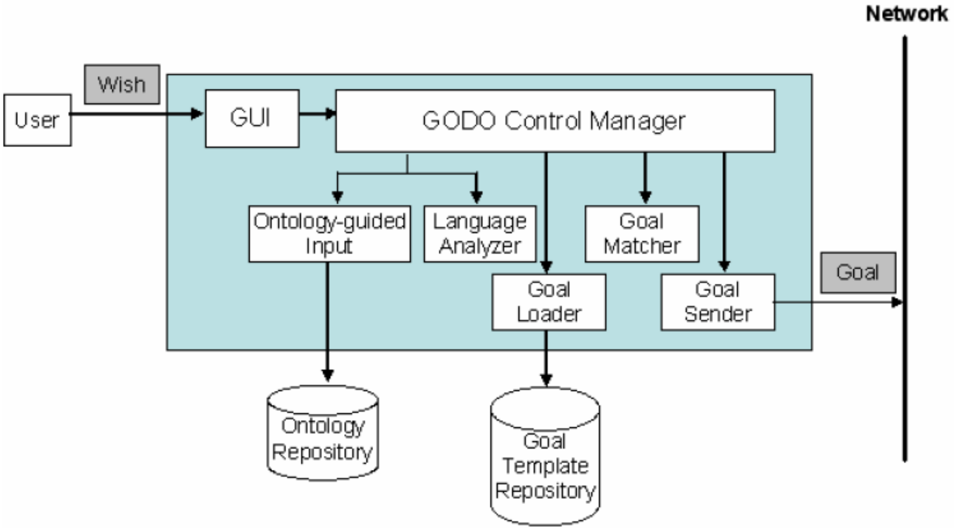


Fig. 13 : The GODO architecture [Gomez et al. 2006]

- The SATIS approach [Mirbel and Crescenzo 2010]:

[Mirbel and Crescenzo 2010; Mirbel and Crescenzo 2009] proposed the SATIS approach, a goal-oriented approach for web-service retrieval in a context of neuroscientist community. It mainly supports a neuroscientist seeking for web services to operationalize an image processing pipeline. The authors used three ontologies in this approach:

- a map ontology to capture user intention. The authors used the Map model of [Rolland 2003] to model user intention before implementing it in an ontology. A map is a labeled directed graph with intentions as nodes and strategies as edges between intentions. The RDF format is used in this approach for the map ontology
- an OWL-S ontology to describe web-services' functionalities
- a domain ontology related to the application domain of the approach (neuroscientist in this case).

The map ontology, the domain ontology and the OWL-S ontology are used to formalize the high-level end-user's intentional requirements and to specify associated generic web service descriptions. The RDF annotations and SPARQL queries (represent patterns descriptions of web services) are assembled into rules, which are regarded as fragments of a search approach for web-service retrieval. These fragments can be reused for further search. In fact, the search mechanism in the SATIS approach consists in operationalizing the image processing pipeline whose needs were elicited during the capture of the user intention. The rendering step is supported with a backward-chaining engine using the CORESE semantic engine[41]. Thus, during the search procedure, the domain expert only selects the intention characterizing his/her image-processing pipeline and the system will search for the rules to use. The high-level intentional needs are created dynamically as needed during the backward-chaining process (such as temporary sub-goals), and this process continue until the descriptions of web services corresponding to the sub-goals are found. Therefore, the captured user intention is considered satisfied. A set of web services' descriptions are given to the domain expert as a result.

---

[41] http ://www-sop.inria.fr/edelweiss/software/corese/

Appendix G: Some Alignment-oriented Approaches

Some of the alignment-oriented approaches cited in the state of the art (section 2.3) are detailed in this appendix with illustrations. The approaches presented below are:

- *the **Goal-oriented framework** of **[Bleistein et al. 2005]** for Business/IT alignment*
- *the **INSTAL method [Thevenet 2009]** related to the strategic alignment*
- *the **framework of [Wieringa et al. 2003]** related to the alignment between application architectures and business contexts*
- *the **Zachman framework [Zachman 2003]** of Enterprise Architecture*
- *the **ARIS Architecture [Ferdian 2001]***

- The goal-oriented framework for Business/IT alignment [Bleistein et al. 2005]:

The authors proposed a conceptual framework for modeling the business/IT alignment by describing the strategic goals of the company and its activities and processes. This framework relies on three steps:

- modeling of the company strategy using a goal-refinement mechanism until the software goals of its IT system
- defining the business context and the company system using the problem-frames technique [Jackson 1999]
- modeling the business process with activity-role diagrams, so that to get, at the end, the description of the system processes.

The "Problem Frames" are a mechanism and notation for structuring the analysis of software requirements and the design of a software solution. It is based on a graph problem which describes the properties of a software problem by representing the existing context and how the stakeholders would like the system to be. In such a diagram, the requirements and contexts are bound by reference links or constraint. The requirements of the strategic level are refined by the concept of progression between problems (Fig. 14), because they are too abstract to design a system solution [Bleistein et al. 2004b].

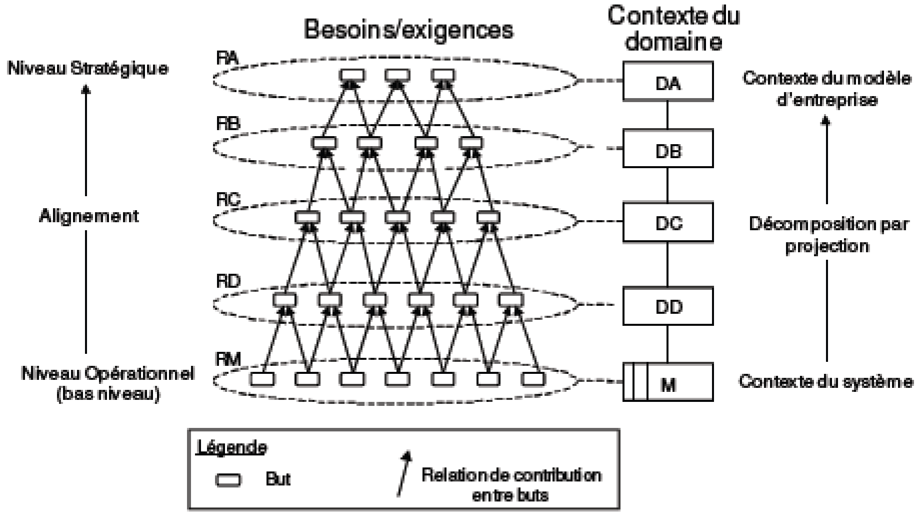

Besoins/exigences
Contexte du domaine
Niveau Stratégique
Alignement
Niveau Opérationnel (bas niveau)
RA
RB
RC
RD
RM
DA
DB
DC
DD
M
Contexte du modèle d'entreprise
Décomposition par projection
Contexte du système
Légende
But
Relation de contribution entre buts

Fig. 14 : The goal model using the problem-frames technique [Bleistein et al. 2005]

Goals can be formulated at different levels of abstraction, from a high strategic level to a low operational level. The entities to align are represented by intentional models and problem frames, and eventually, by the company's own models relating to the business process. The authors used the BMM (Business Motivation Model)[BRG 2010] model of the BRG[42] (Business Rules Group) to organize business strategies. This model is a conceptual framework, relying on two main concepts: "the means" (Mission, strategy, tactic) and "purpose" (vision, goal, objective). The authors proposed to unify this model with the goal model [Bleistein et al. 2006]. The links are made with the elements of the "purpose" concept of the BMM model.

- The INSTAL Method [Thevenet 2009]:

The INSTALL (INtentional STrategic ALignment) method was created to model the alignment between the strategic level and the operational level in companies. The core content of this method relies on a “pivotal model”, modeled with the map formalism [Rolland 2003]. This model can be designed either by aggregation of the strategic and operational elements (i.e. grouping), or by subsumption of these elements (i.e. the conjunction of two levels). At the end, the content of this model consists in the contribution of both.

In order to have a pivot model in which the elements of the two levels to align are subsumed, a generic concept must be found to solve the problem of conceptual discordance. The concept found in this method is the intention (Fig. 15). Thus the intentional-pivot model allows defining links on different levels of granularity and addressing the problems of conceptual discrepancy and offset of levels of abstraction. The alignment links are structurally and semantically rich. A role can be associated to each element involved in a link.

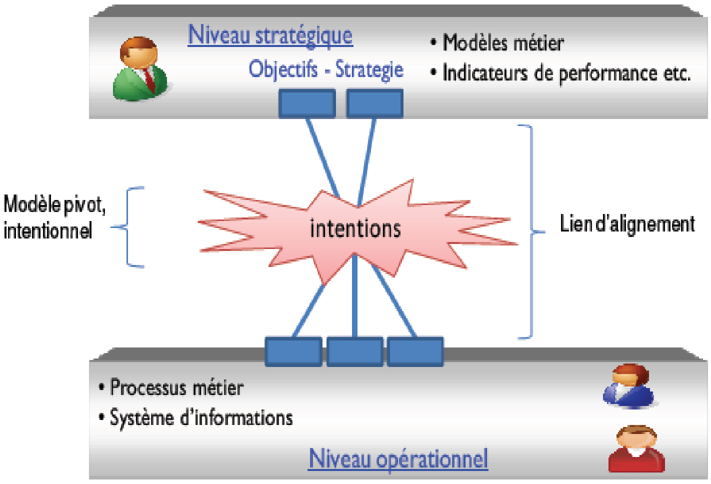


Fig. 15 : The Intentional-pivot model of the INSTAL method [Thevenet 2009]

The strategic concepts (objectives, business plans) and operational concepts (applications, business processes) are defined at different levels of abstraction, and this difference can make difficult the matching between the elements of each level. The pivot model was designed to facilitate the mapping process, by specifying the intentions underlying the alignment. Note that this approach considers that there must be a common intention shared between the elements to align, otherwise there is no alignment.

---

[42] http://www.businessrulesgroup.org/bmm.shtml

- The alignment framework of [Wieringa et al. 2003]:

[Wieringa et al. 2003] proposed a framework to analyze and design the alignment between application architectures and the business context. The authors consider that the alignment problem stems from the connection of three worlds: the social world, the linguistic world and the physical world. The authors proposed then a framework to construct an information system where these three different worlds and their components are aligned. This framework gives descriptions of various elements related to each world, traditionally used to model business processes or to develop information systems. As depicted in Fig. 16, this description starts from the physical world (hardware network) and is abstracted through the layer of each world until the business environment. *The physical world* deals with the network services. *The linguistic world* deals with the platform and application services (e.g. implementation environment, application system, etc.) and *the social world* deals with the business services (e.g. processes, business context, etc.). As an example, a program or a piece of software belongs to the linguistic world whereas people belong to the social world. To align these two worlds, we must ensure that the meaning seen by people to the symbols at the software interface corresponds to the manipulations of these symbols by the software and that these manipulations respond to people expectations. To align software to the physical world, we must allocate it to processing devices with a location in the physical network of the company.

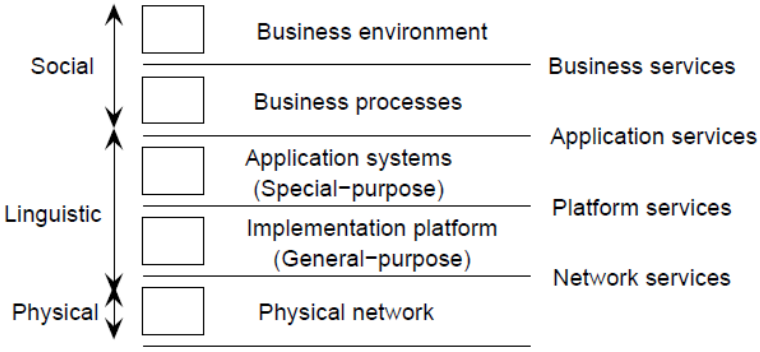
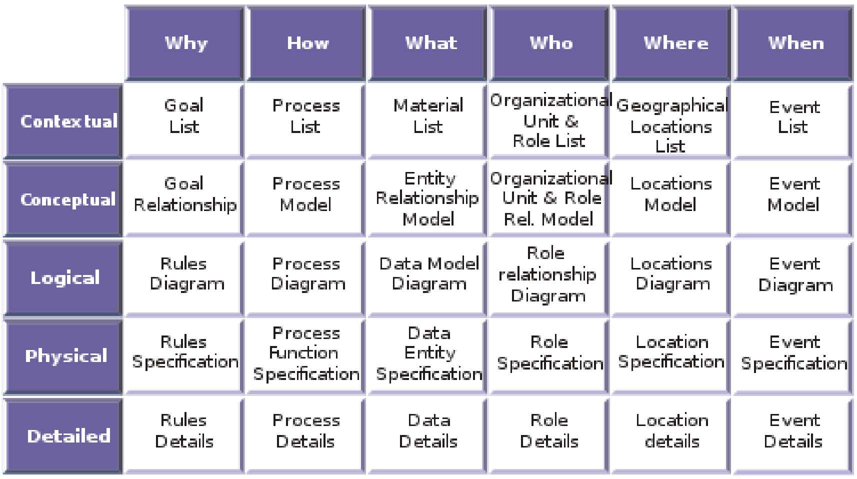


Fig. 16 : The layers of the three worlds to align [Wieringa et al. 2003]

- The Zachman Framework [Zachman 2003]:

The framework for Enterprise Architecture of [Zachman 2003] is a two dimensional classification scheme for descriptive representations of an Enterprise. The author proposed a holistic vision of the company, of its organization and Information System. This vision organizes the enterprise description under six different perspectives. Each perspective underlies a question (Fig. 17):

- Why: the motivation description
- How: the function description
- What: the data description
- Who: the people description
- Where: the network description
- When: the time description



	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organizational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organizational Unit & Role Rel. Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location details	Event Details

Fig. 17 : The perspectives of the Zachman framework

Each perspective is used to describe an entity of the enterprise. The answers to these questions constitute a comprehensive depiction of the subject or object being described in the enterprise. Moreover, the description of each subject is done according to five facets (contextual, conceptual, logical, physical and detailed description). A number of rules governs the relationships between the various descriptions provided by the perspectives. By using this framework, an architect has a complete EA documentation that can be useful for IT systems integration and maintaining.

- The ARIS architecture [Ferdian 2001]:

ARIS is a framework for modeling business processes, usually used for industrial environments [Ferdian 2001]. This framework is composed of four levels of process engineering and five descriptive views to describe this process. The four levels of ARIS provide a process-oriented business management, from organizational engineering to IT implementation. They include *Process Engineering*, *Process Planning and Control*, *Workflow Control* and *Application Systems*. To support the description of these activities, ARIS relies on dividing the business processes into separate views and integrating these views to form a complete view of the whole business process.

The five views of the ARIS architecture are (Fig. 18):

- ***Data view***: related to the business objects/data that belong to the business activity of the company
- ***Function view***: related to the business activities, functions, goals of the company
- ***Organization view***: related to the organizational structure of the company (business areas, services, business actors, etc.)
- **Resource view** related to the IT components and software implementation
- ***Control view***: the processes that make the link between all ARIS views

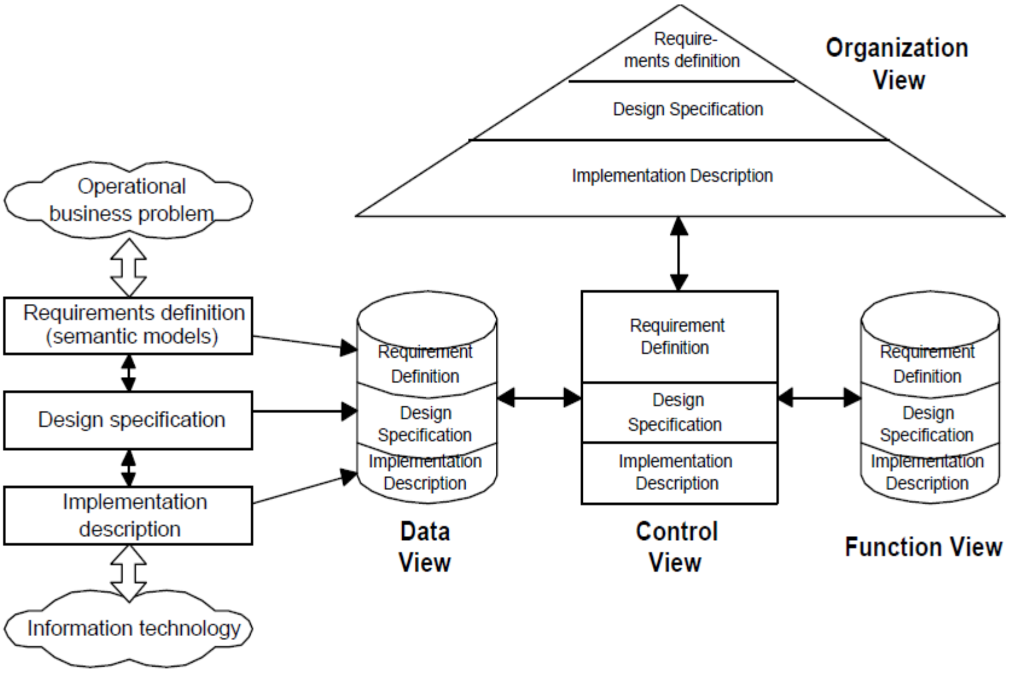


Fig. 18 : The ARIS views

The separation between the views lies in the relationships between components, meaning that they are relatively strong in the same view and relatively weak between different views. A further partitioning inside these views is realized via the descriptive levels. In fact, these levels are ordered according to their proximity to Information Technology, because each view covers a complete description from business-requirement definition to design specification and implementation description.

Appendix H: Algorithm of the Levenshtein Distance

The algorithm of Levenshtein[43] [Chapman 2008] computes the similarity between the strings *a* and *b* using a matrix *L* of dimension $(m + 1) \times (n + 1)$ where *m* and *n* are respectively the lengths of *a* and *b*. The matrix is fulfilled using the minimum result between the deletion (L[i-1, j] + 1), the insertion (L[i, j-1] +1) and the substitution (L[i-1, j-1] + cost) operations. Moreover, a *cost* variable is used to flag a substitution operation. *cost* = 0 when two characters match and 1 otherwise. The algorithm returns the value of $L[m, n]$ related to the minimum number of operations to do for transforming *a* into *b*.

```
Input: String[m] a, String[n] b
Output: number of edits L[m,n]

begin
//Initialization of the Levenshtein matrix
for i←0 to m do
    L[i,0] ← i;
end for
for j←0 to n do
    L[0,j] ← j;
end for
//filling of the matrix
for i←1 to m do
    for j←1 to n do
        if ((a[i-1] == b[j-1])) then
            cost ← 0;
        else
            cost ← 1;
        end if
        L[i,j] ← min ((L[i-1, j] + 1), (L[i, j-1] + 1), (L[i-1, j-1] + cost));
    end for
end for
return L[m,n];
end
```

---

[43] Other sources : http://www.levenshtein.net/
http://en.wikipedia.org/wiki/Levenshtein_distance