



**HAL**  
open science

# Un cadre ontologique générique de modélisation, de capitalisation et de partage de Connaissances Métiers Situées en Ingénierie Système

Olfa Chourabi

► **To cite this version:**

Olfa Chourabi. Un cadre ontologique générique de modélisation, de capitalisation et de partage de Connaissances Métiers Situées en Ingénierie Système. Informatique [cs]. Conservatoire National Des Arts et Métiers, Paris; École Nationale des Sciences de l'Informatique, Tunisie, 2009. Français. NNT : . tel-00446695

**HAL Id: tel-00446695**

**<https://theses.hal.science/tel-00446695>**

Submitted on 13 Jan 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**THESE EN COTUTELLE**

**Présentée pour l'obtention du Diplôme de Doctorat en Informatique du  
CNAM (Paris) et de l'ENSI (Université de La Manouba)**

*Par*

**OLFA CHOURABI**

*Intitulée*

**Un cadre ontologique générique de modélisation, de  
capitalisation et de partage de Connaissances Métiers Situées  
en Ingénierie Système**

Préparée au sein des laboratoires RIADI et CEDRIC : Equipe SEMpIA



Soutenue le 22/12/2009 au CNAM de Paris devant le jury d'examen :

**Président :**

Hervé PINGAUD, Professeur, Ecole des Mines d'Albi-Carmaux, France.

**Rapporteurs :**

Hervé PANETTO, Professeur, Université Henri Poincaré - Nancy I, France.  
Riadh ROBBANA, Professeur, Ecole Polytechnique de Tunisie, Tunisie.

**Examineurs :**

Françoise CARON. Docteur en Chimie Physique, Gérante d'EIRIS Conseil, France.

**Directeurs de Thèse :**

Mohamed BEN AHMED, Professeur émérite, Ecole Nationale des Sciences de l'Informatique  
de l'Université de la Manouba, Tunisie.  
Yann POLLET, Professeur Titulaire de la Chaire d'Intégration des Systèmes, Conservatoire  
National des Arts et Métiers de Paris, France

## Remerciements

Je tiens tout d'abord à remercier chaleureusement mes deux directeurs de thèses Le Professeur émérite Mohamed BEN AHMED et le Professeur Yann POLLET pour m'avoir acceptée dans leurs laboratoires respectifs à l'ENSI et au CNAM. Leur encadrement, leurs conseils, leur aide si précieuse, leur présence, leur patience et leur soutien constant ont été pour moi au cours de ces dernières années source de motivation, d'encouragement, de réconfort et de persévérance. Pour avoir été pour moi des guides avisés et des modèles de rigueur dans le domaine de la recherche, je tiens à leur exprimer ma profonde gratitude.

Je suis très reconnaissante à M. Hervé PANETTO Professeur à l'Université Henri Poincaré - Nancy I et à M. Riadh ROBBANA, Professeur à l'Ecole Polytechnique de Tunisie pour l'honneur qu'ils me font d'accepter d'être les rapporteurs de cette thèse. Je les remercie pour l'intérêt et la disponibilité qu'ils ont manifestés à l'égard de ce travail ainsi que pour leurs relectures, leurs observations judicieuses et leurs remarques constructives qui m'ont aidé à améliorer la qualité de ce manuscrit et qui m'ont fait profiter de la richesse de leurs expériences.

Je tiens à remercier vivement Mme Françoise CARON Docteur en Chimie Physique et M. Hervé PINGAUD Professeur à l'Ecole des Mines d'Albi-Carmaux d'avoir bien voulu faire partie du jury de cette thèse et d'apporter leur caution scientifique à ce travail.

Je tiens à remercier vivement le Ministère des affaires étrangères et européennes, Ambassade de France en Tunisie, de leur prise en charge pour effectuer des séjours en France dans le cadre préparation de cette thèse.

Mes remerciements s'adressent à tous mes collègues du Laboratoire RIADI, ceux la chaire d'intégration des systèmes du CNAM et de l'équipe SEMpIA, qui m'ont fait profiter de leurs conseils et pour m'avoir soutenue surtout dans les moments difficiles.

Mes derniers et non moins profonds remerciements iront à ma famille qui m'a toujours soutenue et encouragée tant dans mes études que dans mes recherches.

---

## Résumé

---

Les travaux de cette thèse concernent la gestion de connaissances au sein des processus techniques d'Ingénierie Système (IS). Ces processus peuvent être considérés comme un espace de création continue de connaissances faisant appel à des savoir-faire multi-métiers et produisant de nombreux « artefacts », allant des spécifications aux composants logiciels et matériels jusqu'au système lui-même, et présentant des degrés très divers de formalisation. A ces éléments, il convient d'ajouter les alternatives de solutions et les justifications de décisions, pour constituer, au final, un ensemble de connaissances métiers implicites aujourd'hui très peu valorisées dans les organisations, et que ces dernières cherchent actuellement à réutiliser au mieux au cours de projets ultérieurs.

Pour fournir des éléments de réponse à cette problématique nous proposons un cadre ontologique générique dédié à la modélisation formelle et consensuelle des savoir-faire métiers de l'IS. Nous analysons ensuite l'application de ce cadre à la description des expériences de projets d'ingénierie que nous dénommons Connaissances Métiers Situées (CMS). Pour gérer ces expériences nous proposons d'une part, un modèle de capitalisation de CMS sous forme d'annotations sémantiques des dimensions situation, but, choix et décisions d'ingénierie. Et d'autre part, un modèle de partage de CMS suggérant les expériences potentiellement utiles au cours de projets ultérieurs. Les propositions conceptuelles de cette thèse ont été opérationnalisées à l'aide du formalisme des Graphes Conceptuels et appliquées dans le domaine de l'Ingénierie de Systèmes de défense.

**Mots clés :** Gestion de Connaissances, Ingénierie Ontologique, Ingénierie Système, Cadre ontologique générique de modélisation de connaissances en IS, Capitalisation et partage de connaissances métiers situées en IS.

---

## Abstract

---

This thesis deals with Knowledge Management issue in Technical System Engineering (SE) Processes. These processes could be considered as a continuous knowledge creation space. In fact, SE projects generate multiple artifacts presenting different formalization degree, such as requirements specification, system architecture, and hardware/ software components. Transitions between project phases stem from decision making processes supported both by generally available domain knowledge and engineering experiences. Most often, this knowledge is only known implicitly, relying heavily on system engineer's personal background. To fully exploit this intellectual capital, it must be explicitated and shared among project teams. To address this issue we propose a generic ontological framework for formal and consensual modelling of SE know-how. We study the potential application of this framework for describing SE project experiences, entitled Situated Explicit Engineering Knowledge (SEEK). To manage these experiences we introduce two conceptual models:

- A knowledge capitalization model for SEEKs: based on semantic annotation of engineering problem resolution records and design rationale.
- A knowledge sharing model suggesting potentially relevant SEEK(s) in subsequent engineering projects.

The conceptual proposal (ontological framework) is operationalized with an extended Conceptual Graphs formalism and applied to defense system engineering domain.

**Key words:** Knowledge Management, Ontology Engineering, System Engineering, generic ontological framework for system engineering knowledge modeling, Situated Explicit Engineering Knowledge capitalization and sharing.

# Table des matières

<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>I. CONTEXTE DE RECHERCHES .....</b>	<b>1</b>
<b>II. OBJECTIFS DE RECHERCHES .....</b>	<b>2</b>
<b>III. CONTRIBUTIONS .....</b>	<b>4</b>
<b>IV. ORGANISATION DU RAPPORT .....</b>	<b>4</b>

---

## **PARTIE I : DE LA GESTION DE CONNAISSANCES EN INGENIERIE SYSTEME**

---

<b>CHAPITRE I : INGENIERIE SYSTEME ET BESOINS EN GESTION DE CONNAISSANCES .....</b>	<b>7</b>
<b>I.1 INTRODUCTION .....</b>	<b>8</b>
<b>I.2 DE L'INGENIERIE SYSTEME (IS) .....</b>	<b>8</b>
<i>I.2.1 Définitions préalables .....</i>	<i>8</i>
<i>I.2.2 Ingénierie Système et processus .....</i>	<i>11</i>
1.2.2.1 Le concept de processus en IS .....	11
1.2.2.2 Modèles du cycle de développement en IS : .....	12
1.2.2.3 Panorama des normes d'Ingénierie Système .....	14
<i>I.2.3 L'IS : un processus inhérent à l'innovation et centré sur les connaissances.....</i>	<i>17</i>
<b>I.3 DE LA GESTION DE CONNAISSANCES .....</b>	<b>20</b>
<b>I.3.1 DEFINITIONS ET TYPOLOGIES DE CONNAISSANCES .....</b>	<b>21</b>
<b>I.3.2 APPROCHES DE GESTION DE CONNAISSANCES.....</b>	<b>25</b>
<b>I.4 GESTION DE CONNAISSANCES ET INGENIERIE SYSTEME .....</b>	<b>27</b>
<i>I.4.1 Analyse des environnements supportant les processus d'IS.....</i>	<i>27</i>
<i>I.4.2 Fonctionnalités attendues d'un Système de Gestion de connaissances en IS.....</i>	<i>29</i>
<b>I.5. CONCLUSION .....</b>	<b>30</b>
<b>CHAPITRE II .....</b>	<b>31</b>
<b>SYSTEMES DE GESTION DE CONNAISSANCES EN INGENIERIE SYSTEME .....</b>	<b>31</b>
<b>II.1 INTRODUCTION.....</b>	<b>32</b>
<b>II.2.CATEGORISATION DES SYSTEMES DE GESTION DE CONNAISSANCES ET INGENIERIE SYSTEME .....</b>	<b>32</b>
<i>II.2.1 Typologie des Mémoires Organisationnelles.....</i>	<i>36</i>
<i>II.2.2 Typologie retenue pour analyser les travaux d'IS .....</i>	<i>39</i>
<b>II.3 SYSTEMES DE GESTION DE CONNAISSANCES METIERS.....</b>	<b>41</b>
<i>II.3.1 Système à Base de Connaissances(SBC).....</i>	<i>41</i>
<i>II.3.2 Systèmes de « Knowledge Based Engineering » (KBE).....</i>	<i>43</i>
<i>II.3.3 les Lignes de Produits (LdP).....</i>	<i>45</i>
<i>II.3.4 Synthèse .....</i>	<i>47</i>
<b>II.4 SYSTEMES DE GESTION D'EXPERIENCES .....</b>	<b>48</b>
<i>II.4.1 Les Systèmes à Base de Cas.....</i>	<i>49</i>
<i>II.4.2 Les Référentiels de conception (« Design repositories »).....</i>	<i>52</i>
<i>II.4.3 Les Mémoires de projets .....</i>	<i>53</i>
<i>II.4.4 les Mémoire orientée processus .....</i>	<i>55</i>
<b>II.5 SYNTHÈSE .....</b>	<b>57</b>

<b>CHAPITRE III:DE L'INGENIERIE ONTOLOGIQUE EN IS.....</b>	<b>60</b>
III.1 INTRODUCTION .....	61
III.2 CONCEPTS DE BASE DE L'INGENIERIE ONTOLOGIQUE (IO) .....	61
<i>III.2.1 Définitions des ontologies</i> .....	61
<i>III.2.2 Typologies des Ontologies</i> .....	63
III.2.2 .1Typologie selon l'objet de conceptualisation .....	63
III.2.2.2 Typologie selon le niveau de complétude .....	64
III.2.2.3 Typologie selon le niveau de granularité.....	65
III .2.3. 4 Typologie selon le niveau d'expressivité.....	65
<i>III.2.3 Processus de développement, méthodes et outils de l'IO</i> .....	67
III.2.3.1 Processus de développement des Ontologies .....	67
III.2.3.2 Méthodes d'IO .....	68
III.2.3.3 Environnements et outils d'IO .....	69
<i>III.2.4 Applications de l'IO</i> .....	70
III.3 DE L'INGENIERIE ONTOLOGIQUE EN INGENIERIE SYSTEME .....	72
<i>III.3.1 L'Ingénierie Ontologique comme base de systématisation de connaissances métiers</i> .....	72
III.3.1.1Ontologies de domaine pour la systématisation des connaissances d'ingénierie .....	73
III.3.1.2 Ontologies de tâches pour la systématisation des connaissances d'ingénierie.....	77
<i>III.3.2 L'Ingénierie Ontologique comme base de partage de connaissances</i> .....	80
III.4 CONCLUSION .....	82

---

**PARTIE II : PROPOSITION D'UN CADRE ONTOLOGIQUE DE MODELISATION, DE CAPITALISATION ET DE PARTAGE DE CONNAISSANCES EN IS**

---

<b>CHAPITRE IV :PROPOSITION D'UN CADRE ONTOLOGIQUE DE MODELISATION DE CONNAISSANCES METIERS EN IS : ONTOIS.....</b>	<b>86</b>
IV.1 INTRODUCTION .....	87
IV.2 METHODOLOGIE D'INGENIERIE DU CADRE ONTOLOGIQUE ONTOIS.....	87
<i>IV.2.1 Rôle des ontologies fondationnelles</i> .....	88
<i>IV.2.2. Ingénierie ontologique et patrons de conception</i> .....	89
IV.3. ARCHITECTURE GENERALE DU CADRE ONTOLOGIQUE ONTOIS .....	91
IV.4. CONCEPTUALISATION DU CADRE ONTOLOGIQUE ONTOIS.....	93
<i>IV.4. 1. Strate fondationnelle : extension de l'ontologie DOLCE</i> .....	93
<i>IV.4. 2. Strate noyau du cadre OntoIS</i> .....	95
<i>IV.4. 2. 1 Facette Système à faire : le Système étudié</i> .....	96
IV.4. 2. 1.1 Patron conceptuel ontologique (PCO) contextuel .....	96
IV.4. 2. 1.2 Patron conceptuel ontologique (PCO) fonctionnel.....	97
V.4. 2. 1.3 Patron conceptuel ontologique (PCO) organique.....	98
<i>IV.4. 2. 2 Facette Système pour faire : le projet</i> .....	99
IV.4.2.2.1 Patron conceptuel ontologique (PCO) des processus d'IS : .....	99
IV.4.2.2.2 Patron conceptuel ontologique (PCO) des ressources .....	100
IV.4.2.2.3 Patron conceptuel ontologique des artéfacts (PCO) .....	100
IV.5. EXEMPLE DE SPECIALISATION DE PATRONS CONCEPTUELS ONTOLOGIQUES (PCO) NOYAUX .....	101
IV.6. CONCLUSION : CLASSES D'APPLICATIONS POTENTIELLES DU CADRE ONTOIS .....	103

## CHAPITRE V:

<b>CAPITALISATION ET PARTAGE DE CONNAISSANCES METIERS SITUEES (CMS) EN IS .....</b>	<b>106</b>
V.1. INTRODUCTION .....	107
V.2. PROPOSITION D'UN MODELE DE CONNAISSANCES METIERS SITUEES (CMS) EN IS .....	107
V.2.1. <i>Position du problème</i> .....	107
V.2.2. <i>Exemple illustratif : cas d'ingénierie d'un système de transport automatique</i> .....	108
V.2.3. <i>Modèle conceptuel de CMS</i> .....	109
V.3. FORMALISATION DE LA PROPOSITION .....	111
V.3.1 MODELE D'ONTOLOGIE.....	112
V.3.1.1 Le niveau terminologique.....	112
V.3.1.2 Le niveau Axiomatique .....	115
V.3.2.MODELE DE CAPITALISATION DE CMS .....	117
V.3.3. MODELE DE PARTAGE DE CMS :.....	121
V.4 CONCLUSION .....	125

---

## PARTIE III : OPERATIONALISATION ET VALIDATION DE LA PROPOSITION

---

## CHAPITRE VI:

<b>OPERATIONNALISATION DE LA PROPOSITION DE GESTION DES CONNAISSANCES METIERS SITUEES (CMS) 127</b>	
VI.1 INTRODUCTION .....	128
VI.2. LE FORMALISME DES GRAPHS CONCEPTUELS (GC) .....	130
VI.3 OPERATIONNALISATION DU SYSTEME DE GESTION DE CMS A L'AIDE DES GC.....	136
VI.3.1 <i>Opérationnalisation du modèle d'ontologie</i> .....	136
VI.3.2 <i>Opérationnalisation du modèle de capitalisation de CMS</i> .....	141
VI.3.3 <i>Opérationnalisation du modèle de partage de CMS</i> .....	145
VI.4 ARCHITECTURE D'UN SYSTEME DE GESTION DE CMS.....	148
VI.5 CONCLUSION .....	150

## CHAPITRE VII : ETUDE DE CAS ET EXPERIMENTATION

<b>CAS D'UN SYSTEME DE DEFENSE : INGENIERIE D'UN MISSILE .....</b>	<b>151</b>
VII.1 INTRODUCTION .....	152
VII.2 INGENIERIE ONTOLOGIQUE D'UN MISSILE .....	153
VII.2.1 <i>Ontologie de la facette contextuelle</i> .....	154
VII.2.2 <i>Ontologie de la facette fonctionnelle</i> .....	158
VII.2.3 <i>Ontologie de la facette organique</i> .....	162
VII.3 SCENARIOS DE CAPITALISATION ET DE PARTAGE DE CMS.....	165
VII.3.1 <i>Scénarios de capitalisation de CMS</i> .....	165
VII.3.2 <i>Scénario de partage de CMS</i> .....	171
VII.4 CONCLUSION : .....	173

<b>CONCLUSION GENERALE .....</b>	<b>174</b>
----------------------------------	------------

<b>LISTE DES PUBLICATIONS.....</b>	<b>176</b>
------------------------------------	------------

<b>REFERENCES .....</b>	<b>178</b>
-------------------------	------------

# Introduction générale

## I. Contexte de recherches

L'Ingénierie Système (IS) est une démarche méthodologique générale mettant en œuvre un processus coopératif et interdisciplinaire de résolution de problèmes, visant à transformer les besoins de parties prenantes, en une solution réalisant un compromis équilibré entre besoins et possibilités technologiques dans le contexte d'engagement de fonctionnalité, coût, délais, qualité.[Meinadier, 2002]. Les systèmes actuels sont marqués par une complexité multiforme aux niveaux organisationnel, fonctionnel, et contextuel.

Pour maîtriser cette complexité, de nouvelles structures organisationnelles ont été proposées en IS : aux organisations séquentielles, cloisonnées par une hiérarchie professionnelle, ont succédé les structures par projets, l'ingénierie concourante ou simultanée, le travail collaboratif et le co-développement avec des partenaires extérieurs, modifiant ainsi les frontières des processus de l'IS. Pour répondre à un besoin d'amélioration continue de ces processus, plusieurs standards d'IS ont été suggérés par l'EIA<sup>1</sup> (EIA 632) et par l'ISO (ISO 15288<sup>2</sup>) ainsi que plusieurs modèles de maturité (EIA/IS 731<sup>3</sup>, CMMI<sup>4</sup>). Depuis ces standards et ces modèles se sont imposés au monde de l'ingénierie préconisant de recueillir, outre évidemment les artefacts d'ingénierie avec leurs différentes versions, une large typologie de connaissances utilisées ou générées au cours du déroulement des processus.

Si l'intérêt de la capitalisation et du partage des connaissances d'ingénierie est largement reconnu, sa mise en œuvre reste trop souvent limitée. Force est de constater que dans la réalité des projets industriels, ce partage de connaissances est généralement fondé sur l'échange informel, voire verbal entre experts et novices. Ces connaissances sont relatives aux concepts des métiers d'ingénierie ou à certains éléments de projets passés, à l'instar des problèmes traités par ces projets, les décisions prises, et les contraintes considérées.

Cette problématique est la préoccupation principale de la discipline de la Gestion de Connaissances (en anglais, Knowledge Management (KM)). Le KM est un domaine de recherche proposant des méthodes, des techniques et des outils assistant la création, l'acquisition, le partage et l'utilisation des connaissances dans une organisation, en vue d'améliorer l'apprentissage individuel, collectif ou organisationnel [Grundstein, 2002]. Le KM s'inscrit dans une approche pluridisciplinaire, pouvant être abordé de plusieurs points de vue humains, socio-organisationnels, techniques et économiques [Wiig,1999]. Ce domaine de recherche a fait l'objet de plusieurs travaux en mobilisant soit les concepts et les démarches

---

<sup>1</sup> EIA : Electronic Industries Alliance. [http://sepo.spawar.navy.mil/632\\_Tutorial.pdf](http://sepo.spawar.navy.mil/632_Tutorial.pdf)

<sup>2</sup> Systems Engineering – System Life-Cycle Processes. <http://www.afis.fr/doc/normes/normes4.htm>

<sup>3</sup> System Engineering Capability Model.(<http://www.geia.org>)

<sup>4</sup> Capability Maturity Model Integration.(<http://www.sei.cmu.edu/cmmi/>)

du domaine de l'Ingénierie de Connaissances (IC) [Charlet et al, 2000] soit plus récemment les concepts de l'Ingénierie Ontologique (IO) [Mizoguchi, 2000].

L'IC est l'étude des concepts, méthodes et techniques permettant de modéliser et/ou d'acquérir les connaissances dans des domaines se formalisant a priori, peu ou pas. L'identification, la modélisation, l'explicitation, l'opérationnalisation et l'utilisation des connaissances sont parmi les activités centrales en IC. La principale application des techniques d'IC est la définition de Systèmes à Base de Connaissances permettant le stockage, la consultation et le raisonnement sur les connaissances stockées dans une organisation. [Furst, 2004]

Quant à l'Ingénierie Ontologique (IO), cette discipline est née de la volonté de fonder un soubassement formel de connaissances indépendante de leurs domaines d'applications pouvant servir à la génération de modèles de représentation de connaissances et de modèles de raisonnement sur ces connaissances pour des Systèmes à Base de Connaissances (SBC), assurant ainsi la portabilité des applications, leurs interopérabilités et leurs réutilisations. [Mizoguchi, 2003] [Bachimont, 2000]

L'analyse des recherches actuelles dans le domaine d'IS [Brandt et al, 2007] [Skyman, 2001] [Sharmin et al, 2009], a révélé que les concepts de capitalisation et de partage des connaissances sont peu explorés et donc peu formalisés, en ce sens qu'ils ont été proposés pour des cas *ad-hoc* ou domaine dépendante.

Dans ce contexte, les travaux de cette thèse proposent un cadre ontologique générique de modélisation, capitalisation et partage de connaissances en IS basé sur une combinaison de concepts fondamentaux du KM et de l'IO. Cette combinaison vise à générer une sémantique globale et consensuelle couvrant plusieurs facettes des processus d'IS et explicitant les connaissances métiers tout au long du cycle de vie de ces processus, en vue de rendre cette sémantique partageable et réutilisable par l'ensemble des parties prenantes dans des situations similaires et quel que soit le domaine d'application d'ingénierie.

## II. Objectifs de recherches

L'objet d'étude retenu dans le cadre de cette thèse est la modélisation, la capitalisation et le partage de connaissances dans les processus techniques d'Ingénierie Système (IS) : couvrant l'ensemble des activités allant de l'expression des besoins à la proposition d'une solution.

En IS, la transformation d'un besoin émergent en une proposition de système fait appel à de multiples activités intellectuelles faisant passer progressivement de concepts abstraits à la définition rigoureuse de produits/artéfacts.

Au cours de tout projet d'IS, plusieurs catégories de connaissances sont générées. C'est ainsi qu'au cours des phases conceptuelles les connaissances générées concernent les spécifications, les fonctions, le comportement, les contraintes et les alternatives de solutions du système à produire. Durant les phases ultérieures ces connaissances concernent les composants physiques (matériels et logiciels). Ces connaissances sont représentées dans des formats structurés, non structurés et semi structurés selon des modèles de produits (modèles relationnels, objets, modèles de documents).

De plus ces connaissances sont situées dans un contexte de projets (phases de processus d'IS, contraintes de projets, ressources (humaines, logicielles, matérielles, méthodologiques). Cette large typologie de connaissance constitue un savoir-faire valorisable dans des projets ultérieurs.

Afin de couvrir toutes les dimensions cognitives d'un processus d'IS, nous avons dégagé deux principales classes de connaissances à gérer:

- Les connaissances métiers : les « pratiques standards » matérialisées sous la forme de normes et de modèles de processus guidant les activités d'ingénierie ainsi que les « pratiques institutionnalisées » matérialisées par les bonnes pratiques du métier, ou liées à une organisation,
- Les expériences de projets : les « pratiques ajustées » matérialisées par les produits de projets. Par « produit », nous entendons ici tout artefact produit au cours du projet, « livrables » décrivant les produits d'une étape de la réalisation, mais aussi tout autres documents ou structures de données produites (alternatives, justifications, trace des validations, vérifications, etc.).

Dans ce contexte, un système la gestion de connaissances en IS devrait répondre aux exigences suivantes:

- La Systématisation des connaissances métiers : explicitation des connaissances implicites du domaine ou du secteur d'activité d'Ingénierie.
- La Capitalisation des processus décisionnels en cours de projets : représentation formelle des expériences cruciales générées dans un projet, car une grande partie des décisions de projet reste implicite ou, au mieux, existe sous une forme informelle et non structurée dans des dossiers de justification de définition (DJD).
- Le Support proactif des processus d'IS : Un système proactif a pour finalité de proposer les éléments de connaissances cruciaux permettant la résolution idoine du problème en cours et ce, en tenant compte de la situation d'ingénierie.

### III. Contributions

Les contributions principales de travaux de cette thèse s'articulent autour des deux volets suivants :

**Proposition d'un cadre ontologique générique pour la modélisation de connaissances d'Ingénierie Système** : débouchant sur une conceptualisation et une opérationnalisation multi-facette et multi-strate. Ce cadre ontologique a pour objectif de couvrir les diverses facettes décrivant les processus d'IS à savoir : la facette « système à faire » capturant explicitement une conceptualisation du domaine du problème et des exigences, du domaine fonctionnel, et du domaine physique (ou organique (assemblage, composants matériels ou logiciels.)). La facette « système pour faire » capturant explicitement le contexte d'un projet d'Ingénierie sous forme de processus, de ressources, et d'artéfacts.

**Proposition d'un modèle conceptuel de capitalisation et de partage de Connaissances Métiers Situées (CMS)** : En IS, la transformation d'un besoin en une proposition de système fait appel à de multiples activités intellectuelles faisant passer progressivement de concepts abstraits à la définition rigoureuse de produits/artéfacts. Une CMS capture ce processus hypothético déductif sous forme d'annotations sémantiques multidimensionnelle basées sur le cadre ontologique que nous proposons. Nous proposons un modèle de capitalisation de CMS constitué de quatre dimensions d'annotation : la dimension situationnelle, la dimension téléologique (but d'ingénierie), la dimension argumentative et la dimension décisionnelle. Le modèle de partage de CMS est formalisé par un appariement des représentations des dimensions situationnelles et téléologiques des CMS avec celles des représentations des situations de projets en cours. Les résultats du processus d'appariement sont ensuite organisés sous forme de treillis de CMS traduisant un une hiérarchisation des connaissances nécessaires et /ou potentiellement utiles dans une situation d'ingénierie.

### IV. Organisation du rapport

Cette thèse est organisée autour des trois parties suivantes :

#### **Première partie : De la Gestion de Connaissances en Ingénierie Système (IS) :**

La première partie présente une analyse de la problématique de Gestion de Connaissances en IS et dresse un état de l'art des travaux qui s'apparentent à cette problématique. Le premier chapitre présente les principes fondateurs des disciplines d'IS et de gestions de connaissances. Une synthèse des apports potentiels de la gestion de connaissances aux processus d'IS est présentée en fin de ce chapitre. Le deuxième chapitre est consacré à un état de l'art des approches de gestions de connaissances en IS. Ce chapitre propose une

nouvelle classification des travaux significatifs de gestion de connaissances selon une vision IS. Le troisième chapitre est consacré à l'analyse des fondements de la discipline de l'Ingénierie Ontologique et des classes de ses applications potentielles en IS.

La première partie est clôturée par un positionnement par rapport aux travaux analysés permettant de poser les principes fondateurs d'un cadre ontologique générique qui supporterait l'articulation des différentes propositions étudiées dans la littérature pour la modélisation, la capitalisation et le partage des connaissances en IS.

### **Deuxième partie : Proposition d'un Cadre ontologique générique de Modélisation, Capitalisation et de Partage de Connaissances Métiers Situées en Ingénierie Systèmes. :**

La deuxième partie est organisée en deux chapitres. Le quatrième chapitre est dédié à la conceptualisation d'un cadre générique de systématisation de connaissances métiers en IS dénommé OntoIS. Ce chapitre présente les modules ontologiques proposés pour systématiser les connaissances d'IS, en partant de l'extension d'une ontologie fondationnelle, vers une onto noyau d'IS puis vers des modules ontologiques qui se spécialisent selon le contexte d'ingénierie. Le cadre OntoIS est un soubassement cognitif pour de multiples applications i.e. plusieurs systèmes de gestions de connaissances en IS peuvent être bâtis en se basant sur ce cadre ontologique. Nous discutons des applications potentielles de ce cadre dans un contexte de capitalisation et de partage de connaissances d'IS.

Dans le cinquième chapitre nous nous focalisons sur une application possible du cadre OntoIS qui est de type capitalisation d'expériences et de décisions cruciales en cours de processus d'IS. Ces expériences sont dénommées Connaissances Métiers Situées (CMS). Ce chapitre décrit en particulier les modèles de capitalisation et de partage de CMS.

### **Troisième partie : Opérationnalisation et validation**

La troisième partie est organisée en deux chapitres. Le sixième chapitre présente une proposition d'opérationnalisation des modèles de connaissances et de raisonnement à base du formalisme des Graphes Conceptuels(GC). Le cadre ontologique OntoIS est exprimé sous la forme de support étendu de GC, le modèle conceptuel de capitalisation de CMS est exprimé sous la forme de graphe conceptuels emboîtés typés et le modèle partage de CMS est fondé sur les opérateurs de raisonnement offerts par ce formalisme, en particulier la projection de GC. Le septième chapitre est consacré à une étude de cas dans le domaine des systèmes de défense. Le cadre ontologique générique a été spécialisé pour systématiser les connaissances métiers du domaine des missiles selon trois facettes : facette contextuelle, facette fonctionnelle et facette organique. Pour illustrer le processus de gestion de connaissances décisionnelles de projets d'IS, des scénarios de capitalisation et de partage de Connaissances Métiers Situées ont été implantés sur une plate-forme de gestion de Graphes Conceptuels (GC).

# **Première partie**

## **De la Gestion de Connaissances en Ingénierie Système (IS)**

# **Chapitre I : Ingénierie Système et besoins en Gestion de Connaissances**

---

## **Plan du chapitre**

---

- I.1 Introduction
- I.2 De l'Ingénierie Système
  - I.2.1 Définitions préalables
  - I.2.2 Ingénierie Système et processus
  - I.2.3 L'IS : un processus inhérent à l'innovation et centré sur les connaissances
- I.3 De la Gestion de Connaissances
  - I.3.1 Typologies de connaissances
  - I.3.2 Approches de Gestion de Connaissances
- I.4 Gestion de Connaissances et Ingénierie Système
  - I.4.1 Analyse des environnements existants supportant les processus d'IS
  - I.4.2 Fonctionnalités attendues d'un Système de Gestion de Connaissances en IS
- I.5 Conclusion

## I.1 Introduction

Les travaux de cette thèse se focalisent sur la problématique de Gestion de Connaissances en Ingénierie Système (IS), il convient alors d'analyser les principes fondateurs<sup>5</sup> de ce métier ainsi que ses spécificités par rapport aux autres activités de résolution de problèmes. Ce chapitre débute par la présentation du domaine de l'IS au travers des activités qui constituent les processus d'Ingénierie. La réalisation de ces activités nécessite l'utilisation de la part des Ingénieurs Systèmes, de leurs connaissances et savoir-faire métier. Nous étudions dans ce contexte les principales approches de Gestion de Connaissances Organisationnelles et leurs apports potentiels aux processus d'IS.

## I.2 De l'Ingénierie Système (IS)

Construire ou utiliser un objet technique complexe fait appel à la notion de système. Cette notion ancienne dans les sciences physiques et humaines, est maintenant courante dans les pratiques industrielles et opérationnelles. En la structurant en un corps de connaissances l'Ingénierie Système (IS), dans la suite du document a produit un mode de pensée et une façon d'appréhender les affaires par une approche structurée pour passer du besoin à la solution [Meinadier, 2002]. Avant de décrire cette approche, nous commençons par une présentation des définitions de base de la discipline d'IS.

### I.2.1 Définitions préalables

Pour pénétrer dans l'univers de l'IS, il est nécessaire de comprendre ce qui caractérise les concepts de système, d'ingénierie et d'Ingénierie Système.

– *Définition du concept de Système*

Avant de proposer une définition de ce qu'est un « système », faisons remarquer que les définitions dans la littérature ne manquent pas à cet égard. Etant nombreuses et diverses, nous en sélectionnons les suivantes que nous estimons plus pertinentes tant pour notre problématique que pour notre contexte :

- Un système est un ensemble de composants inter-reliés qui interagissent les uns avec les autres d'une manière organisée pour accomplir une finalité commune. [NASA, 1995]
- Un système est un ensemble intégré d'éléments qui accomplissent un objectif défini. [INCOSE<sup>1</sup>, 2004].

---

<sup>1</sup>International Council On systems Engineering: <http://www.incose.org/>

Pour notre part, nous caractérisons les systèmes conçus par l'homme en les définissant comme suit : « *Un système est un ensemble de composants qui collaborent à la réalisation d'un ensemble de tâches en vue de fournir un ensemble de services, cet ensemble est soumis à un environnement donné et interagit ainsi avec un sous-ensemble des éléments de cet environnement* ».

– **Définition du concept d'Ingénierie :**

Il faut distinguer clairement trois concepts clés utilisés de manière interchangeable en IS. [AFIS, 2009 a],[Meinadier, 2002]

**Les génies :** regroupant les méthodes, modèles, techniques et outils d'un métier permettant de réaliser des produits, par exemple le génie logiciel pour la réalisation de logiciels.

**L'ingénierie :** regroupant les méthodes, modèles, techniques et outils permettant de concevoir des systèmes impliquant la mise en œuvre de plusieurs génies.

**L'intégration :** regroupant les méthodes, modèles, techniques et outils permettant de réaliser des systèmes par assemblage de systèmes.

Ces trois concepts interviennent dans le métier qu'il est convenu d'appeler aujourd'hui Ingénierie Système [AFIS, 2009 a].

– **Définition de l'Ingénierie Système (IS)**

L'Ingénierie Système est définie dans [ANSI/EIA 632] ainsi : « L'Ingénierie Système est une approche interdisciplinaire qui englobe tous les efforts techniques pour développer et vérifier un ensemble de solutions relatives aux systèmes, aux utilisateurs et aux processus dans un cycle de vie total et intégré pour satisfaire des besoins client ».

L'Ingénierie Système réunit [Turki, 2008] :

- Les efforts techniques reliés au développement, fabrication, vérification, déploiement, opérations, support, mise à disposition et formations d'utilisateurs des produits et processus d'un système.
- La définition et la gestion de la configuration d'un Système.
- Les processus de gestion de projet
- Le développement d'informations pour la prise de décisions »

Selon le DoD (Department of Defence): « L'Ingénierie Système est un processus appliqué par une équipe en vue de transformer des besoins client en un système effectif ». Cette définition limite l'Ingénierie Système aux activités techniques.

Nous nous référons dans cette thèse à la définition proposée dans [Meinadier, 2002]

Selon [MEINADIER,2002] l'Ingénierie Système est un processus collaboratif et interdisciplinaire de résolution de problèmes s'appuyant sur les *connaissances, méthodes et techniques* issues des **sciences** et de l'**expérience** mise en œuvre pour définir un système qui satisfasse un besoin identifié.

Deux types de systèmes sont généralement distingués en IS [AFIS, 2009 b] :

- Les systèmes à dominantes technologiques développés par l'IS, on parle alors de « *système ingéniérisé* » ou de « système à faire » ou encore de « système étudié » : ensemble de sous-systèmes et constituants (matériels, logiciels, organisations et compétences humaines) et de leurs interfaces, sièges des interactions recherchées, organisés pour répondre à un besoin.
- Les systèmes à dominante organisationnelle mis en œuvre pour réaliser cette ingénierie de systèmes, on parle alors de « système ingéniérisant » de « système pour faire » ou encore de « projet » : ensemble d'équipes, de méthodes et de processus permettant de concevoir, produire, vérifier, distribuer, déployer, exploiter, maintenir en condition opérationnelle et retirer du service le système.

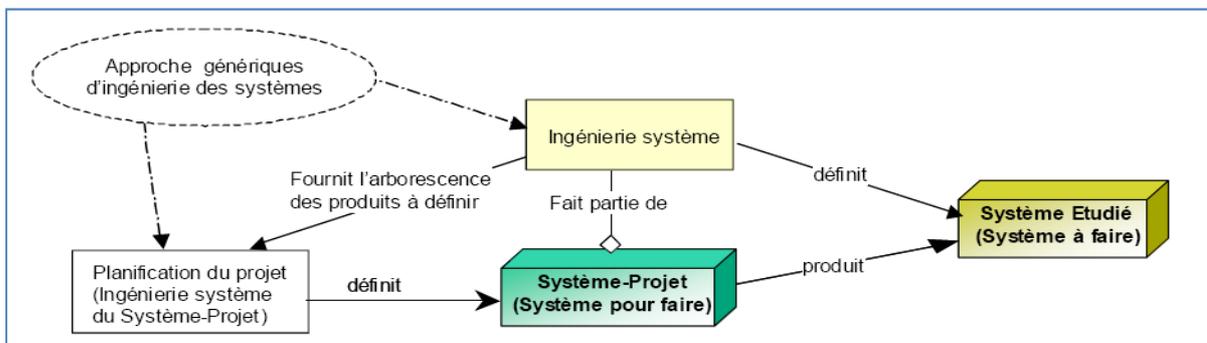


Figure I.1 Relations entre IS, système à faire et système pour faire [AFIS, 2009 b]

## I.2.2 Ingénierie Système et processus

Cette section a pour but d'analyser les concepts de base de notre objet d'étude dans le cadre de cette thèse : les processus d'IS. Dans un métier aussi diversifié que l'IS, il a été constaté d'une part, que les types d'activités à réaliser constituaient des invariants par rapport aux différents types de projets et de secteurs d'activités et d'autre part que les organismes d'IS passaient par les mêmes étapes pour progresser dans la maîtrise de ces activités. [AFIS, 2009 a].

Ainsi, une certaine formalisation générique des pratiques du métier d'IS s'est fondée sur les processus qui enchaînent ces activités. Cette formalisation se traduit dans des normes définissant les processus d'IS et leurs activités, et aussi dans des modèles de maturité<sup>6</sup> permettant d'évaluer la capacité d'un organisme d'IS à maîtriser son métier en fonction de sa maîtrise de ses processus.

### I.2.2.1 Le concept de processus en IS

Selon l'Association Française de l'Ingénierie Système (AFIS) Un processus est défini comme *« un ensemble partiellement ordonné d'activités ou de sous-processus corrélés et interactifs qui transforment des éléments d'entrée en élément de sortie. Cela faisant, il répond à sa finalité (mission) dans un environnement donné. Il est soumis à des contraintes et consomme des ressources. Chaque processus est déclenché par l'occurrence d'au moins un événement »* [AFIS, 2009 a]. La figure I.2 présente le modèle générique d'un processus.

Suivant ce modèle générique, un processus est un ensemble d'activités interactives coordonnées pour transformer progressivement des éléments d'entrée en produits. Les éléments d'entrées peuvent être de différentes natures, il peut s'agir d'énergie, de matière ou d'information. Chaque activité est une action de transformation d'une entrée en sortie. Cette transformation est réalisée en utilisant un sous-ensemble de ressources (humaines, matérielles, logicielles, méthodologiques).

---

<sup>6</sup> Capability Maturity Model : <http://www.afis.fr/praut/eval/eval5.htm>



Figure I.2 Modèle générique d'un processus [FIORESE, 2005]

Plusieurs contextes de définition des processus ont été distingué [FIORESE, 2005]

- Le **processus normalisé** : défini dans des normes, il décrit les types d'activités à réaliser et de résultats attendus de ces activités. Les normes résultent d'un certain consensus entre des experts des entreprises les plus avancées,
- Le **processus défini** ou **institutionnalisé** dans un organisme : résultant souvent de l'ajustement d'une norme aux besoins de l'organisme, il intègre les bonnes pratiques acquises et les méthodes choisies par l'organisme pour réaliser les activités. La conception et l'amélioration des processus institutionnalisés relève de l'**ingénierie des processus**,
- Le **processus ajusté** à un projet : c'est une adaptation d'un processus institutionnalisé aux justes besoins du projet. Il prend en compte les **contraintes** du projet et en consomme des **ressources**. Cet ajustement est un des éléments de la planification du projet dont les tâches techniques sont généralement des instances d'activités des processus appliquées aux éléments de l'arborescence des produits à réaliser (constituants du système et produits contributeurs) : chacun de ces éléments doit être spécifié, conçu, intégré, vérifié et validé).

Pour maîtriser la complexité du métier d'IS, ces processus sont généralement organisés en phases selon des modèles de cycle de développement de système. Nous en présentons dans la section suivante deux modèles que nous estimons pertinents dans notre contexte d'étude.

### I.2.2.2 Modèles du cycle de développement en IS :

Une multitude de cycles de développement ont été proposés dans la littérature, citons à titre d'exemple : le modèle en cascade [Royce, 1970], le modèle en V [Forsberg, 1995] et le modèle du cycle en spirale de Boehm [Boehm, 1988].

Nous présentons dans ce qui suit le modèle en V qui constitue un modèle de référence en IS.

**Le modèle en V** : le modèle en V est composé principalement de trois phases [Forsberg, 1995]:

- a) Une phase de conception et de justification (branche descendante du V), faisant passer du besoin à une définition justifiée de la solution, de façon incrémentale et itérative, comportant d'une part une analyse des besoins et contraintes des différentes parties prenantes, débouchant sur la spécification du problème et une spécification des exigences auxquelles le système doit satisfaire. Et d'autre part, une conception du système, consistant à élaborer: l'architecture fonctionnelle qui est un arrangement de fonctions et leurs interactions, définissant le séquençement de leur exécution, les flux de données et de contrôle qui le conditionnent et les performances requises pour répondre aux exigences. l'allocation des fonctions sur les composants, préalablement identifiés, et l'architecture organique (technique) des constituants, supportant l'architecture fonctionnelle et définie par les exigences attribuées aux constituants et à leurs interfaces (qui supportent les interactions entre constituants). Le résultat est une spécification des exigences pour chaque constituant.
- b) Une phase d'acquisition ou de réalisation des constituants, sous-traitée aux métiers spécifiques concernés ou à des fournisseurs. Le développement des constituants complexes suit également un cycle de développement.
- c) Une phase d'intégration et de validation (branche ascendante), comportant: La construction progressive du système par assemblage de ses constituants<sup>7</sup> en validant que leurs interactions confère bien les comportements attendus aux différents niveaux d'assemblage. Et l'intégration du système dans son environnement d'exploitation, suivie de sa qualification opérationnelle vérifiant qu'il répond au besoin opérationnel

L'Ingénierie Système (IS) fait référence à l'ensemble des deux branches du cycle de développement.

---

<sup>7</sup> Notons que la terminologie varie selon les auteurs ou les organismes d'IS pour désigner la décomposition d'un système. Le terme « constituant » est le plus générique, il est parfois remplacé par sous-systèmes, organes, modules, blocs constitutifs, composants, pièces, etc. selon le niveau hiérarchique de décomposition considéré

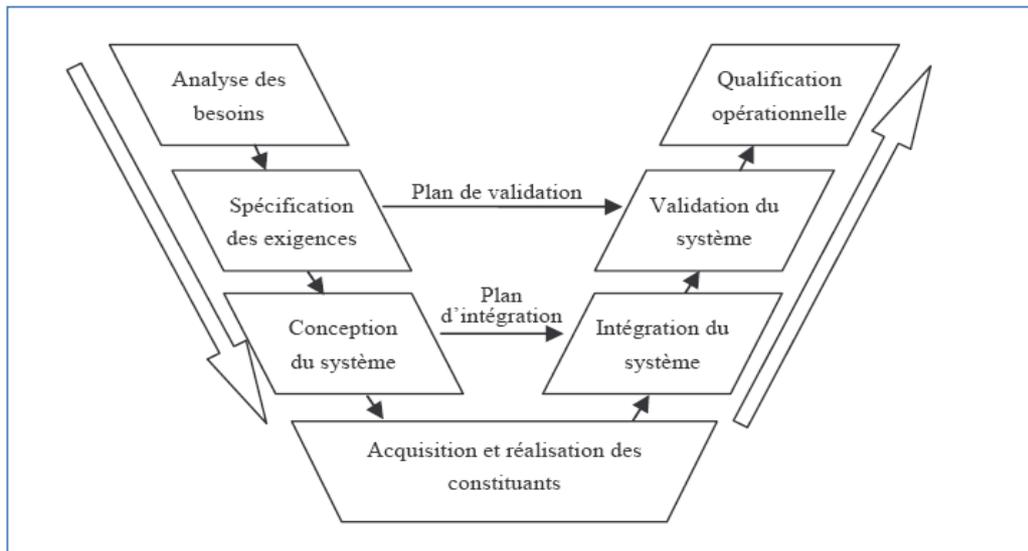


Figure I.3. Le cycle de développement en V [Forsberg,1995]

### I.2.2.3 Panorama des normes d'Ingénierie Système

Les normes d'Ingénierie Système [FIORESE, 2005] formalisent les processus et activités de l'IS, et classent les processus en quatre principales catégories :

- Processus techniques qui participent à la transformation des besoins en solution, (portée : de l'analyse des exigences jusqu'à la définition physique du système)
- Processus de management (du projet) qui participent à la maîtrise du projet,
- Processus contractuels qui assurent les relations avec le(s) client(s) et les sous-traitants du projet,
- Processus d'entreprise qui ont pour rôle de développer le potentiel en IS de l'entreprise en manageant les domaines communs aux différents projets d'IS.

Trois principales normes générales d'IS décrivant les processus du métier d'IS sont actuellement disponibles. Elles en définissent les types d'activité à réaliser et de résultat produit. Elles recouvrent des champs différents, de manière d'autant plus approfondie que leur champ est plus limité et, ainsi se complètent.

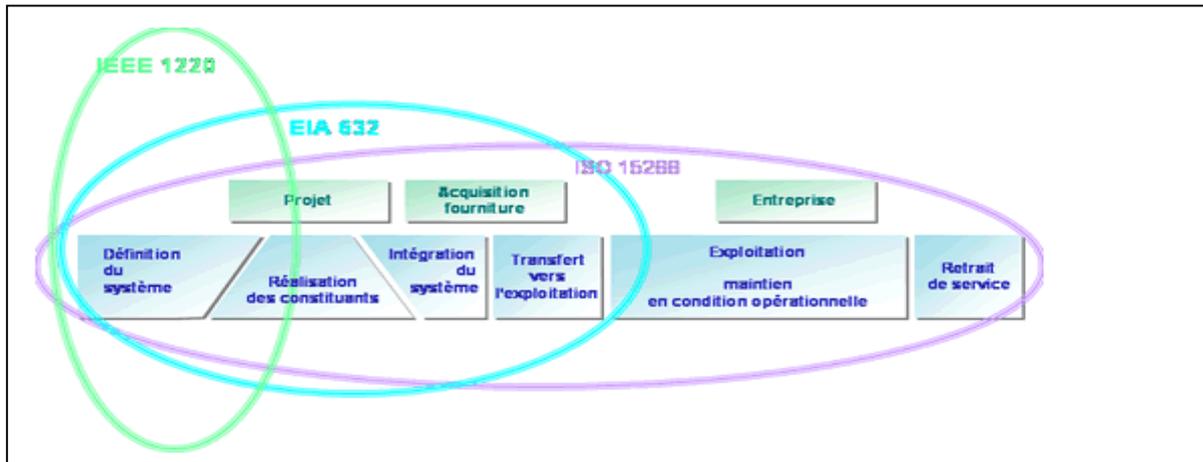


Figure I.4 Couverture respective des trois normes par rapport aux différents types de processus d'IS [AFIS, 2009]

L'objet d'étude retenu dans cette thèse est la catégorie « processus technique ». Les autres processus d'entreprise ou de management de projets décrits dans les autres normes d'IS sont en dehors de la portée de notre réflexion dans ce travail. Ainsi, la norme IEEE1220, relatives aux processus techniques d'IS est plus amplement détaillée dans ce qui suit.

La norme IEEE 1220 (**Standard for application and Management of the Systems Engineering Process**), dont la version originelle date de 1994, est issue du standard militaire MIL STD 499B, distinguent cinq processus : l'analyse des exigences, l'analyse fonctionnelle et allocation, la synthèse, l'analyse système et la maîtrise de l'IS.

Les trois processus d'analyse des exigences, d'analyse fonctionnelle et allocation et de synthèse, finement détaillés, comprennent chacun leur sous-processus de vérification ou de validation.

Le processus d'analyse des exigences raffine les besoins définis dans le cahier des charges (définissant les fonctions de services et les contraintes globales d'un système à développer) en vérifiant leur faisabilité et rajoutant les exigences<sup>8</sup> des parties prenantes concernées par l'ingénierie du système.

Le processus d'analyse fonctionnelle qui conçoit l'architecture fonctionnelle du système constituée de réseau de sous fonctions transformant des flux et des processus de ce réseau lui permettant de reconstituer les fonctions de services du système avec leurs scénarios opérationnels.

<sup>8</sup> Exigences fonctionnelles spécifiant les fonctions de services du système en termes de transformation de flux, de scénarios d'échanges avec l'environnement, de performances et d'autres parties exigences non fonctionnelles, traduction des contraintes globales du cahier des charges tel que des exigences de maintenabilité ou de testabilité

Le processus de synthèse qui conçoit la spécification de la solution sous forme d'une architecture physique de constituants, dument spécifiés par leurs exigences détaillées d'interfaces et de réalisation, supportant l'architecture fonctionnelle satisfaisant l'ensemble des exigences système. Cette architecture physique est une projection de l'architecture fonctionnelle sur des constituants physiques. [Meinadier, 2002].

Le processus d'analyse système a pour but d'analyser dans un cadre pluridisciplinaire les problèmes (conflits d'exigences ou solutions alternatives) issus des processus principaux afin de préparer les décisions.

Le processus de maîtrise de l'IS (control, en anglais) concerne tout particulièrement la gestion technique de l'Ingénierie Système et la maîtrise de l'information tant du système que du projet.

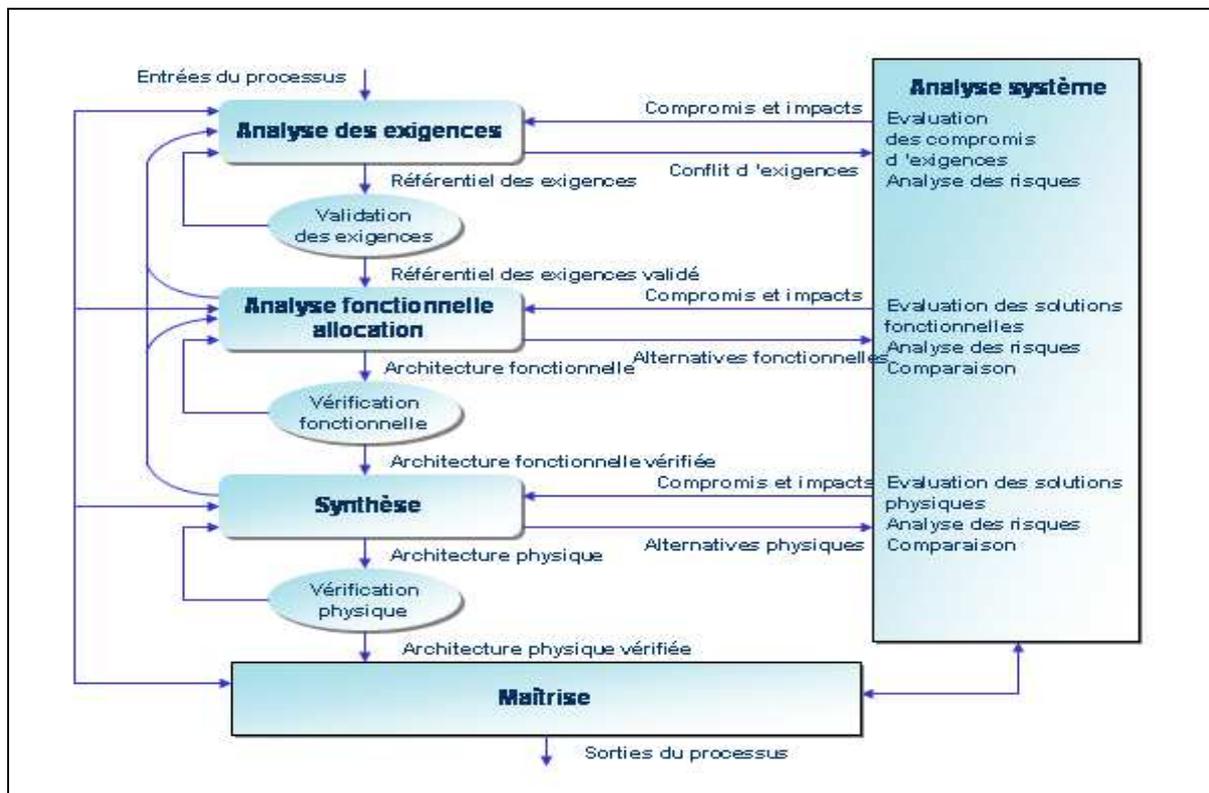


Figure I.5 Les processus d'IS selon la norme IEEE 1220

### I.2.3 L'IS : un processus inhérent à l'innovation et centré sur les connaissances.

Aujourd'hui, Les ressources organisationnelles critiques ne sont plus matérielles mais plutôt informationnelles, cognitives et intangibles. Les marchés demandent de plus en plus d'innovation, de créativité et cela avec des temps de conception de plus en plus réduits. Ceci exige donc un échange toujours plus rapide et généralisé des savoir-faire de chacun des acteurs de l'organisation [Darses, 2002]. Dans cette problématique générale, l'Ingénierie Système ne fait pas exception. En effet, les processus l'IS ont été qualifiés dans la littérature [Zdrahal et al. 2003] de processus centrés sur les connaissances (*En anglais, Knowledge Intensive Process*) car ils présentent les caractéristiques suivantes :

- **Processus multi-métiers et complexes :** Les situations d'ingénierie font appel à un collectif hétérogène, du fait des différents profils et des différents métiers des ingénieurs systèmes. On associe d'ailleurs au processus d'ingénierie les futurs utilisateurs qui intervenaient auparavant en aval du processus et qui sont aujourd'hui totalement intégrés au développement du système. Ce collectif se doit de collaborer à la réalisation d'une production cognitive (un système). Dans un tel contexte, chaque acteur fournit une description du système ingénierisé en utilisant sa propre terminologie dépendante de son point de vue métier (i.e mécanique, logiciel). Pour permettre la collaboration entre ces différents acteurs, il est nécessaire de se conformer à une représentation sémantique commune des concepts et des relations caractérisant les métiers d'ingénierie.

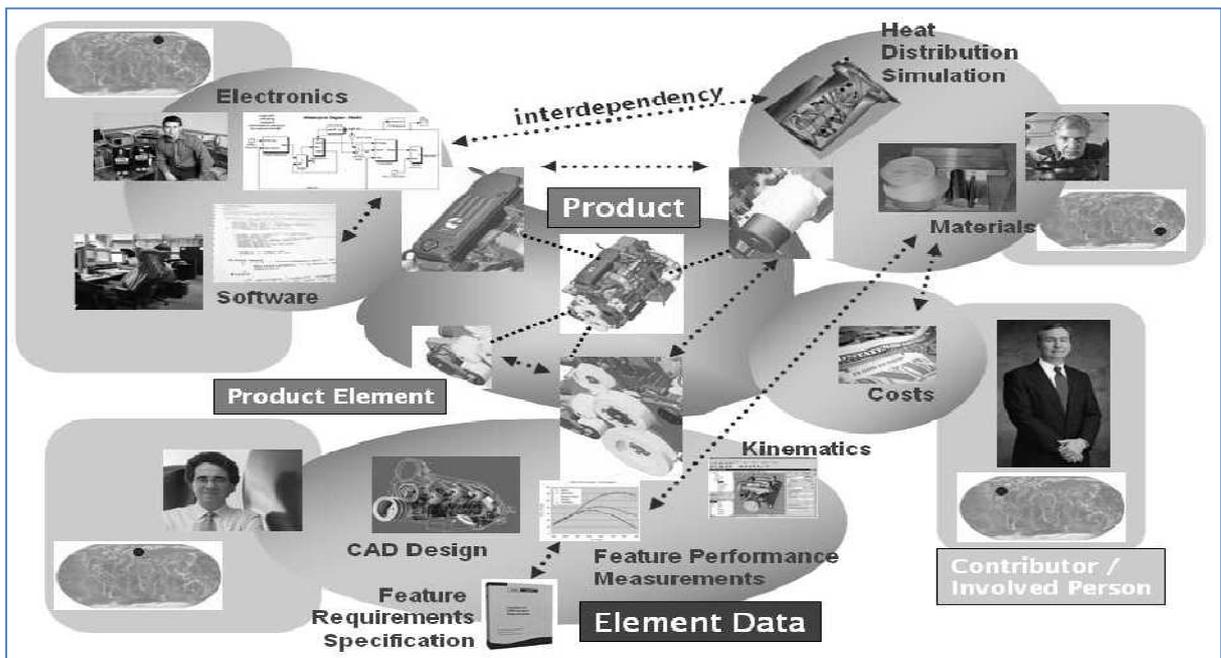


Figure I.6 Processus multi-métiers et complexes [Keller et al, 2007]

- **Processus créatifs et non déterministes:** Les processus techniques d’IS revêtent un caractère créatif et propositionnel. Ces processus peuvent être caractérisés comme des problèmes partiellement définis (*en anglais, ill defined problems*) [Zdrahal et al., 2002] en effet, dans en cours d’un processus d’IS le problème et la solution sont souvent explorés et évoluent en même temps. De ce fait, ces processus ne sont planifiables qu’à un degré de granularité large. Comme nous l’avons présenté dans les processus techniques d’IS (section I.2.3), plusieurs degrés de liberté sont permis au cours de l’ingénierie d’un système. Ainsi, l’évolution des contextes, la recherche progressive d’optimisation, l’émergence de nouvelles exigences, les conséquences des choix techniques, les non-conformités détectées en vérification et validation conduisent à des rétroactions sur les activités du processus qui se trouve ainsi parcouru plusieurs fois. En situation de résolution de problèmes, l’investissement intellectuel des Ingénieurs Systèmes est souvent considérable. Les décisions et les choix d’ingénieries retenues dans de tels cas sont importants à capitaliser afin de limiter l’effort à fournir lorsque des situations similaires se présenteront. Il est alors nécessaire d’assister les Ingénieurs Systèmes dans leurs tâches de résolution de problèmes en leur évitant de produire de nouveaux efforts impliquant un niveau d’expertise élevé. Pour cela, les connaissances qu’ils manipulent doivent être mémorisées tout en étant facilement réutilisables. Ce qui amène à poser les deux problèmes de recherches suivants : Comment formaliser les connaissances candidates à la réutilisation ? et Comment retrouver ces connaissances validées?

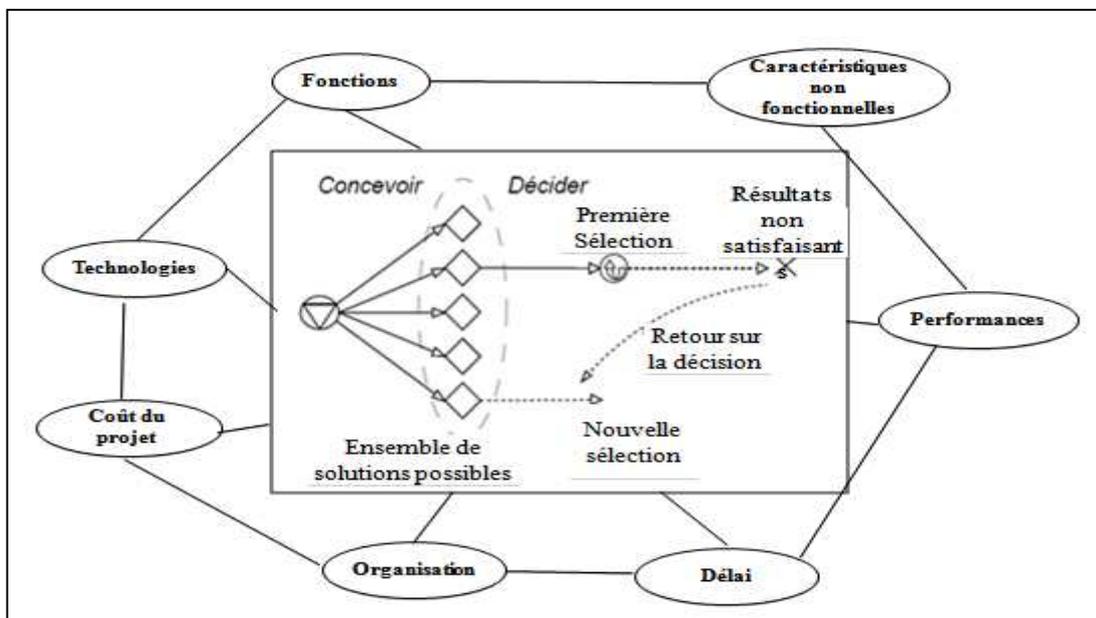


Figure I.7 Processus créatifs et non déterministes, d’après [Eynard, 1998]

Depuis quelques années déjà, la gestion des connaissances et des acquis de l’expérience apparaît comme un composant clé au sein des modèles d’amélioration des processus d’IS. Les

approches de maturité, telles que CMMI<sup>9</sup> ou SPICE<sup>10</sup> préconisent aux organismes d'IS de capitaliser leurs connaissances pour améliorer leurs processus et pour satisfaire au mieux les besoins de leurs clients.

En se référant au standard d'IS, IEEE 1220, l'amélioration continue de processus peut être adressée à travers :

- L'application d'un programme d'auto-évaluation déterminant le degré de maturité des pratiques d'IS comme le CMMI
- La capitalisation et le transfert d'expériences émergentes dans chaque projet.

Nous nous approfondirons dans le cadre de nos travaux le deuxième point pour l'amélioration des processus d'IS. Cette problématique, qui renvoie aux concepts de capitalisation et de partage de connaissances a été abordée dans les travaux de Gestion de Connaissances Organisationnelles dont les principaux concepts sont détaillées dans le paragraphe suivant. Cette étape est un préalable nécessaire à la proposition d'une démarche de gestion de connaissances pour le métier de l'IS.

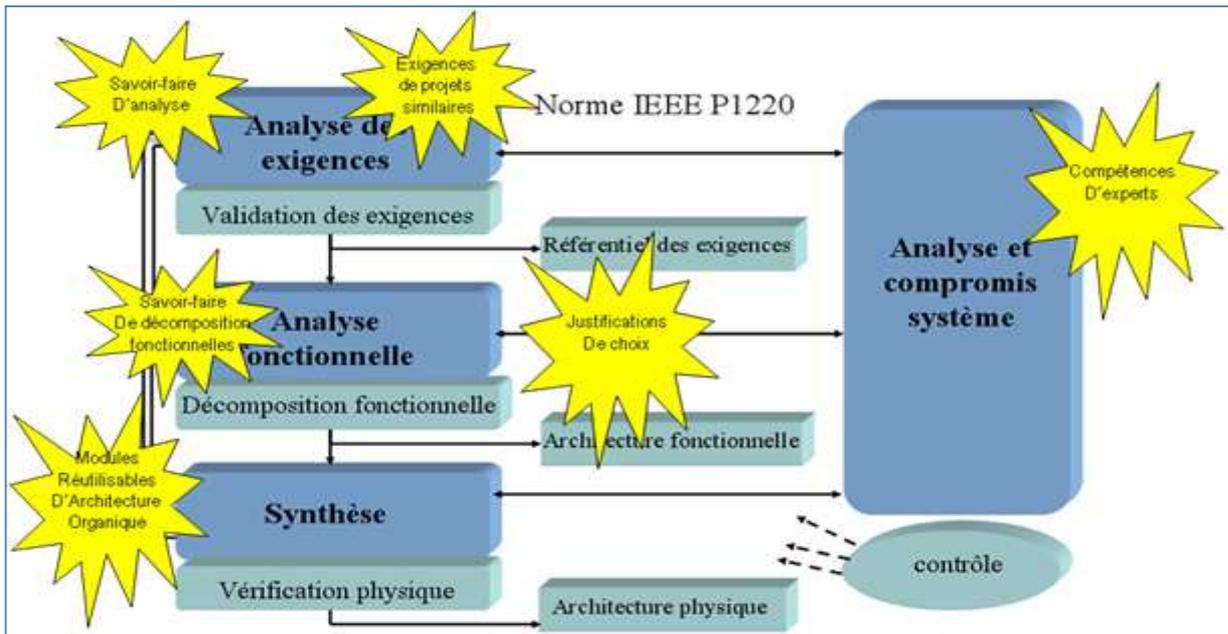


Figure I.8 Le processus d'IS vue comme un espace de création continue de connaissances

<sup>9</sup> Capability Maturity Model Integration . [www.sei.cmu.edu/cmmi](http://www.sei.cmu.edu/cmmi)

<sup>10</sup> Software Process Improvement and Capability Determination. [www.sqi.gu.edu.au/spice](http://www.sqi.gu.edu.au/spice)

## I.3 De la Gestion de Connaissances<sup>11</sup>

La Gestion de Connaissances (Knowledge Management (KM,) en Anglais), est un domaine de recherche pluridisciplinaire proposant des méthodes, des techniques et des outils assistant : la création, l'acquisition, le partage et l'utilisation des connaissances dans une organisation, en vue d'améliorer l'apprentissage individuel, collectif ou organisationnel [Grundstein, 2000]. [Prusak, 2001] souligne que ce domaine de recherche n'est pas un concept nouveau mais plutôt une intégration de concepts et techniques issus des domaines de la psychologie, la sociologie, de l'Intelligence Artificielle, de l'ingénierie de systèmes d'information, de l'économie, de la gestion des ressources humaines et du comportement des organisations.

Ainsi, plusieurs dimensions ou perspectives sont à prendre en considération lors de la mise en place d'une démarche de KM dans une organisation. Dans son étude sur le concept de KM [Wiig, 1999] considère les quatre dimensions suivantes :

- **Dimension Organisationnelle (*Enterprise effectiveness focus*)** : dont l'objectif est de définir la structure organisationnelle la plus adéquate pour supporter la création et le partage de connaissances,
- **Dimension Technique (*IM et IT focus*)** : qui supporte la capture, la formalisation et l'accès aux connaissances à travers la combinaison intégrée de technologies d'informations et d'ingénierie de connaissances,
- **Dimension Humaine (*People focus*)** : qui concerne la motivation des membres de l'organisation pour enrichir et réutiliser les connaissances,
- **Dimension capital intellectuel (*Intellectuel Asset focus*)** : qui met en œuvre des méthodologies de valorisation des actifs immatériels de l'organisation.

---

<sup>11</sup> Selon [Wilson, 2002], la gestion des connaissances est souvent confondue avec la gestion d'informations. Il montre alors que le monde industriel abrite sous le terme *knowledge management* un nombre important d'outils divers et variés, dans lesquels, il est uniquement possible de manipuler des informations . Wilson explique que la confusion, existante dans la littérature, entre connaissance et information est parfois la cause d'échecs de la gestion des connaissances. En effet, les approches uniquement centrées sur les aspects technologiques s'avèrent souvent peu efficaces par rapport aux coûts engagés dans les infrastructures informatiques auquel doit être ajouté celui de formalisation de connaissance et de remplissage.

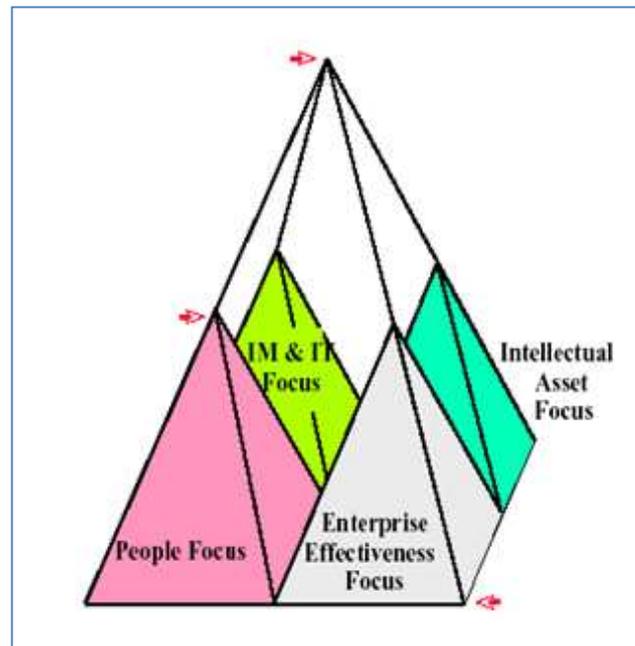


Figure I.9 Dimensions du KM [ Wiig , 1999]

### I.3.1 Définitions et Typologies de connaissances

Plusieurs définitions relatives au concept de connaissance ont été proposées dans la littérature. Nous en donnons ici, les définitions qui nous semblent les plus significatives dans le cadre de nos travaux.

**Définition 1** [Nonaka & Takeuchi, 1995 ] « *croyances vraies et justifiées* »

**Définition 2** [Sveiby, 1997] : « *Capacité d'agir dans un contexte* »

**Définition 3** [Davenport, 1999] « *la connaissance est un mélange d'expériences, de valeurs, d'informations contextuelles et de perspicacité d'experts fournissant un cadre d'évaluation et d'interprétation de nouvelles expériences* »

Afin de mieux appréhender ce concept, plusieurs typologies de connaissances ont été proposées dans la littérature :

Les sciences cognitives distinguent trois types de connaissances : déclaratives, procédurales et conditionnelles [Ben AHMED, 2007].

Les **connaissances déclaratives** correspondent aux connaissances théoriques, au "savoir", à la connaissance des faits, des règles, des lois ou des principes... Ce sont des connaissances qui sont définies d'une manière univoque (on les qualifie aussi de connaissance statiques). Elles sont généralement formulées dans un langage formel.

Les **connaissances procédurales** correspondent au comment de l'action, aux étapes, à la procédure de réalisation de l'action, au "savoir-faire". Ce sont des connaissances dynamiques qui ne peuvent se développer que dans un contexte d'action. Ce sont des connaissances qui

sont le résultat ou le fondement d'un processus de représentation et/ ou de raisonnement selon des procédures.

Les **connaissances conditionnelles** sont elles aussi des connaissances dynamiques, elles correspondent au pourquoi et au comment de l'action. Ce sont des connaissances stratégiques qui permettent de déterminer le moment et le contexte dans lequel il est approprié d'utiliser telle procédure, telle connaissance. Ce type de connaissance est une des caractéristiques de l'expertise. Les connaissances conditionnelles sont également *responsables du transfert des apprentissages*, c'est à dire de la capacité d'utiliser les connaissances dans un contexte différent de celui dans lequel elles ont été acquises. Il s'agit là d'une préoccupation centrale de la science cognitive. Le développement des connaissances conditionnelles est un des moyens de permettre ces transferts, en explicitant les conditions d'applications des connaissances ou procédures apprises.

En ergonomie cognitive, trois types de connaissances sont identifiées [Ben AHMED, 2007].

**Connaissances** : face à une nouvelle situation, les informations sont identifiées comme des *symboles* à partir desquels l'opérateur humain réalise des inférences en mettant en œuvre un modèle mental du système auquel il est confronté. Un comportement basé sur les connaissances est un *comportement cognitif* ;

**Règles** : les informations son identifiées comme des *signes* et il s'agit de mettre en œuvre des procédures d'exécution stéréotypées antérieurement acquises, sans choix complexe entre plusieurs alternatives. Il s'agit d'un *comportement procédural* ;

**Habilité** (« skills ») : l'information est saisie comme un signal qui déclenche une activité sensori-motrice automatisée acquise par la pratique. Il s'agit d'une *activité automatisée quasiment inconsciente*.

En **intelligence artificielle**, on distingue différents types de connaissances qui présentent certains points en commun avec les typologies présentées ci-dessus :

**Connaissance procédurale** : elle représente comment un problème est résolu. Elle indique comment réaliser une tâche ;

**Connaissance déclarative** : elle décrit ce qui est connu du problème (des énoncés décrivant les objets, les concepts, etc.) ;

**Méta Connaissance** : elle décrit la connaissance sur la connaissance. Par exemple, quelle connaissance est adaptée à un moment donné pour résoudre une tâche donnée ;

**Connaissance heuristique** : appelée aussi connaissance de surface. C'est une connaissance empirique acquise par un expert au fil de son expérience passée, elle sert à guider le raisonnement ;

**Connaissance structurée** : décrit un modèle mental de l'expert sous forme de structure (concepts, sous-concepts, objets, etc.).

Dans le domaine de la Gestion de Connaissances la typologie la plus utilisée est celle proposée par [ Polanyi, 1996] qui distingue deux catégories de connaissances

- **Connaissances Explicites (formalisables)** : elles peuvent être codifiées formalisées et transmissibles sous forme de documents réutilisables indépendamment du sujet qui les a spécifiées.
- **Connaissances Tacites (difficilement formalisables)** : ce sont des connaissances personnelles créées à partir des expériences et de l'intuition. Elles représentent les tours de main qu'un individu a acquis lors de son travail au fil des années. Ces connaissances sont difficiles à spécifier et à communiquer, car elles sont dépendantes de l'individu.

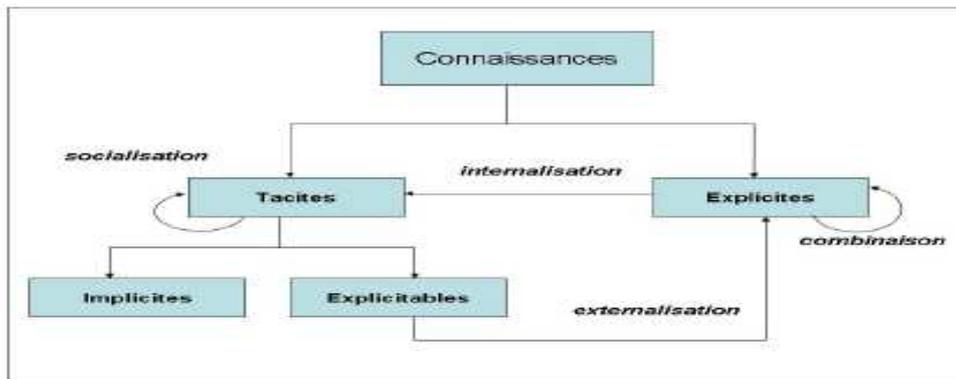


Figure I.10 Modèle de création de connaissances [Nonaka & Takeuchi, 1995]

Le modèle de [Nonaka & Takeuchi, 1995] sur l'apprentissage organisationnel repose ainsi sur l'analyse de modes de transfert des connaissances tacites aux connaissances explicites. La figure I.10 illustre le lien entre ces deux types de connaissances.

- Le lien de **socialisation** est celui de l'apprentissage, de l'intégration au sein d'un groupe d'un individu qui va en acquérir les connaissances tacites. Il ne peut pas être considéré ni être transmis en dehors de l'exercice d'une activité.
- Le lien de **combinaison** est celui qui permet d'associer des connaissances explicites d'individus différents, le vecteur étant la communication de l'information.
- **L'internalisation** correspond à l'enracinement de connaissances explicites dans des savoir-faire tacite. Ce mécanisme les relie au contexte de leur utilisation. on peut ainsi les comparer à des réflexes.
- **L'externalisation** permet de rendre explicites, en formalisant, des pratiques.

En fonction de ces différentes typologies de connaissances nous proposons dans la figure I.11 une classification des connaissances en IS.

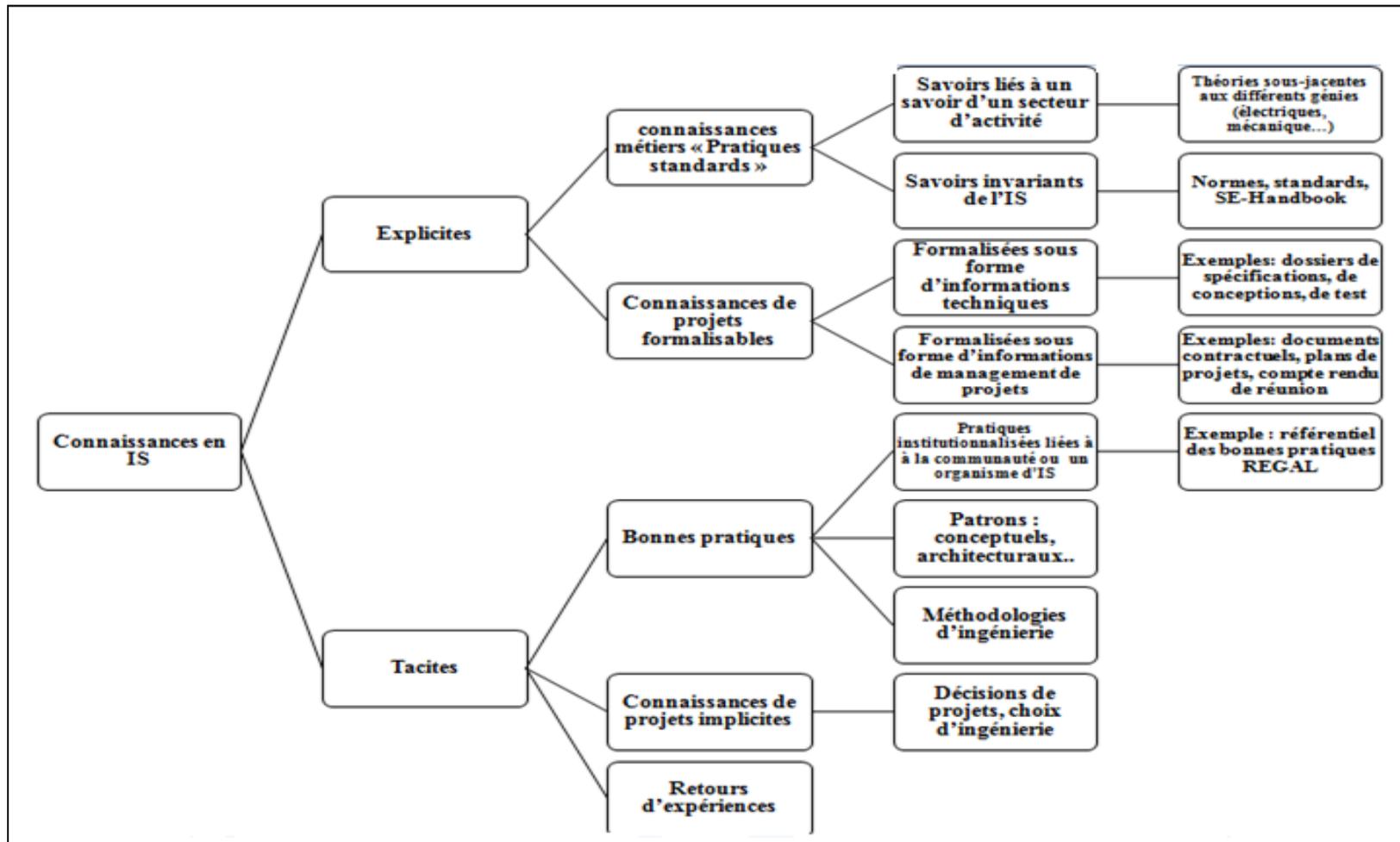


Figure I.11 Vers une proposition d'une typologie de connaissances en IS

### I.3.2 Approches de Gestion de Connaissances

Diverses classifications d'approches de KM ont été proposées dans la littérature [Zacklad et al., 2001] [Dieng et al., 2001]. Ainsi sont distinguées les approches : sociales et coopératives, ascendantes, descendantes, décisionnelles et organisationnelles.

Dans le cadre de nos travaux nous nous positionnons par rapport à la classification originellement proposée par [Hansen et al., 1999] et adoptée plus récemment par [McMahon, et al, 2004] pour étudier la Gestion de Connaissances en Ingénierie de Conception ( *en anglais, design engineering*). Ces auteurs distinguent deux approches de KM : les approches de *codification* et les approches de *personnalisation*.

**Les approches de codification :** Ces approches visent à transformer les connaissances implicites en connaissances explicites et sont ainsi basée sur des processus d'externalisation de connaissances [Nonaka & Takeuchi, 1995]. Les approches de KM ascendantes, descendantes, décisionnelles et organisationnelles font partie de cette catégorie. Ces approches sont dénommées aussi : approches de capitalisation des connaissances. Pour [Grundstein, 2000], « *capitaliser les connaissances, c'est considérer certaines connaissances utilisées et produites par l'entreprise comme un ensemble de richesses et en tirer des intérêts contribuant à augmenter la valeur de ce capital* ». Il caractérise la problématique de la capitalisation des connaissances par cinq facettes et leurs interactions, illustrées par la figure I.12. Cette approche de KM est souvent réifiée sous la forme de Systèmes de Gestion de Connaissances ou Systèmes de Mémoires Organisationnelles qui seront détaillées dans le chapitre II de cette thèse.

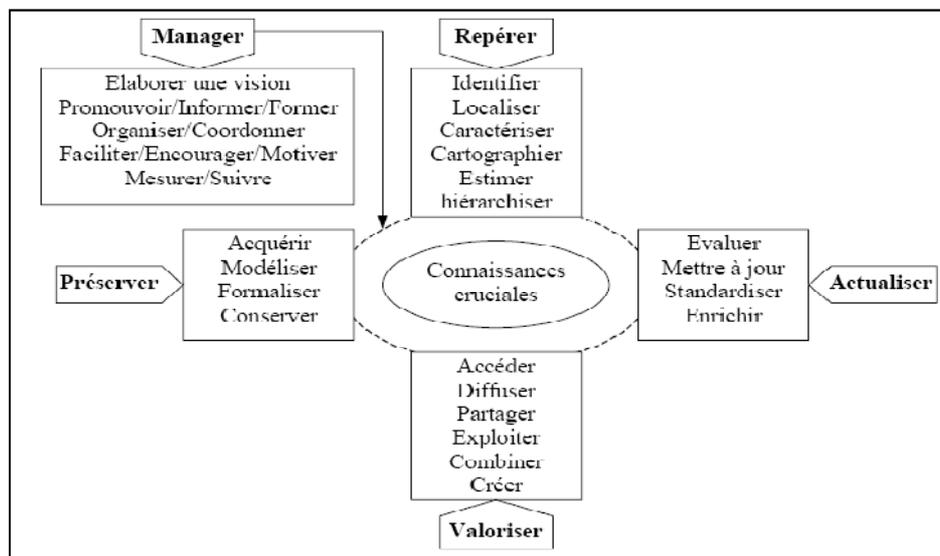


Figure I.12 La problématique de la capitalisation des connaissances dans les organisations [Grundstein, 2000].

**Les approches de personnalisation :** Les approches dites de personnalisation considèrent que les connaissances sont liées aux détenteurs de ces connaissances et qu'elles sont difficilement explicites. Ainsi ces approches sont principalement centrées sur le partage de connaissances au travers d'interactions directes entre acteurs (processus de socialisation [Nonaka & Takeuchi, 1995].). Les communautés de pratiques [Wenger, 1998] sont un exemple de ce type d'approches de KM. Pour [Wenger, 1998] les membres d'une communauté de pratique sont liés par leurs interactions et leurs activités communes. Les communautés de pratiques se caractérisent autour de trois dimensions : un *engagement mutuel*, une *entreprise commune* et un *répertoire partagé*. Leurs membres participent à des actions collectives dont ils négocient le sens les uns avec les autres. Les résultats de ces interactions créent des ressources appartenant au répertoire partagé de la communauté. Ces répertoires partagés peuvent permettre aux acteurs d'échanger des informations ou des conseils pour résoudre des problèmes précis.

Un Système de Localisation d'Expertise [Rus et al, 2001] ou *réseau de connaissances* [Bush et Tiwana, 2005]. (*en anglais, knowledge network*) est un autre exemple de ce type d'approche de KM. Ces systèmes proposent de gérer les connaissances d'une organisation en fournissant des « pointeurs » vers des personnes sources de connaissances pour d'autres. Ces systèmes sont basés sur des profils caractérisant les compétences des experts.

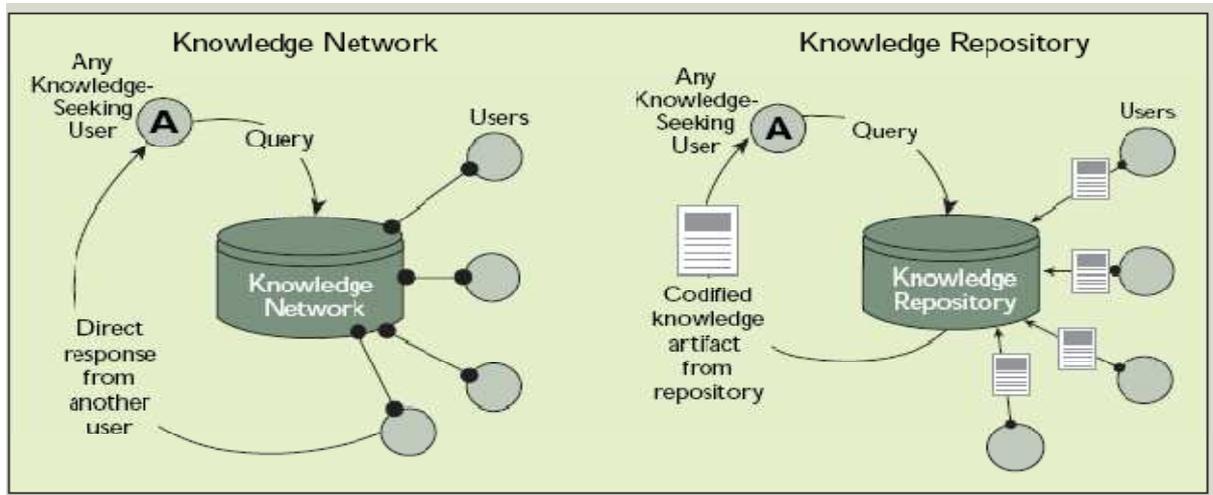


Figure I.13 Référentiel de Connaissances vs. Réseau de connaissances [Bush et Tiwana, 2005]

## I.4 Gestion de Connaissances et Ingénierie Système

Les processus techniques d'IS sont supportés par de multiples environnements et outils. Nous nous focalisons dans cette section sur les outils permettant de gérer les informations générées en cours de projets d'IS et nous analysons les points clés qu'il faut prendre en considération pour aboutir à des systèmes de gestion de connaissances en IS.

### I.4.1 Analyse des environnements supportant les processus d'IS

Une classification des outils supportant les processus techniques d'IS a été proposée dans [AFIS, 2009]. Nous proposons de nous retreindre ici au cœur du métier de l'IS en présentant les outils supportant le processus d'ingénierie des exigences et le processus de conception de systèmes :

- **Outils d'ingénierie d'exigences** [AFIS, 2009b]: le processus d'ingénierie d'exigences regroupe le processus de définition et le processus de gestion des exigences système. les outils d'ingénierie d'exigences assistent l'ingénieur système à la saisie, la dérivation ou l'allocation des exigences, à la saisie des liens d'association de chaque exigence avec sa source, avec sa justification, avec les hypothèses dans lesquelles elle est applicable. Les outils DOORS<sup>12</sup> et SLATE<sup>13</sup> sont des exemples de cette catégorie d'outils
- **Les outils de conception de systèmes** [AFIS, 2009b] : parmi les principaux outils de cette catégorie nous citons :
  - Les outils de modélisation systémique : permettant de représenter les aspects structurels, comportementaux et sémantiques d'un système (par exemple des outils de modélisation en SysML).
  - Les outils de modélisation analytique : destinés à évaluer les performances et la fiabilité des systèmes, simuler le comportement des systèmes et de proposer des optimisations de solutions de conception.

Ces outils génèrent des informations sur le système en cours de sa définition. Ces informations sont partagées par l'ensemble des participants à la réalisation du système qui l'utilisent pour générer de nouvelles informations. [Meinadier, 2002] distingue trois approches de gestion de ces informations de projets d'IS et constituant des supports de connaissances sur le projet :

- La base de données intégrée : ayant pour but de répertorier les données du projet d'IS et de tenir à jour leurs liens.

---

<sup>12</sup> DOORS: Dynamic Object Oriented Requirements System.

<http://www.telelogic.com/products/doorsers/doors/>

<sup>13</sup> SLATE: System Level Automation Tool for Enterprises. <http://www.sdrc.com/slate/>

- Les systèmes de gestion de documentation technique : comprenant l'ensemble des documents concernant le système et sont généralement constitués par extraction personnalisées de la base de données intégrée.
- Les systèmes de gestion de configuration : ayant pour objectif de maintenir à jour l'état de la définition du système pendant le processus de développement

Les environnements d'IS sont principalement destinés à modéliser un système selon plusieurs points de vue (exigences, fonctions, constituants) et à gérer les liens de traçabilités des informations générées en cours de projets. Ils n'ont pas été conçus pour des fins de gestion de connaissances d'ingénierie, dans le sens que la problématique d'accès et de partage de cette masse d'information n'a pas été adressée. Les informations de projets sont représentées sous forme de schémas, de modèles ou de documents. La réutilisation d'un artefact d'ingénierie dans le contexte d'un nouveau projet (par exemple, une architecture fonctionnelle, un modèle de conception, un choix d'ingénierie) consiste en une simple recherche par mots clés dans un système de gestion de documentation technique.

La préoccupation principale des systèmes de gestion de connaissances d'ingénierie [Skyman, 2000] [Mizoguchi, 2003] est de doter ces informations d'une représentation formelle pour rechercher et réutiliser plus pertinemment dans le contexte de nouveaux projets.

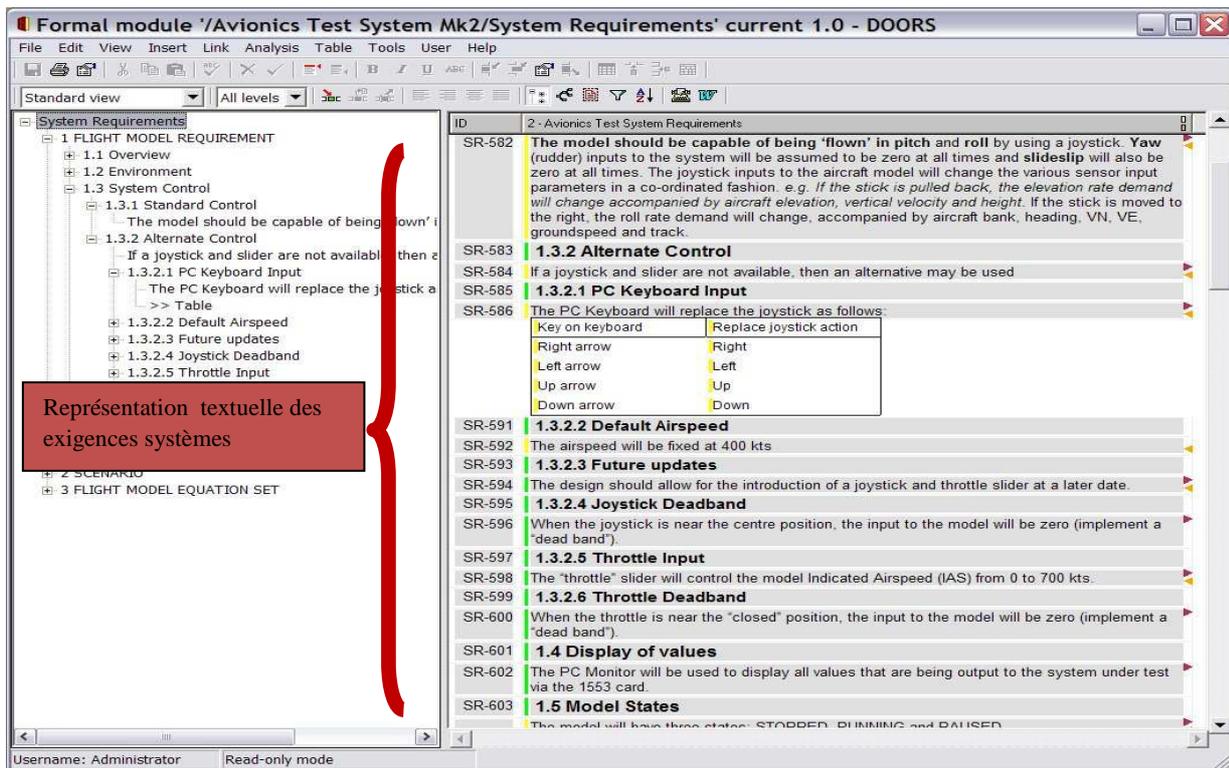


Figure I.14 : Capture d'écran d'un outil de gestion d'exigences. [U'Ren , 2003]

## I.4.2 Fonctionnalités attendues d'un Système de Gestion de connaissances en IS

Nous pensons que la Gestion de Connaissances en IS doit s'inscrire dans le cadre d'un changement de paradigme d'une vision centrée informations vers une vision centrée connaissances. Nous synthétisons dans cette section les principes fondateurs d'une vision centrée connaissances appliquée aux processus d'IS

- P1 : La représentation explicite et consensuelle des connaissances liées à la dimension « système à faire »: L'IS fait appel à de multiples savoir-faire d'un métier ou d'un secteur d'activité d'ingénierie i.e. les fonctions, les constituants d'une classe de système. Ces savoir-faire sont généralement implicites et détenus par les experts d'un domaine. En représentant explicitement les concepts et les relations d'un métier d'ingénierie nous disposerons d'une signature fonctionnelle et relationnelle d'un langage formel de représentation et la sémantique associée ou, encore, de primitives non logiques d'un langage de représentation et la sémantique associée [Bachimont, 2000]. Cette représentation favorise le partage des connaissances métiers et aide à décrire formellement connaissances générées en cours de projets.
- P2 : La représentation explicite et consensuelle des connaissances liées à la dimension « système pour faire » : l'IS fait appel à un ensemble de compétences, de processus, de méthodes organisées pour répondre à un besoin d'ingénierie. En représentant explicitement ces éléments nous obtiendrons un modèle capturant clairement le contexte d'un projet d'ingénierie.
- P3 : La capitalisation des connaissances cruciales générées en cours de projets : un projet d'IS génère plusieurs artefacts comme les spécifications de systèmes, les choix et les décisions d'ingénierie. La description de ces artefacts par le biais de concepts et de relations tirées des modèles définis en P1 et P2 permet de capturer la sémantique précise de leurs contenus.
- P4 : Le support proactif des processus d'IS : Un système proactif a pour finalité de proposer les connaissances capitalisées permettant la résolution idoine du problème en cours et ce, en tenant compte de la description d'une situation d'ingénierie par le biais de concepts et de relations tirées des modèles définis en P1 et P2.
- P5 : Intégration avec les environnements existants d'Ingénierie Système : un système de gestion de connaissances ne doit pas perturber les pratiques actuelles des ingénieurs système. Il n'est pas destiné à remplacer les environnements d'ingénierie existants, mais doit s'interfacer avec ces environnements.

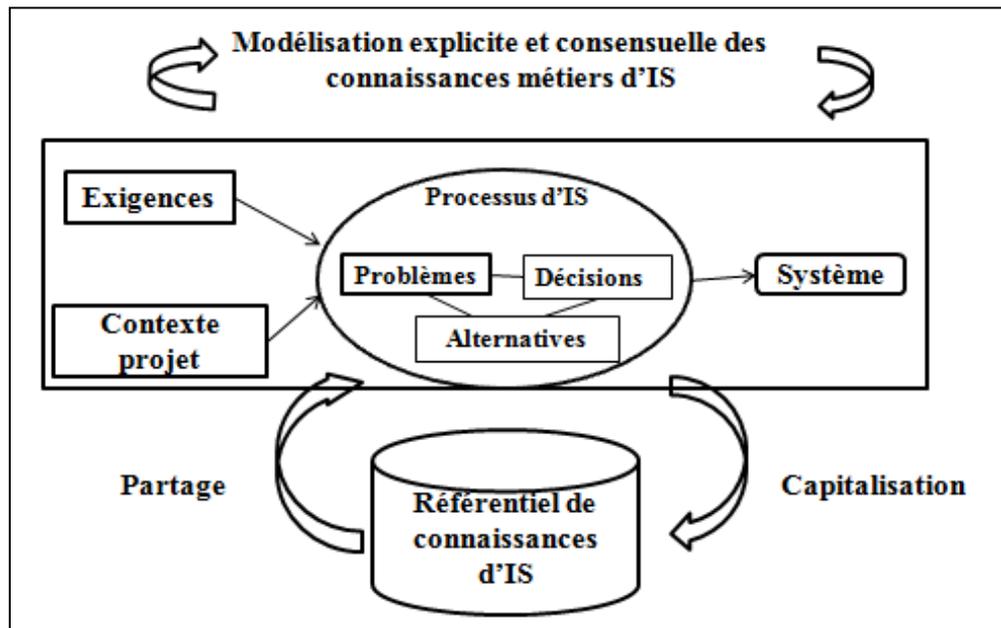


Figure I.15 Fonctionnalités attendues d'un système de gestion de connaissances en IS : modélisation, capitalisation et partage

## I.5. Conclusion

Dans ce chapitre nous avons vu que l'IS constitue un terrain privilégié pour la gestion de connaissances parce qu'il s'agit d'une activité complexe où la routine est mêlée à l'innovation et à la créativité. Nous avons étudié essentiellement les processus techniques d'IS, des processus complexes où la résolution coopérative de problèmes et la prise de décision est omniprésente.

L'analyse des environnements existants supportant les processus d'IS révèle que la gestion de connaissances reste encore des domaines peu explorés et peu formalisés. En fonction des exigences que nous avons déterminées pour un système de gestion de connaissances en IS, nous étudions les travaux de l'état de l'art selon deux volets : le premier volet analyse les systèmes de gestions de connaissances destinés à gérer les savoirs-faires métiers et les expériences de projets dans des domaines transposables à l'IS. Le deuxième volet étudie les apports potentiels de l'Ingénierie Ontologique à la modélisation consensuelle et au partage des connaissances d'IS.

# Chapitre II

## Systemes de Gestion de Connaissances en Ingénierie Système

---

### Plan du chapitre

---

- II.1. Introduction
- II.2. Catégorisation des systèmes de Gestion de Connaissances et Ingénierie Système
  - II.2.1 Typologies de Mémoires Organisationnelles
  - II.2.2 Typologie retenue pour analyser les travaux d'IS
- II.3 Systèmes de Gestion de Connaissances Métiers
  - II.3.1 Systèmes à Base de Connaissances (SBC)
  - II.3.2 Knowledge Based Engineering (KBE)
  - II.3.3 Lignes de produits (PLM)
- II.4 Systèmes de Gestion d'expériences
  - II.4.1 Systèmes à base de cas
  - II.4.2 Mémoires de projets
  - II.4.3 Référentiels de Conception
  - II.4.4 Systèmes de Gestion de connaissances orientées processus
- II.5 Synthèse

## II.1 Introduction

Après une présentation du processus d'émergence des connaissances dans les processus d'IS, ainsi que de la contribution de ces connaissances comme support à des situations de résolution coopérative de problèmes (chapitre I), nous étudions dans ce chapitre les apports potentiels des approches de Gestion de Connaissances aux processus d'IS.

Ayant introduit les concepts fondamentaux du domaine de la Gestion de Connaissances (KM), nous proposons dans ce chapitre de détailler les stratégies de KM dites de « codification »<sup>14</sup> réifiées sous la forme de Mémoires Organisationnelles (MO) (cf. II.2) en étudiant plus particulièrement leurs applicabilité aux différents métiers d'IS<sup>15</sup>. Les principaux objectifs de ce chapitre sont : -Recenser et caractériser un ensemble de travaux de KM représentatifs proposées dans la littérature, -Etablir une classification de ces approches afin d'identifier quelles facettes de la problématique de gestion de connaissances en IS semblent couvertes, et au contraire, celles qui ne le sont pas, - Fournir une synthèse articulant les différentes propositions pour répondre aux besoins de la Gestion de Connaissances en IS.

## II.2.Catégorisation des Systèmes de Gestion de Connaissances et Ingénierie Système

Le processus de gestion de connaissances d'une organisation s'articule principalement autour de trois points-clés : capitalisation, partage et création [Ermine, 2000]. La capitalisation vise à savoir d'où l'on vient, savoir où l'on est, pour mieux savoir où l'on va, c'est-à-dire recenser l'actif immatériel que représente le savoir et savoir-faire de chacun des acteurs de l'organisation. Le partage consiste à passer de l'intelligence individuelle à l'intelligence collective. Enfin, la création consiste à innover pour survivre. La mise en œuvre d'un tel processus selon une approche sociotechnique est souvent supportée par des Systèmes de Gestion de Connaissances (SGC). Il n'existe pas à notre connaissance aujourd'hui de définition précise de cette classe de système. Certains auteurs comme [Alavi et al., 2001] les considèrent comme une classe de Systèmes d'Information dédiée à la gestion des connaissances organisationnelles « *Knowledge Management Systems (KMS) refer to a class of information systems applied to managing organizational knowledge. They are IT-based systems developed to support and enhance the organizational processes of knowledge creation, storage, retrieval, transfer and application* ». Pour [Furst, 2004] les SGC permettent de sauvegarder, transmettre et partager entre plusieurs utilisateurs des connaissances sur support informatique, sont parmi les principaux domaines d'application des Systèmes à Base de Connaissances (SBC)<sup>16</sup>.

---

<sup>14</sup> Par opposition aux approches de personnalisation (chapitre I section I.3.2)

<sup>15</sup> Plus particulièrement des travaux du domaine de l'Ingénierie de Conception (Design Engineering)

<sup>16</sup> Détaillés dans la section II.3.1

[Maier, 2002] propose une synthèse intéressante des origines conceptuelles et techniques des SGC. Comme l'illustre la figure II.1, le concept de SGC (dénomé par différents termes dans la littérature, comme « Entreprise Knowledge Medium » (volet II) est le résultat d'une combinaison synergique de plusieurs techniques, par exemple des techniques d'Intelligence Artificielle (volet I) supportant les théories de la Gestion de Connaissances comme l'Apprentissage Organisationnel<sup>17</sup> (volet III), donnant lieu à des systèmes, des outils et des plateformes implémentant les stratégies de KM de codification et/ou de personnalisation. Plus précisément [Maier, 2002] distingue trois classes de fonctions<sup>18</sup> de SGC à savoir : les fonctions d'intégration, les fonctions d'interaction et les fonctions de liaison « *Bridging functions* ». Ces fonctions reflètent la classification des stratégies de KM de [Hansen, 1999] (cf chapitre I, section I.3.2). Les fonctions de liaison mettent en œuvre une approche mixte supportant à la fois des stratégies de codification et de personnalisation. Les fonctions d'interaction supportent les stratégies de personnalisation et sont principalement centrées sur le partage de connaissances au travers d'interactions directes entre acteurs et passent souvent un processus de socialisation. Les communautés de pratiques [Wenger, 1998] et les systèmes de localisation d'expertise [Rus et al, 2001] sont des exemples de ce type d'approches de KM. Les fonctions d'intégration supportent les stratégies de codification. Elles visent à « transformer » les connaissances implicites en connaissances explicites. Elles sont principalement basées sur des processus d'externalisation [Nonaka & Takeuchi, 1995] permettant de passer d'une mémoire de travail souvent individuelle à une Mémoire d'Entreprise ou Organisationnelle [Grundstein, 1995]. C'est cette approche que nous adoptons et que nous détaillons dans la suite de ce chapitre car les mécanismes de capitalisation de connaissances en IS restent encore des domaines peu explorés et donc peu formalisés, contrairement aux approches de personnalisation qui sont communs dans les pratiques de l'IS (Exemple : référentiel REGAL<sup>19</sup> pour le partage des connaissances de la communauté d'IS).

---

<sup>17</sup> Apprentissage Organisationnel : L'apprentissage organisationnel est la capacité, pour l'organisation, d'accroître, au fil du temps, l'efficacité de son action collective [Argyris et Schön 1978] ; théorie faisant référence aux connaissances, savoir-faire, techniques et pratiques diverses qu'une organisation peut développer

<sup>18</sup> Deux classes de fonction d'un SKM ont été originellement proposées dans [Zack, 1999] : fonctions d'intégration et fonctions d'interaction. [Maier, 2002] a ajouté les fonctions de liaison

<sup>19</sup> REGAL : [www.afis.fr](http://www.afis.fr)

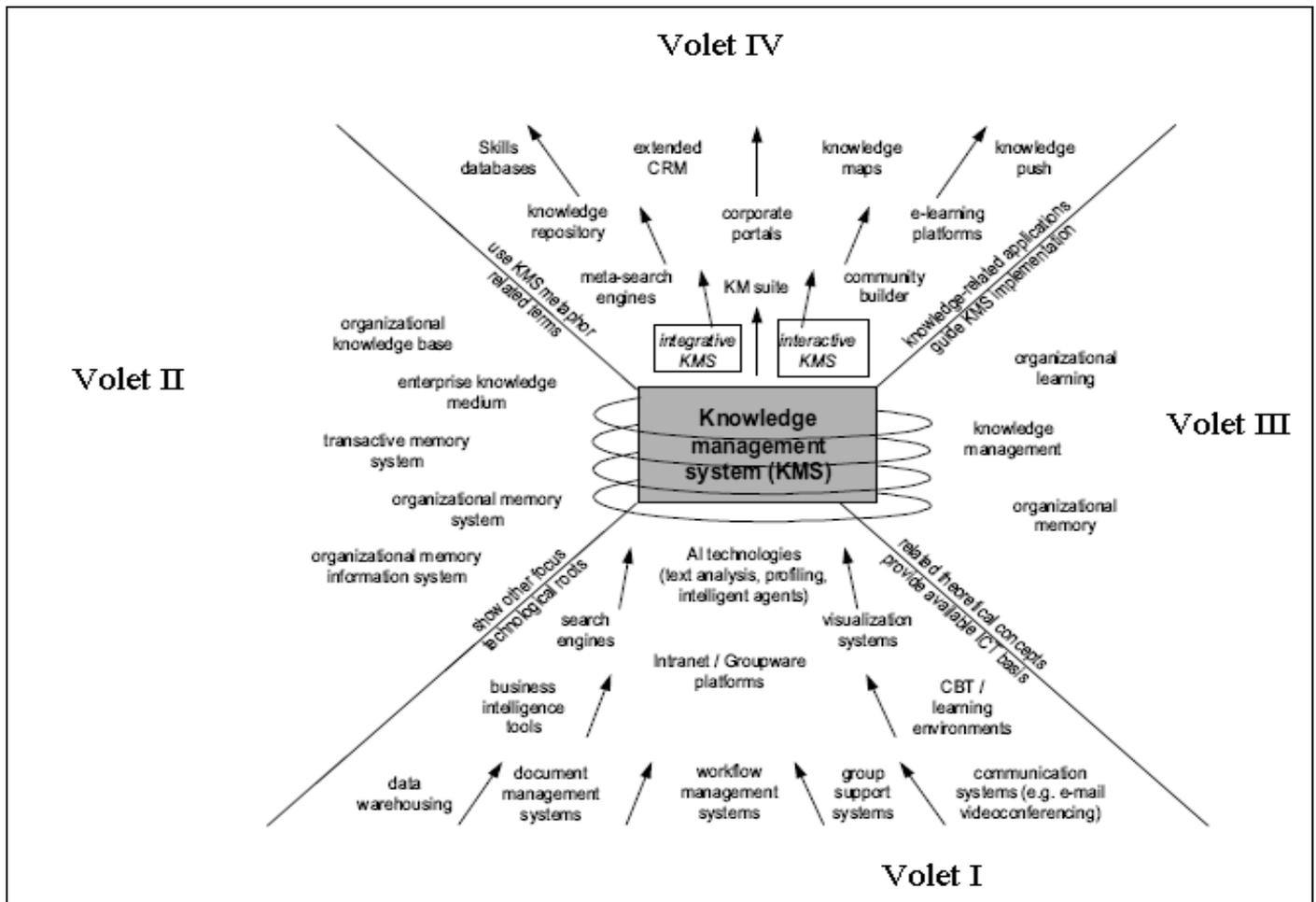


Figure II.1 : Fondements Conceptuels et Techniques des SGC.[Maier, 2002]

Dans le cadre de nos travaux nous retiendrons que selon une approche de codification, un SGC est un système dédié à la gestion d'une Mémoire Organisationnelle (MO) ce qui nous amène à nous focaliser sur les concepts de base d'une MO dans la suite<sup>20</sup>. Nous présentons une définition consensuelle qui constitue une bonne synthèse sur des définitions sur le concept de Mémoire Organisationnelle. Selon [Dieng et al, 1998] une Mémoire Organisationnelle (MO) est : *La matérialisation explicite, désincarnée et persistante des connaissances et informations cruciales dans une organisation, afin de faciliter leur accès, partage et réutilisation par des membres de l'organisation, dans le cadre de leurs différentes tâches individuelles ou collectives.* La figure II.2 illustre le concept d'un SGC dédié à la gestion d'une MO.

<sup>20</sup> Système de gestion de connaissances et système de gestion de mémoire organisationnelle sont utilisées d'une manière interchangeable dans la suite de ce chapitre

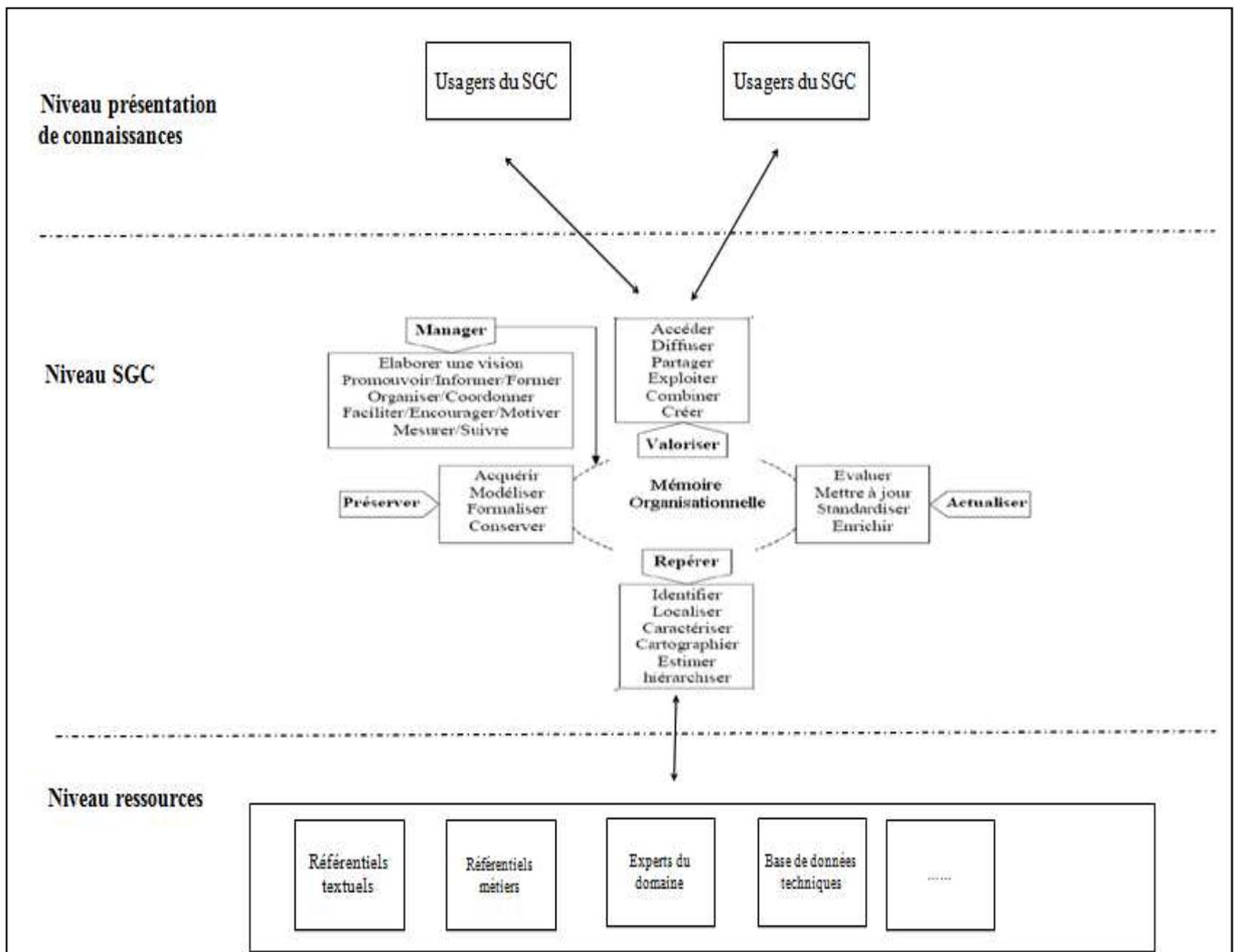


Figure II.2 Architecture d'un SGC à la gestion d'une MO. Adapté de [Grudstein, 2000], [Jatinder et al, 2008]

Une MO peut avoir différentes portées et différentes granularités, sa mise en place est fondée sur un ensemble d'étapes. Plusieurs modèles du cycle de vie de la mémoire organisationnelle ont été proposés dans la littérature [Dieng et al, 1998] [Heijst, 1996] [Abecker et al, 1998] [Barthès, 1996]. Ces modèles incluent communément les processus de détection de besoins, d'acquisition, de construction, de diffusion, d'évaluation et d'évolution. Nous présentons dans ce qui suit une description synthétique de chaque processus.

- **Détection de besoins** : consiste à identifier les problèmes à résoudre par une approche de KM. Pour la détection des besoins les recherches actuelles portent sur les modèles d'entreprises (en particulier les modèles centrés processus), les modèles d'apprentissage organisationnel et sur les méthodes centrés parties prenantes.
- **Acquisition** : au cours de cette étape les sources de connaissances existantes et potentielles sont identifiées et capturées sous une forme accessible.
- **Construction** : concerne le choix d'une technique de matérialisation (documentaire, base de connaissances, base de cas..) des composants de la mémoire organisationnelle.
- **Diffusion** : Les principaux problèmes à résoudre sont : l'organisation et l'indexation possibles de la mémoire pour améliorer sa diffusion, la recherche des éléments appropriés de la mémoire en réponse à une demande de l'utilisateur ou à une

dissémination proactive ("push") des éléments appropriés vers l'utilisateur et l'adaptation de la réponse à l'utilisateur, en particulier à sa tâche.

- **Evaluation** : Les travaux actuels se focalisent sur l'évaluation du capital humain d'une entreprise, sur les métriques dédiées à la mémoire d'entreprise, sur les méthodes et outils dédiés pour évaluer les performances d'une mémoire organisationnelle.
- **Evolution** : Pour la maintenance et l'évolution de la mémoire d'entreprise, les travaux actuels étudient les problèmes liés à l'ajout de nouvelles connaissances, à la suppression ou modification des connaissances obsolètes, ainsi que les problèmes de cohérence sous-jacents à l'extension coopérative de la mémoire.

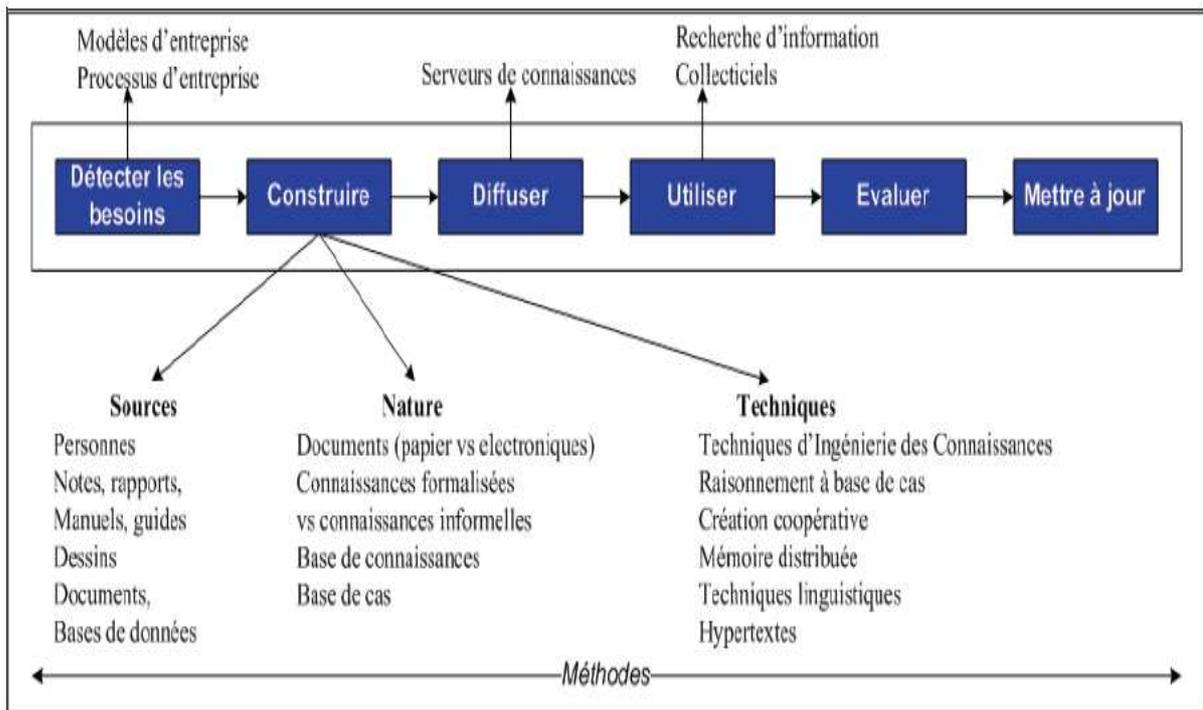


Figure II.3 : Processus de développement de Mémoires Organisationnelles [Dieng et al , 2001].

## II.2.1 Typologie des Mémoires Organisationnelles

L'analyse des travaux existants nous a permis de présenter une typologie des approches de définition de mémoires organisationnelles selon trois volets :

- Mode d'acquisition et de diffusion des connaissances,
- Contenu de la Mémoire Organisationnelle,
- Démarches de réification de la Mémoire Organisationnelle.

**Typologie selon le Mode d’acquisition et de diffusion des connaissances :**

[Heijst et al, 1996] considère que l'existence d'une MO repose sur l'acquisition de la connaissance et sur la diffusion de cette connaissance. Ces auteurs classifient les types de mémoire suivant le type d'acquisition (active ou passive) et le type de diffusion (active ou passive). La combinaison de ces critères définit quatre types de mémoires d'entreprise :

- **Le grenier de connaissance** : utilisé comme une archive de connaissances, accédée et mise à jour au besoin.
- **L’éditeur de connaissance** : il s’agit d’une mémoire organisationnelle gérée par un acteur dont le rôle consiste à diffuser les connaissances aux utilisateurs potentiels.
- **L’éponge à connaissance** : Une mémoire organisationnelle activement mise à jour et consultée en différé.
- **La pompe à connaissance** : une mémoire organisationnelle où la connaissance est capturée et diffusée lors de l’exécution des processus métiers.

	Acquisition active	Acquisition passive
Diffusion active	Pompe à connaissances	Editeur de connaissances
Diffusion passive	Eponge à connaissances	Grenier de connaissances

Tableau II.1 : Typologie de MO selon le Mode d’acquisition et de diffusion des connaissances

**Typologie selon le Contenu des Mémoires Organisationnelles**

La mémoire organisationnelle peut se décliner en quatre sous-mémoires [Pomian, 1996] :

- **Mémoire de profession ou technique ou métier** : se rattache à un métier. Elle s’intéresse à l’aspect opérationnel de l’entreprise, c’est-à-dire à l’expérience acquise liée au travail et permettant à l’entreprise de vivre. Elle est constituée des connaissances liées à un métier et nécessaires à l’exécution des tâches des individus de l’entreprise en vue d’une activité particulière. Ces connaissances rassemblent les connaissances tacites inscrites dans le cerveau des hommes, leurs savoir-faire, leurs compétences, et les connaissances explicites, contenues dans les bases de données, des références, manuels techniques, plans de fabrication, etc.
- **Mémoire managériale** : rassemble les connaissances pertinentes pour les activités de l’organisation à tous les niveaux. Elle peut ainsi inclure des informations sur les structures organisationnelles présentes et passées, sur les ressources humaines, etc. cette mémoire est liée à l’organisation, aux activités, produits, et acteurs comme les clients, sous-traitants, fournisseurs
- **la mémoire individuelle**, constituée par le statut, les compétences, savoir-faire et les activités d’un acteur donné ;

- La mémoire projet est directement liée à une mission particulière réalisée au sein de l'entreprise. Un groupe de personnes est formé pour un temps donné pour réaliser un projet bien défini. La mémoire projet rassemble les connaissances, savoir-faire, compétences, documents qui sont nécessaires à l'accomplissement de ce projet ;

### **Typologie selon la démarche de réification des Mémoire Organisationnelles**

Par démarche de réification, nous entendons les modèles de représentation et de raisonnement sur les connaissances des MOs. Plusieurs modèles ont été appliqués pour la gestion des connaissances des MOs présentant des degrés de formalisation variant du niveau informel vers le niveau formel. Au niveau informel citons par exemple les MO documentaires se basant sur des techniques d'« ingénierie documentaire » et des modèles classiques de recherches d'information (RI)<sup>21</sup>. A un niveau semi- formel citons par exemple les mémoires à base de cas et les mémoires reposant sur les techniques de modélisation de la logique de conception [Matta, 1998].(cf, sectionII.4) Au niveau formel, nous trouvons les MO à base de connaissances SBC et les Webs Sémantiques Organisationnels<sup>22</sup>[Dieng et al, 2001] . Ces derniers auteurs adoptent une approche de modélisation, fondée le concept d'ontologie. Une ontologie est la spécification explicite d'une conceptualisation [Gruber, 1993] . Les recherches scientifiques dans le domaine de l'Ingénierie des Connaissances, présentent l'émergence de travaux sur l'Ingénierie Ontologique [Mizoguchi, 2000], [Gómez-Pérez, 1999], les apports des ontologies pour la représentation des connaissances et leur utilisation pour les systèmes de gestion de MO [Dieng et al. ,2004]. L'Ingénierie Ontologique ainsi que ses applications potentielles pour la capitalisation et le partage de connaissances organisationnelles constituent un fondement théorique principal de nos propositions c'est la raison pour laquelle, nous lui consacrerons le chapitre III de cet état de l'art.

---

<sup>21</sup> Modèles de RI : exemples : modèle booléen, modèles vectoriels, modèles probabilistes. [http://www-lipn.univ-paris13.fr/~rozenknop/Cours/MICR\\_REI/Seance5/introduction-RI.pdf](http://www-lipn.univ-paris13.fr/~rozenknop/Cours/MICR_REI/Seance5/introduction-RI.pdf)

<sup>22</sup> En faisant l'analogie entre les ressources du web et les ressources d'une entreprise, [Dieng et al., 2001] propose de matérialiser la mémoire d'entreprise à travers un «web sémantique d'entreprise (WSE)» (ou « web sémantique d'organisation (WSO)») en utilisant les ontologies qui fournissent un cadre formel pour décrire les différentes sources de connaissances de l'organisation et qui guident la création d'annotations sémantiques facilitant la description, le partage et l'accès à ces sources.

## II.2.2 Typologie retenue pour analyser les travaux d'IS

Compte tenu de la diversité des typologies des MO, de l'absence d'une définition consensuelle des connaissances en IS (produit, processus, expériences..) et de la multidisciplinarité de l'IS ; chaque secteur d'activité d'Ingénierie (Aéronautique, Automobile..) propose une classification spécifique systèmes de MO dépendante de son domaine d'application ; nous proposons la démarche suivante pour analyser les travaux de l'état de l'art :

- Nous nous focalisons tout d'abord sur la catégorie de connaissance gérée par le Système de Gestion de la MO : dans cette première dimension d'analyse nous distinguons : -les **connaissances métiers** correspondants d'une part aux connaissances théoriques « savoir » et d'autre part aux pratiques institutionnalisés<sup>23</sup> « savoir-faire » des experts d'un métier. Ces connaissances constituent un espace de connaissances formalisable. -Les **expériences** résultantes de la mise en œuvre effective des connaissances métiers au cours des projets d'IS. ces connaissances constituent un espace de connaissances capitalisables.
- Nous analysons les classes d'SGC dédiés à chaque catégorie de connaissances, en mettant l'accent sur les facettes de description de ces connaissances : sous l'angle du système étudié, sous l'angle décisionnel, sous l'angle processuel.

Les travaux de références de ce chapitre se focalisent sur la classe de processus non déterministes ; où les objectifs sont partiellement définis et pour lesquels il n'existe pas a priori une solution idoine (« *where the problem is ill structured and there is no one right solution* »). Ces travaux sont essentiellement issus du domaine l'Ingénierie de la Conception (Design Engineering), et du domaine du Génie Logiciel.

---

<sup>23</sup> Invariants d'un métier comme les fonctions ou les constituants d'un système.

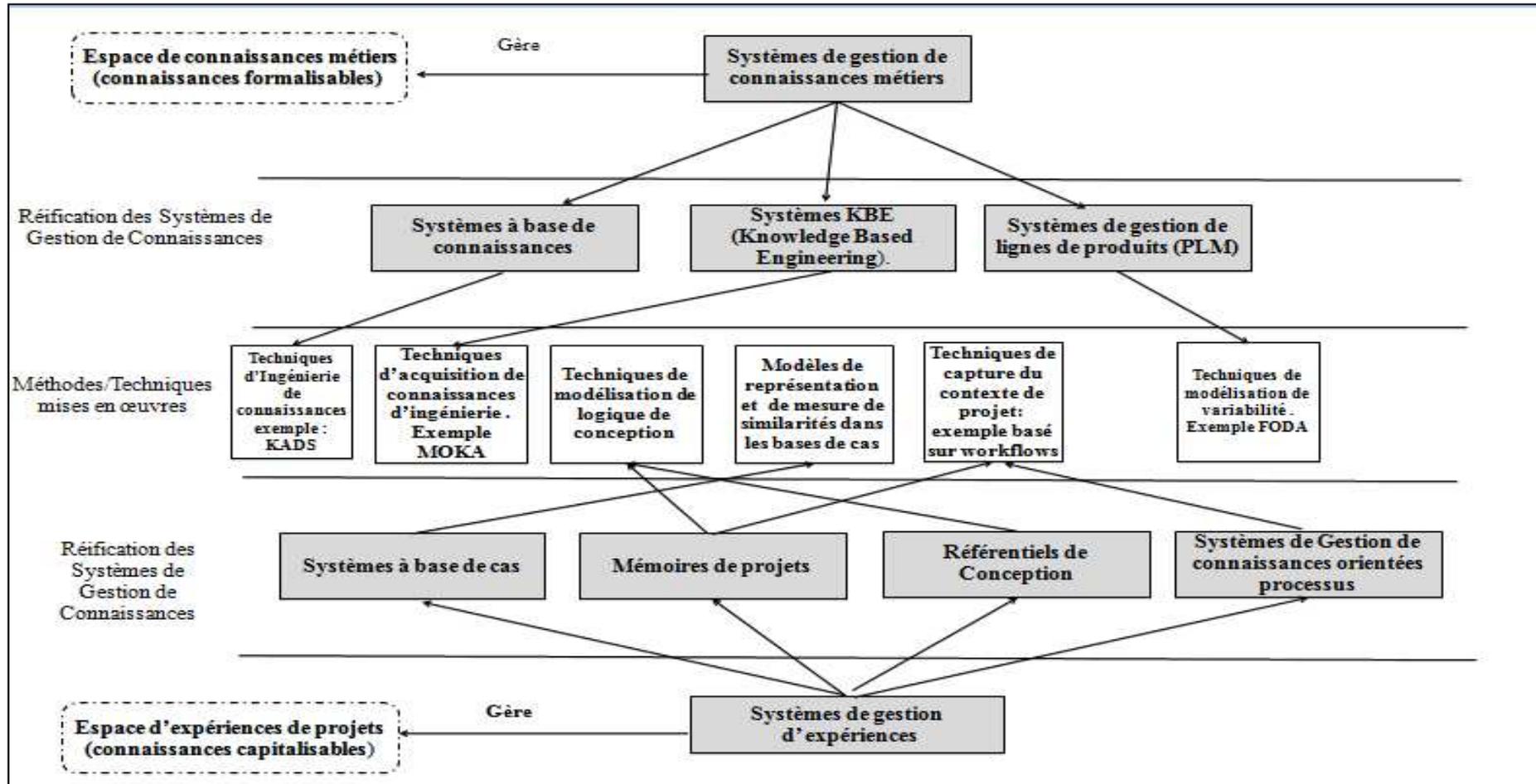


Figure II.3 Typologie retenue pour l'analyse des travaux relevant de l'IS

## II.3 Systèmes de Gestion de Connaissances Métiers<sup>24</sup>

Nous étudions dans cette classe les Systèmes à Base de Connaissances, les Systèmes de « Knowledge Based Engineering » (KBE) et les Lignes de Produit.

### II.3.1 Système à Base de Connaissances(SBC)

Les SBC sont, de manière générale, des systèmes d'information dédiés à la manipulation automatique de connaissances. [Fürst, 2004]. La conception, le développement et l'étude des SBC relèvent d'une branche de l'Intelligence Artificielle : l'Ingénierie des Connaissances (IC). Selon [Bachimont, 2000] l'IC est définie comme « *la modélisation formelle de problèmes pour lesquels les seules connaissances dont on dispose sont de nature linguistique ou cognitive* », ce qui distingue les SBC des systèmes d'information généraux dédiés à la manipulation de données symboliques et/ou numériques. Pour [Charlet et al., 2000]. « *L'Ingénierie des Connaissances est le domaine qui correspond à l'étude des concepts, méthodes et techniques permettant de modéliser et/ou d'acquérir les connaissances pour des systèmes réalisant ou aidant les humains à réaliser des tâches se formalisant a priori peu ou pas* ».

L'IC vise à pallier les insuffisances des Systèmes Experts (SE). Si ces derniers n'avaient pour objet que la résolution automatique de problèmes, l'objectif d'un SBC n'est plus la simulation du processus cognitif de l'expert dans sa tâche de résolution de problème mais de construire un **modèle** qui fournit le même résultat que celui de l'expert [BEN AHMED, 2007]. Ainsi l'IC est perçue comme un processus de modélisation de connaissances d'expert par opposition à un processus d'extraction et de formalisation de ces connaissances à l'aide de règles de production. Dans ce contexte, plusieurs méthodes d'IC ont été proposées [Charlet, 2000]. Ces méthodes sont organisées en trois activités principales : -la première activité est dévolue à la modélisation et à l'acquisition (capture ou élicitation) des connaissances.-la deuxième activité consiste en l'opérationnalisation des modèles de connaissances<sup>25</sup> de la première activité,-la troisième activité est réservée à la validation des modèles, des méthodes et des outils conçus et opérationnalisés. [BEN AHMED, 2007].

Ces méthodes peuvent être classées en trois types d'approches : approche ascendante, approche descendante et approches mixtes. Ces approches de construction de modèle d'expertise ont été comparées dans [Dieng et al., 2000]. Nous retiendrons de cette comparaison que l'approche descendante présente l'avantage majeur de séparer la connaissance du domaine de son utilisation et permet ainsi de définir des composants

---

<sup>24</sup> Dénommés aussi « profession memory » dans les travaux de [Matta, 2002]

<sup>25</sup> Pour modéliser la richesse sémantique des connaissances, de nouveaux formalismes sont introduits, qui représentent les connaissances au niveau conceptuel, y compris la « structure cognitive » d'un domaine. Les langages à base de Frames, les Logiques de Descriptions et les Graphes Conceptuels sont des exemples de tels formalismes [Fürst, 2004]

génériques et réutilisables, tandis que l'approche ascendante présente un modèle structuré d'expertise, mais décrit dans une terminologie propre au problème et manquant d'abstraction. Le tableau II.2 donner un aperçu global de quelques méthodes d'IC

METHODES	SOURCES DE CONNAISSANCES	MODES DE RECUEIL ET DE MODELISATION	TYPLOGIES DE CONNAISSANCES	TYPES DE MEMOIRES
REX	Experts + Documents	Entretiens	Objet descriptif, Point de vue, Terme	Fiches de retour d'expérience
MASK	Experts + Documents	Entretiens + Extraction des documents	Triangle sémiotique: Information, Contexte, Signification.	Activités du domaine
CYGMA	Experts + Documents	Entretiens + Extraction des documents	Connaissances singulières, terminologiques, ontologiques, factuelles, faits initiaux, de buts, d'intégrité, existentielles, synthétiques, stratégiques, structurelles, comportementales, opératoires	Activités du domaine
Atelier FX	Documents	Observation + Extraction semi-automatique des documents	Terme, Donnée	Catalogue raisonné de documents
Componential Framework	Experts	Entretiens	Tâche, Méthode, Information	Activités du domaine
CommonKADS	Experts	Entretiens	Tâche, Inférence, Concept, Relation, Expression	Activités du domaine
KOD	Experts + Documents	Entretiens + Extraction Manuel partir de documents	Taxinomie, actinomie, Schéma,	Activités du domaine
MEREX	Experts	Entretiens	Fiches d'incidents	Incidents

Tableau II.2 Tableau récapitulatif d'un ensemble de méthodes d'IC [Dieng et al, 01].

L'application des méthodes d'IC en IS peut être illustrée par les recherches menées par [Benmahamed el al, 2006]. Les auteurs appliquent la méthode MASK en vue de l'élaboration d'une démarche stratégique pour le transfert des savoir-faire dans le domaine des métiers pétroliers. Des séances de travail collectifs ont été menée dans une division PED (Petroleum Engineering & Development) pour définir l'ensemble des modèles MASK. Ce processus a eu comme résultat la définition d'un Livre de Connaissances de domaine formalisant les savoir-faire métiers critiques du domaine pétrolier. Les travaux de [Monticolo, 2008] sont

plus génériques en ce sens que l'auteur propose un méta-modèle d'acquisition de connaissances pour les processus de conception de produit intitulé : RIO basé sur les concepts de Rôle, d'Interaction et d'Organisation. Ce modèle générique est utilisé pour capturer la cartographie des connaissances métier utilisées lors des projets de conception.

### II.3.2 Systèmes de « Knowledge Based Engineering » (KBE)

Dans le domaine de la conception de produits, la proportion de tâches répétitives augmente par rapport à celle des tâches à forte valeur ajoutée. Les systèmes KBE (*Knowledge Based Engineering*) ou systèmes d'ingénierie à base de connaissances ont suscités beaucoup d'intérêt dans les organisations centrées sur les activités de conception.

A la différence des outils traditionnels de CAO<sup>26</sup>, une application KBE permet l'acquisition des intentions se situant en amont de la conception en représentant les « pourquoi » et « comment » d'un modèle en plus de la description du « quoi ». [Susca *et al.*, 2000].

Ce type d'applications s'appuie sur des connaissances métier pour permettre l'automatisation ou la semi-automatisation des tâches de conception dites routinières. L'objectif étant de réduire les délais de conception et de favoriser l'innovation en libérant du temps de créativité aux concepteurs. Un inconvénient majeur de cette classe de système réside dans la difficulté de réutilisation des connaissances dans d'autres applications [Hew *et al.* 2001].

La méthodologie MOKA<sup>27</sup> (Methodology and tools Oriented to Knowledge Based Engineering Applications) a été proposée pour permettre la réutilisation des connaissances métiers en se basant sur une modélisation des produits et des processus de conception.

Le modèle produit décrit le « quoi » de la conception. Le modèle inclut des vues, classes et des attributs prédéfinis. Il y a cinq vues dans le modèle produit:

- La vue « structure » inclut les classes : structure, produit, assemblage, pièce, et la classe *feature*. C'est la vue principale du méta modèle à partir de laquelle tous les autres liens sont faits.
- La vue « fonction » inclut les classes : Fonction, principe de solution, solution technique. Cette vue est en liaison avec la vue « structure » et la vue « comportement »
- La vue « comportement » inclut les classes : Comportement, état et transitions. Cette vue est attachée à la structure
- La vue « technologie » décrit les technologies, le processus de fabrication et le matériel utilisé.
- La vue « représentation » décrit la géométrie de la structure.

---

<sup>26</sup> Systèmes de conception assistée par ordinateur : exemple CATIA

<sup>27</sup> <http://web1.eng.coventry.ac.uk/moka/lifecycle.htm>

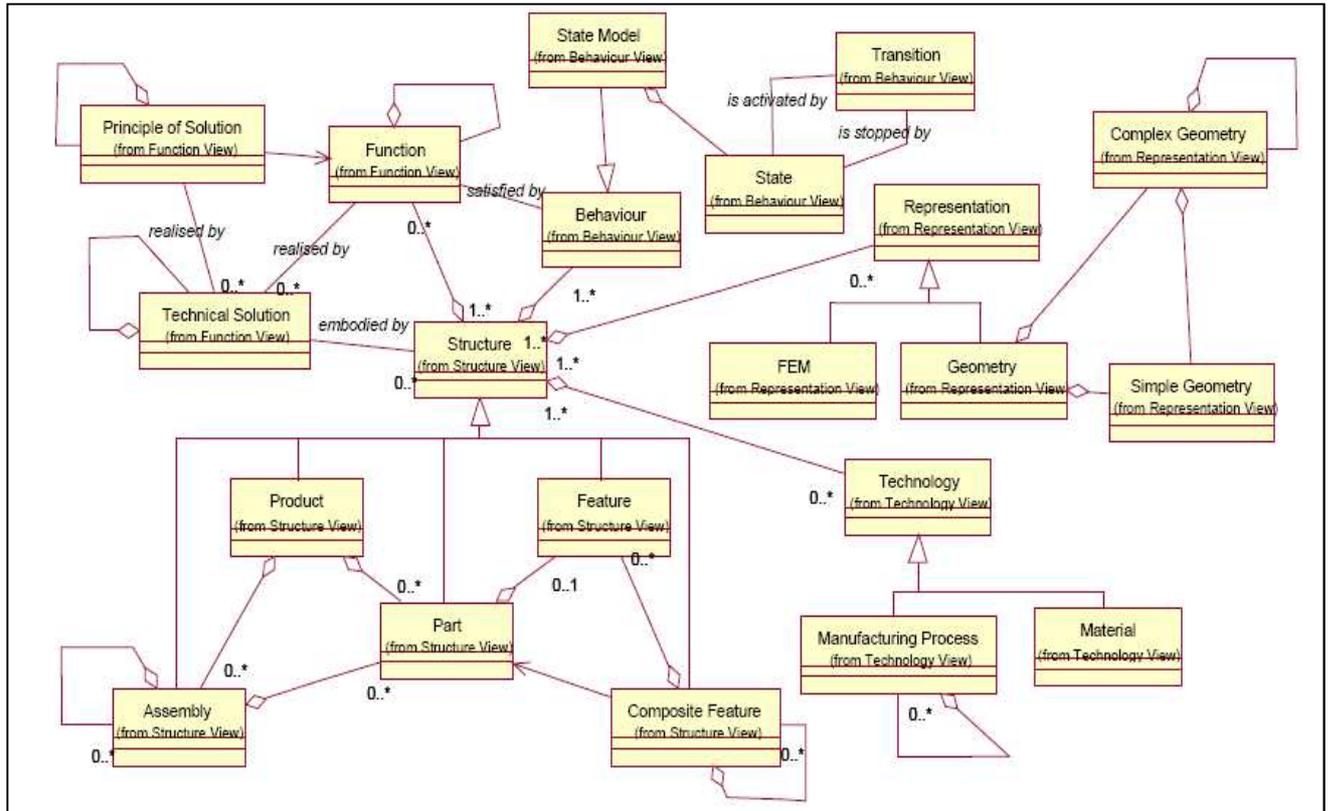


Figure II.4 Le méta modèle produit MOKA<sup>28</sup>

La méthodologie MOKA contribue en particulier, aux phases de capture (collecte et structuration des connaissances brutes) et de formalisation (développement d'un modèle produit et processus). Les objectifs sont la définition d'une méthodologie pour structurer et formaliser les connaissances, et le développement d'un outil capable de supporter la méthodologie et d'illustrer le couplage avec une plate-forme KBE [Oldham et al, 1998]. Cette méthode fournit les moyens d'acquérir et de représenter les connaissances liées au produit et à son processus de conception. La phase d'acquisition de connaissances consiste à recueillir des fiches « ICARE » définissant les éléments de connaissances suivants [Klein et al, 2000] :

- Illustration pour identifier des cas déjà vécus ou des cas particuliers,
- Contrainte pour les éventuelles limitations imposées sur le produit,
- Activité pour décrire le processus de conception,
- Règle pour traduire les contraintes dans le processus de conception,
- Entité pour décrire le produit selon les points de vue structurel, fonctionnels ou comportemental.

<sup>28</sup> <http://web1.eng.coventry.ac.uk/moka/lifecycle.htm>

Cette méthodologie a été largement appliquée en industrie. Plusieurs exemples d'applications de la méthodologie Moka peuvent être trouvés dans [Wiels, 1997]. Dans [Doche et al, 1998] et [Diagne et al, 1998], les auteurs présentent l'application de la méthode Moka à un système avionique. Plusieurs groupes industriels mettent en œuvre cette méthodologie pour modéliser et capitaliser la conception de leurs produits [Wiels, 1997].

<b>Entity form:</b>	water pump assembly	
<b>is-a:</b>	assembly	
<b>description:</b>	The water pump system as a whole	
<b>source:</b>	<<link to text in Document>>	
<b>subclasses:</b>	directly-driven-water-pump indirectly-driven-water-pump one-stream-pump two-stream-pump	
<b>attributes:</b>	pump-flow rotational-speed weight shape	cm <sup>3</sup> /s 1/s kg geometric-shape
<b>constraints:</b>		
<b>part-relations:</b>	has-driving-syst. has-transmission has-housing has-pump-syst.	driving-syst. transmission housing pump-syst.
<b>relations:</b>	Cooled-motor location	Motor geometric-space

<b>Constraint form:</b>	has-driving-system
<b>is-a:</b>	Structural constraint Part relation
<b>description:</b>	The part relation from the water pump to its driving system
<b>source:</b>	<<link: text in Document>>
<b>domain:</b>	water-pump
<b>range:</b>	driving-system
<b>conditions:</b>	[1,1]; [1,1]
<b>cardinality</b>	
<b>constraints</b>	water-pump.rot-speed = driving-system.rot-speed

Figure II.5 : Exemples de fiches ICARE [Klein et al, 2000]

### II.3.3 les Lignes de Produits (LdP)

L'approche « Lignes de Produits » est une transposition des chaînes de production au monde du système et du logiciel [Ziadi, 2003]. Le principe est de minimiser les coûts de construction de logiciels dans un domaine d'application particulier en ne développant plus chaque logiciel séparément, mais plutôt en le concevant à partir d'éléments réutilisables, appelés « assets ». Un élément réutilisable peut ainsi être une exigence, un modèle, un composant, un plan de test ou tout simplement un document.

Selon [Salinesi, 2009], les lignes de produit sont devenues le moyen principal de capitaliser et réutiliser les artefacts en IS. Les domaines d'application sont nombreux, allant de systèmes complexes tels que des véhicules automobiles, aux systèmes d'information de type progiciels de gestion intégrée. Le principe essentiel est de développer puis d'utiliser des modèles conceptuels de la ligne de produit, dans lesquels sont capitalisés tous les éléments que l'on veut réutiliser au niveau des produits, ainsi que les contraintes d'agencement de ces éléments.

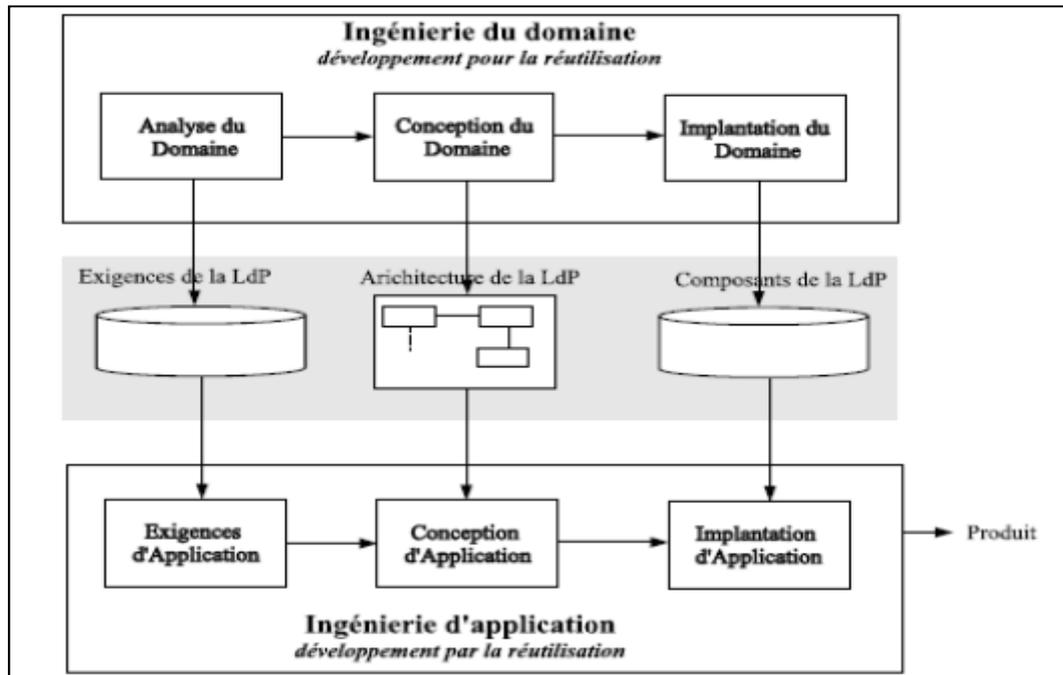


Figure II. 6 Ingénierie des lignes de produits [Ziadi, 2003]

La première difficulté liée à cette approche réside dans la conception d'une architecture permettant de définir plusieurs produits. Les membres d'une ligne de produits sont caractérisés par leurs points communs, mais aussi par leurs différences (aussi appelées « points de variation »). La gestion de cette variabilité est un des concepts clé des lignes de produits. Par exemple dans le domaine automobile, des voitures sont fabriquées à partir d'un ensemble d'éléments communs (roues, volant, vitres, etc...) mais peuvent comporter certaines caractéristiques qui les différencient (nombre de chevaux du moteur, présence ou non de la climatisation, ...).

La deuxième difficulté réside dans l'utilisation d'une ligne de produits. L'Ingénierie d'un système (ou « dérivation de produit ») consiste en partie à figer certains choix vis -à-vis des points de variation définis dans la ligne de produits. Certains choix sont incompatibles entre eux. Si nous reprenons l'exemple du domaine automobile, une voiture comporte généralement un seul moteur, et il faut alors choisir entre une motorisation essence ou diesel. De la même manière, un choix particulier lors de la dérivation d'un logiciel peut exclure certaines variantes. Par exemple le choix d'un cabriolet à toit amovible exclura la possibilité de choisir un toit ouvrant. Une ligne de produits doit donc aussi intégrer des contraintes de cohérence permettant de faciliter les choix lors de la dérivation.[Ziadi, 2003].

Deux classes de techniques sont proposées pour gérer les Lignes de Produits : -les techniques d'ingénierie de domaine (développement pour la réutilisation) dédiées principalement à

l'analyse des besoins (comme FODA<sup>29</sup>), la conception (comme UML) et l'implémentation (comme les langages à objets) et -les techniques d'ingénierie d'application (développement pour la réutilisation) pouvant être classées selon [Jezequel, 2008] en deux sous catégories : dérivation par configuration(automatisant la dérivation du produit grâce à un *configureur* ayant comme entrée les besoins applicatifs et se basant sur les contraintes du domaine modélisées dans la LdP) et dérivation transformation (s'appuyant l'IDM<sup>30</sup> (*model-driven engineering* (MDE) pour dériver des produits à partir d'une modélisation de la ligne de produit.

Les travaux de recherches portants sur les Lignes de Produits sont issus principalement du domaine du Logiciel [Jezequel, 2008], [Ziadi, 2003]. Comme exemple d'application industrielles de ce concept nous citons : les travaux de [Fannmy,2009] ,de la société ADN, utilisant les LdP dans l'ingénierie automobile et les travaux de [Beaugard,2009] de la société THALES proposant l'utilisation de deux LdPs pour capitaliser les connaissances des missions de satellites dans le contexte télécommunication.

### II.3.4 Synthèse

#### Avantages

- Les approches de gestion de connaissances métiers s'adaptent essentiellement aux processus d'ingénierie routiniers
- La réutilisation des connaissances métiers favorise l'innovation en libérant du temps de créativité aux ingénieurs systèmes

#### Inconvénient

- Nécessitent un effort considérable d'abstraction de modèles
- Nécessitent plusieurs sessions d'acquisition de connaissances avec les experts du métier.
- Les connaissances modélisées sont faiblement contextualisée

L'analyse de ces approches nous a permis essentiellement de dégager les facettes de description des connaissances métiers en nous référant essentiellement aux points de vues proposés par la méthode MOKA (point de vue structurel, fonctionnel, comportemental). Ces approches couvrent une première composante de notre problématique celle de la formalisation des connaissances métiers en IS, la deuxième composante liée à la capitalisation des expériences en IS est étudiée dans la section II.4.

---

<sup>29</sup> FODA : Feature-Oriented Domain Analysis.

<sup>30</sup> IDM : a [software development methodology](#) which focuses on creating models, or abstractions, more close to some particular domain concepts rather than computing (or algorithmic) concepts. It is meant to increase productivity by maximizing compatibility between systems, simplifying the process of design, and promoting communication between individuals and teams working on the system.

## II.4 Systèmes de Gestion d'Expériences

Les démarches de Gestion d'Expériences s'inscrivent dans le cadre général des Systèmes de Gestion de Connaissances, et constituent une alternative aux méthodes d'Ingénierie de Connaissances (section II.4.1) relativement complexes à mettre en œuvre. En effet, la gestion des expériences permet de rendre relativement transparente la phase d'acquisition de connaissances « *acquisition bottleneck* » car elle est effectuée dans un contexte<sup>31</sup> d'ingénierie.

Réutiliser les expériences passées pour améliorer la résolution des problèmes courants est une démarche extrêmement commune dans les pratiques industrielles. D'après [Rakoto, 2004] il est « **plus facile pour les acteurs de formaliser leur expertise à partir d'expériences vécues plutôt que de chercher à expliciter des connaissances génériques non contextualisées.** ». Une expérience est définie comme la fusion entre des connaissances techniques et formelles intégrées à la mise en pratique de ces connaissances lors de situations effectives de résolution de problèmes [Rakoto, 2004].

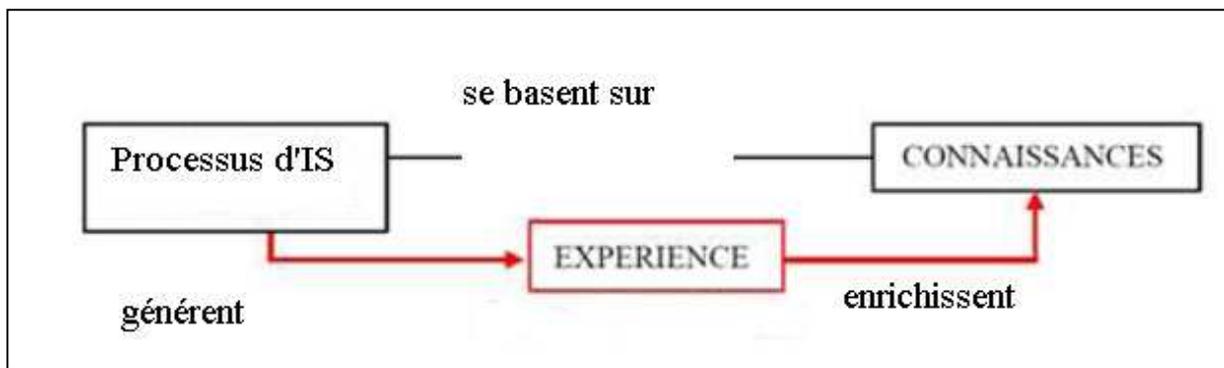


Figure II.7 : L'expérience comme vecteur de production de connaissances

Les concepts fondateurs des systèmes dédiés à la gestion des expériences ont été proposés dans les travaux de [Bergman, 2002a] qui définit la gestion d'expérience comme : « *Experience management is a special kind of knowledge management, focussing on the dissemination of specific knowledge situated in a particular problem-solving context* ».

<sup>31</sup> Par contexte nous entendons la description d'une situation d'ingénierie en fonction des compétences des acteurs, des activités d'ingénierie, des contraintes et des exigences du système étudié.

[Bergman, 2002] formalise la gestion d'expérience par un uplet ( P, (C,L), F) où :

- P : est une description d'un espace de problèmes selon des dimensions dépendantes de la nature de la tâche à résoudre (conception, diagnostic, validation etc.).
- (C,L) : désigne une expérience élémentaire. ce couple est constitué des dimensions de description de l'expérience C et de la description du contenu de cette expérience L
- F : C = F(P): fonction de transformation de l'espace de description du problème P vers les dimensions de description de l'expérience C.

En se basant sur ce modèle général, la réutilisation des expériences se décline en quatre étapes : - la description d'un problème en cours,- la réécriture de cette description de l'espace P vers l'espace C,- le choix de l'expérience la plus appropriée au problème en cours,- l'application et l'adaptions de l'expérience. Le choix de la solution appropriée est basé sur le modèle de représentation et de raisonnement des expériences. [Bergman, 2000b] introduit le concept de *mesure d'utilité*.

Nous étudions dans ce qui suit quelques démarches de réification des systèmes de gestion d'expériences en les illustrant par des travaux relatifs aux métiers de l'IS. La première classe de système (II.4.1) présente un paradigme de raisonnement applicable aux classes (II.4.2, II.4.3 et II.4.4).

### II.4.1 Les Systèmes à Base de Cas

Le raisonnement à partir de cas (RÀPC) est un paradigme de raisonnement qui permet de résoudre de nouveaux problèmes en adaptant les solutions de problèmes passés déjà résolus. Il a pour objectif de résoudre un nouveau problème, appelé problème cible, à l'aide d'un ensemble de problèmes déjà résolus, appelés problèmes sources. [Aamodt, 1994].

Un système à base de cas est un système particulier de gestion des connaissances qui répond à un cycle de raisonnement, le cycle du RÀPC. Le cycle du RÀPC se décompose en cinq étapes. Chaque étape joue un rôle particulier dans la résolution du problème et mobilise des connaissances spécifiques [Cordier et al., 2006].

-Elaboration : L'élaboration d'un nouveau problème (cas cible) représente l'acquisition des informations connues sur le nouveau problème. Le cas doit être représenté d'une manière similaire à un cas source.

-Remémoration : La remémoration des cas (sources) les plus similaires signifie la recherche des correspondances entre descripteurs des cas de la base et du cas à résoudre. Différentes techniques peuvent être utilisées : calcul d'un degré d'appariement des descripteurs (similarité entre deux cas), pondération éventuelle des descripteurs d'un cas.

-Adaptation : L'adaptation des cas afin de résoudre un nouveau problème se fait en réutilisant la solution d'un cas similaire.

-Révision : La révision de la solution proposée signifie l'évaluation de la solution dans le monde réel (bilan d'un cas), la vérification par introspection de la base de cas en considérant la qualité des cas.

-Apprentissage : l'apprentissage des nouvelles connaissances travers une mémorisation d'un nouveau cas cible dans la base de cas

Ce cycle est illustré par la figure II.8

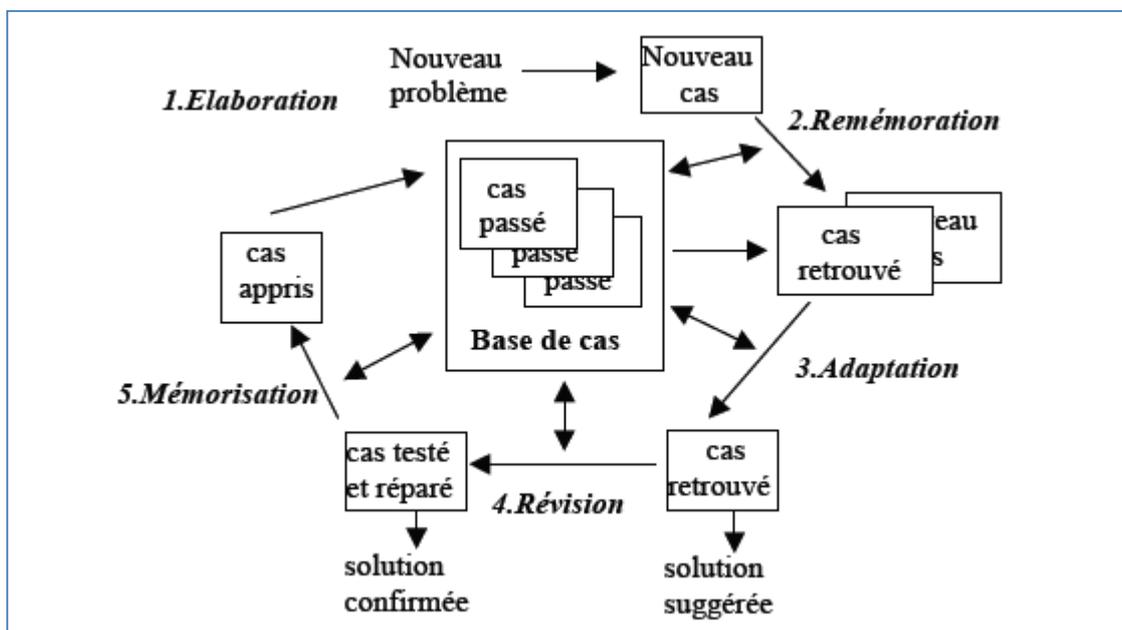


Figure II.8 Le cycle de raisonnement à partir de cas [RENAUD et al., 2007]

Dans le cadre de nos travaux, nous nous intéressons plus particulièrement aux phases d'élaboration et de remémoration. L'élaboration de la description du nouveau problème dépend essentiellement du modèle de représentation et de raisonnement de connaissances adopté. Dans [Bergmann et al., 2003] trois différents types de systèmes de RàPC par rapport à la représentation des cas sont distingués : - *les approches textuelles* : les cas sont, soit non structurés sous forme de texte libre, soit semi structurés sous forme de texte découpé en plusieurs parties telles que « problème », « solution », - *les approches structurelles* : les cas sont représentés sous forme de paires <attribut, valeur>. Un attribut représente une caractéristique importante du domaine applicatif. La représentation des cas peut être sur un seul niveau (plate) ou sur plusieurs niveaux (hiérarchiques), - *les approches conversationnelles* : le modèle conversationnel est interactif et emploie à côté de la description du problème et de la solution une série de questions et de réponses pour compléter la description du problème

Dans [Assali et al., 2009] une autre classe d'approches de RAPC est considérée : Les *approches knowledge-intensive du RAPC (KI-CBR)* pour lesquelles les connaissances du domaine jouent un rôle fondamental. Dans les systèmes de KI-CBR, les ontologies jouent un rôle important : comme référentiel conceptuel pour décrire les cas, comme structure de connaissances ou les cas sont localisées, et comme source de connaissance permettant le raisonnement sémantique dans les méthodes de calcul de similarité.

La remémoration dépend essentiellement de la représentation de cas, cette phase se base sur le calcul de similarité entre les cas de la base de cas. Ces calculs de similarités peuvent être : - locales où les mesures sont établies au niveau des caractéristiques de cas. Les mesures les plus communes sont basées sur la notion de la distance.

Elles dépendent du type de descripteur (numérique, symbolique, taxonomique), -globales où les mesures sont établies au niveau de cas ou d'objet par agrégation de similarités locales. Actuellement ce sont les mesures de similarités conceptuelles qui sont utilisées dans les phases de remémoration. Ces mesures sont synthétisées dans [Zargayouna et al., 2004].

Le RAPC est particulièrement adapté aux situations dans lesquelles la théorie du domaine est faible ou peu formalisable. Les Systèmes à Base de Cas, ont été alors principalement appliqués au processus de conception de produits. Les projets KRITIK [Goel et al, 1989], et CADET [Sycara et al, 1991] sont des travaux représentatifs de cette classe de systèmes. CADET capitalise dans une base de cas les descriptions conceptuelles des spécifications d'un système exprimées sous la forme de caractéristiques fonctionnelles, comportementales et de contraintes physiques. KRITIK, décrit un cas par les caractéristiques fonctionnelles d'un système pointant sur une solution décrivant sa structure physique.

Les travaux de recherches de [Champin, 2002] s'inscrivent aussi dans le cadre de processus de conception de produit et ont proposés la formalisation des traces de processus décisionnels (intitulées les épisodes de conceptions) dans base de cas intégrée au système de CAO CATIA.

Dans le domaine du Génie logiciel le cadre générique de capitalisation de connaissances « Experience factory » [ Basili, 1995], se base sur le raisonnement à base de cas. La capitalisation de connaissances concerne la totalité les artefacts produits dans un processus de développement de logiciel sous forme de « paquetages » : comme les paquetages de produits (exemples : diagrammes de conception, documents de tests), paquetages de processus (exemples : modèles de processus, fragment de méthodes..), des paquetages d'outils (comme les outils de génération de code).

L'analyse des travaux sur le RAPC, nous a permit de conclure que ce concept s'adapte bien au processus d'IS mais présente une difficulté majeure quant à la gestion des évolutions et la réutilisation des connaissances. Ceci est principalement du à une conception *ad hoc* et spécifique au domaine d'application des base de cas.

## II.4.2 Les Référentiels de conception (« *Design repositories* »)

Cette classe de systèmes a été proposée par [Skyman et al, 2001], comme support aux processus de développement de produits collaboratifs et décentralisés. Dans ce contexte, les auteurs expliquent que les bases de données techniques ne sont plus suffisantes pour répondre aux besoins des usagers. Les acteurs des processus de développement de produit ne partagent plus simplement des données géométriques de produits mais des spécifications, des fonctions, et des contraintes de produits.

Pour répondre à ces nouveaux besoins, les auteurs le concept de “*design repository*” comme la nouvelle génération de systèmes de développement de produit et le définissent comme « *a knowledge-based approach for supporting engineering design. Design repositories capture not only what is designed, but also how and why the product is designed. Design repositories enable engineers to capture the evolutionary nature of product knowledge and information throughout the design process* ». [Skyman et al, 2001][Regli, 2000]

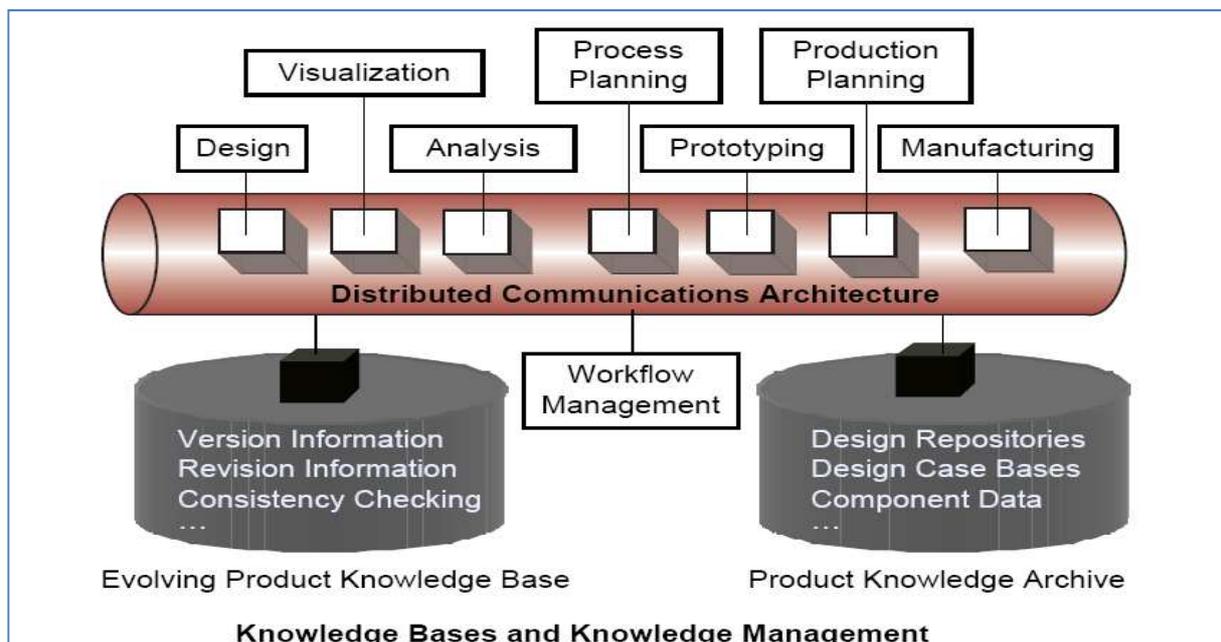


Figure II.9 les référentiels de conception selon [Skyman, 2001]

L’objectif principal étant de capturer plus « vision complète de ce qui est produit en cours de conception ». Cette vision n’est pas encore appliquée, il s’agit d’une proposition d’un concept de ce qui peut constituer une nouvelle génération de support de processus d’ingénierie

Deux catégories de connaissances capitalisables dans les référentiels de conception ont été identifiées :

- Les connaissances liées aux produit ( *Product Knowledge Representation*) : Cette catégorie est constituée de spécifications, d'exigences, de taxonomies fonctionnelles et de flux , de taxonomies structurelles et de contraintes de produits.
- Les connaissances liées à la logique de conception (*Design Rationale*<sup>32</sup>) : cette catégorie désigne les raisons de choix de conception d'un produit et permettent de capitaliser (tracer) le raisonnement conduisant à la définition d'un produit.

Le projet de recherche "NIST Design Repository"<sup>33</sup> a eu comme objectifs l'étude d'architectures techniques implémentant les propositions théoriques des référentiels de conception. Des propositions d'implémentations se sont basées sur les technologies du web sémantiques<sup>34</sup>. Par exemple les travaux de [Kopena et al., 2003]<sup>35</sup> se sont focalisés sur la capitalisation et la réutilisation des connaissances fonctionnelles (*Functional Modeling of Engineering Designs for the Semantic Web.*)

### II.4.3 Les Mémoires de projets

Un projet est une intention de réaliser une idée donnée, d'atteindre un certain objectif, de construire une certaine ébauche [Matta et al, 2001], [Dieng et al, 2001]. La réalisation d'un projet suit généralement plusieurs étapes ascendantes jusqu'à l'aboutissement à l'objectif visé. Tout au long de ces étapes, l'idée à la base du projet, se précise au fur et à mesure jusqu'à sa concrétisation. Dans ce type de situation, une confrontation de propositions provenant de disciplines diverses est un moteur important de production de connaissances. Les mémoires de projets sont de systèmes de gestion des connaissances spécifiques, qui se focalisent essentiellement sur les connaissances émergentes dans une situation de projet.

D'après [Matta, 2004], la mémoire projet est une matérialisation possible de la mémoire organisationnelle : *La mémoire de projet est une mémoire des connaissances et des informations acquises et produites au cours de la réalisation des projets. Elle peut être obtenue à travers une capitalisation au fil de l'eau de l'activité de l'entreprise, notamment de la logique de conception.*

La mémorisation « au fil de l'eau » consiste à tracer<sup>36</sup> et ensuite à structurer les connaissances produites dans une activité en cours pour garantir une réutilisation de ces connaissances.

---

<sup>32</sup> Présentation plus détaillée dans la section II.4.3

<sup>33</sup> Conduit à « Institute of Standards and Technology

<sup>34</sup> W3C. <http://www.w3.org/>

<sup>35</sup> <http://sites.computer.org/debull/A03dec/kopena.ps>

<sup>36</sup> La traçabilité permet de garder une trace de la mémoire épisodique dans la quelle des associations spatio-temporelles des événements sont décrites

Deux classes de connaissances sont gérées dans une mémoire de projets [Bekhti, 2003] à savoir :

- Le contexte de projet : constitué de
  - o le déroulement et l'organisation d'un projet : le processus et l'ordonnancement des activités, les acteurs impliqués avec leur rôle dans le projet et leurs compétences ;
  - o l'environnement de travail : méthodes, techniques et outils utilisés, objectifs, exigences et contraintes, références, procédures qualités, normes, directives et réglementations.
- La logique de conception décrit essentiellement :
  - o les problèmes rencontrés : description et classification
  - o le mode de résolution des problèmes : propositions de solutions, argumentations, décisions.

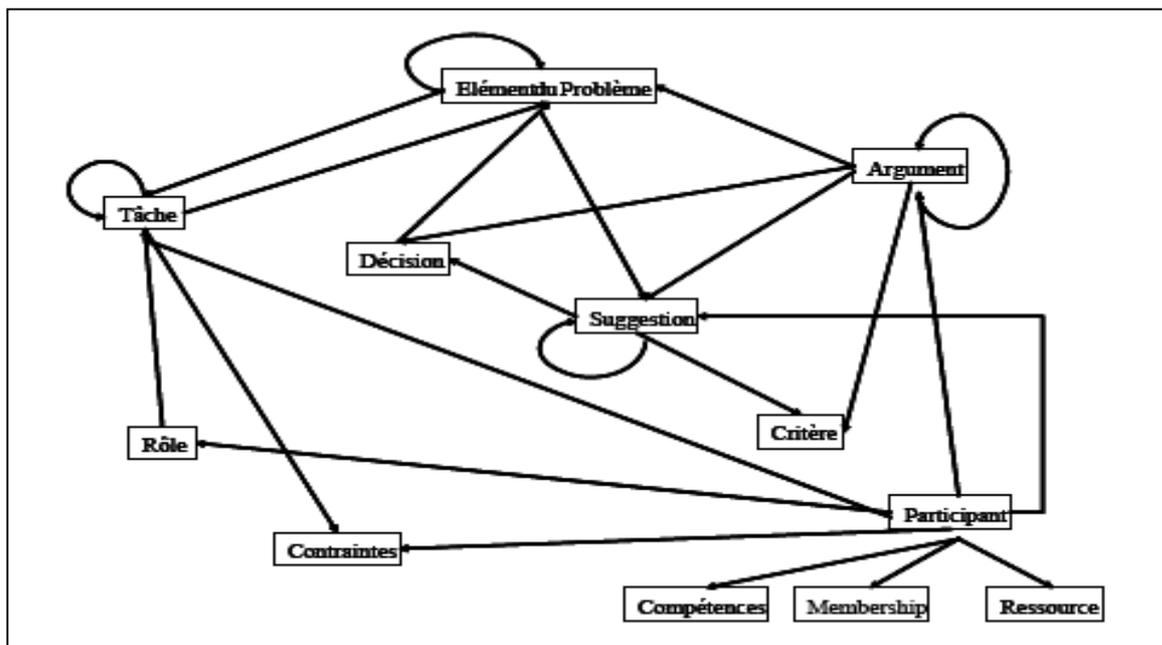


Figure II.10 Modèle conceptuel d'une mémoire de projet [Bekhti, 2003]

Le tableau II.3 donne un aperçu sur les principaux modèles de logique de conception proposés dans la littérature. [Matta, 2004].

Comme travaux représentatifs de cette classe de système de gestion de connaissances, nous citons les travaux de [Matta, 1998] appliqués à la gestion des conflits entre acteurs au sein des processus de conception concurrente et les travaux de [Longueville et al, 2003] appliqués à la conception de produits innovants dans le domaine automobile.

Approche	Logique de conception	Résultats du projet	Gestion du projet	Outils	Types d'applications
IBIS	Arbre: Issue/ Position/ Argument			gIBIS QuestMap	Conception
QOC	Arbre: Question/ Option/ criteria				Conception
DRAMA	Arbre: Goal/ Option + Table de critères			DRAMA	Conception + Tous types d'applications
DIPA	Graphe de résolution de problèmes: Étapes, opinions, arguments, rôles et décisions			Memo-Net	Tous types d'applications
DRCS	Graphe: Entité/ Relation	Graphe: Entité/ Relation	Graphe: ordonnancement de tâches Graphe: Entité/Relation	DRCS system	Ingénierie concurrente
EMMA			Arbre : Goal/Plan	EMMA	Génie de logiciels
SAGACE		Graphe de Flux		Systémographe	Tous types d'applications

Tableau II.3 Exemple de modèles de logique de conception

## II.4.4 les Mémoire orientée processus

Dans une mémoire orienté processus, le système de gestion de connaissance est développé autour d'un modèle de processus. Cette stratégie considère les processus comme un moyen important de relier les connaissances à leur contexte de création et d'utilisation. Le principe directeur des systèmes de gestion de connaissances orientés processus est résumé par [Abecker et al , 2000] dans ces termes *“The process-centred approach follows the philosophy that people do not simply need more information, but information pertinent to the tasks they are carrying out at the time.”*

Les connaissances capitalisées dans cette classe de système sont décrites selon trois points de vue : -le point de vue organisationnel décrivant les activités et les acteurs de l'organisation, -le point de vue informationnel décrivant les types de ressources gérés par l'organisation et- le point de vue domaine décrivant les concepts du métier de l'organisation.

Plusieurs projets de recherche illustrent la mise en œuvre cette classe de systèmes. Citons par exemple les systèmes Milos [Holz, 2003], EULE[Reimer , 2000] et KnowMore [Abecker et al, 2001].

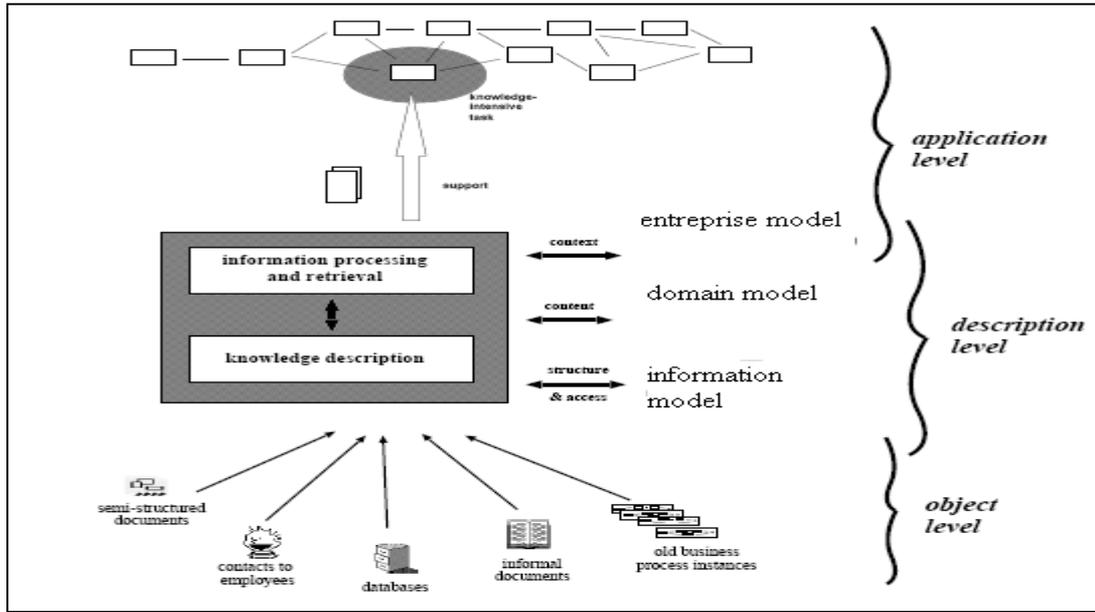


Figure II.11 Concept de mémoire orientée processus [Abecker, 2000]

## II.5 Synthèse

Nous avons cherché à mettre en évidence à travers cette analyse de l'état de l'art les principales facettes de description des connaissances gérées dans les Systèmes de Gestion de Connaissances. Les deux classes de systèmes de gestion de connaissances métiers et de gestion d'expériences sont au cœur de notre problématique car nous visons la gestion de ces deux classes de connaissances. Les travaux analysés émanent de courants de recherches différents comme l'Ingénierie de Connaissances et l'Ingénierie des Systèmes d'information. Chaque courant de recherche propose une étude de la connaissance selon son point de vue spécifique. Nous remarquons que propositions se chevauchent et qu'il est impératif d'aboutir à un cadre unificateur des ces différentes propositions. Dans le tableau II.4 une comparaison des approches étudiées selon la dimension facettes de description de connaissances est fournie. (En unifiant les différentes terminologies utilisées dans la littérature).

Ayant dégagé les facettes principales de description des connaissances, nous nous focalisons sur une deuxième composante de notre problématique : la modélisation des connaissances. Les différents Systèmes de Gestion de Connaissances étudiées dans ce chapitre possèdent toutes une partie dédiée à la modélisation des connaissances (exemple méta modèle de produit dans MOKA). Il s'agit d'un point convergence des ces différentes approches même si ce n'est pas toujours mis en évidence.

Ces modèles sont souvent formalisées sous une forme relativement élémentaire et spécifique au domaine d'application (une simple catégorisation terminologique est effectuée dans la méthodologie d'IC REX, il n'y a pas de relations entre concepts pour la méthodologie d'IC MKSM), ce qui limite leurs potentiels de réutilisabilité.

Les difficultés des Systèmes de Gestion de Connaissances sont dues aux facteurs suivants [BEN AHMED, 2007] :

- Les hypothèses de base d'un SGC sont formulées implicitement ce qui limite le partage et la réutilisation des connaissances;
- Il n'existe pas de base de connaissances de un haut niveau à partir de laquelle il est possible de générer d'autres bases de connaissances;
- Il n'existe pas de processus d'accumulation de connaissances

L'Ingénierie Ontologique a été précisément conçue en vue de pallier ces difficultés en leur apportant une élégante solution car les connaissances considérées par l'I. O. sont cumulables, partageables et révisables. Ce qui est fondamental en Ingénierie Ontologique (IO) c'est que la structure ontologique constituée d'hyper-concepts (concepts fondamentaux) constitue le soubassement cognitif (un solide fondement) à partir duquel des modèles de SGC peuvent être générés ; modèles dont les caractéristiques cardinales sont : partage, accumulation réutilisation et interopérabilité de connaissances. [BEN AHMED, 2007]. Le rôle d'une ontologie envers une base de connaissances est de lui fournir les définitions des concepts utilisés dans la représentation des connaissances ; un rôle est aussi de spécifier les contraintes entre les concepts grâce auxquelles la base vérifie deux propriétés importantes : sa consistance et la transparence ; deux propriétés nécessaires pour rendre les connaissances partageables et réutilisables. Nous nous intéressons à l'Ingénierie Ontologique et à ses apports au domaine d' l'IS dans le troisième chapitre de cette thèse.

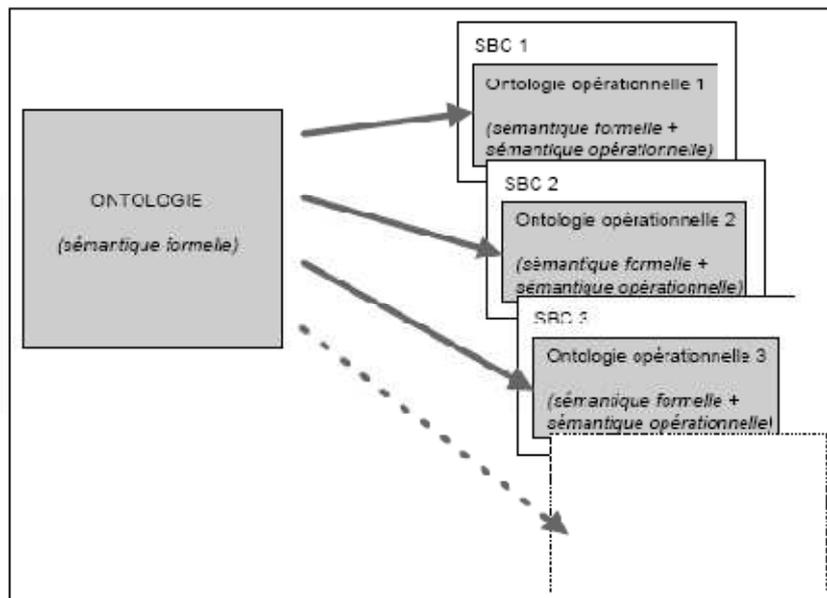


Figure II.12 : l'Ingénierie Ontologique comme soubassement cognitif des Systèmes de Gestion de Connaissances [FÜRST,2004]

Classe de Système de Gestion de Connaissances		Exigences / contraintes	Fonctions	Structure	Artefacts	Processus	Ressources	Argumentaire (logique de conception)
<b>Systèmes de gestion de connaissances Métiers</b>	SBC	Formalisables dans des modèles de domaines (exemple méthodologie <i>CommonKADS</i> )			Non couvert	Modèle de tâche	Modèle d'agent	Non couvert
	KBE	Capturé dans des fiches de contraintes	Vue fonctionnelle du méta modèle produit MOKA	Vue structurelle du méta modèle produit MOKA(part, assembly..)	Non couvert	Non couvert	Non couvert	Non couvert
	LdP	Analyse du domaine	Conception du domaine	Implantation du domaine	Non couvert	Non couvert	Non couvert	Non couvert
<b>Systèmes de gestion d'expériences</b>	Systèmes à base de cas	Peut inclure toutes ces dimensions en fonction du cas d'application						
	Référentiels de conception	Connaissances de produits			Non couvert			Connaissances liées à la logique de conception
	Mémoires de Projets	Non couvert			Contexte de projet			logique de conception
	Mémoires orientées processus	Modèle de domaine			Modèle informationnel	Modèle organisationnel		Non couvert

Tableau II.4 Eude comparative des facettes de description des connaissances en IS

# Chapitre III:

## De l'Ingénierie Ontologique en IS

---

### Plan du chapitre

---

III.1 Introduction

III.2 Concepts de base de l'Ingénierie Ontologique (IO)

III.2.1 Définitions des Ontologies

III.2.2 Typologies des Ontologies

III.2.3 Processus de développement, méthodes et outils de l'IO

III.2.4 Rôles de l'Ingénierie Ontologique

III.3 De l'Ingénierie Ontologique en Ingénierie Système

III.3.1 l'Ingénierie Ontologique comme base de systématisation de connaissances en IS

III.3.2 l'Ingénierie Ontologique comme base de partage de connaissances en IS

III.4 Conclusion

## III.1 Introduction

La notion d'ontologie est tout d'abord apparue dans le domaine de la philosophie. Dans ce contexte une « Ontologie » est une théorie de l'existence. C'est en ces termes qu'elle est définie : « Partie de la métaphysique qui s'applique à l'être en tant qu'être, indépendamment de ses déterminations particulières. » (Le Petit Robert). En IA et avec l'émergence de l'Ingénierie des Connaissances (IC), les Ontologies sont apparues comme des réponses aux problématiques de représentation et de manipulation des connaissances au sein des systèmes informatiques, c'est cet aspect que nous nous proposons de traiter dans ce chapitre. Ce chapitre s'articulera autour de deux parties : nous analyserons dans une première partie les fondements de l'Ingénierie Ontologique puis nous nous focaliserons sur les applications potentielles de l'IO en IS.

## III.2 Concepts de base de l'Ingénierie Ontologique (IO)

### III.2.1 Définitions des ontologies

Il existe depuis l'émergence de la mode ontologique au sein de la communauté informatique au sens large du terme, plusieurs définitions de l'ontologie. Un consensus semble se dessiner ces dernières années à partir de multiples acceptions suggérées dans la littérature. Nous présentons dans ce qui suit les principales définitions.

Du point de vue de l'Intelligence Artificielle, une ontologie est définie comme « **une spécification explicite d'un conceptualisation** » [GRUBER, 1993]. Cette définition pionnière due à Tom GRUBER est considérée par une large partie de la communauté comme étant référentielle. Une conceptualisation est une vision du monde-cible qui se fonde sur un ensemble de concepts universaux et d'autres particuliers au monde-cible

Du point de vue des Systèmes à Base de Connaissances (SBC), une ontologie est définie par R. MIZOGUCHI comme « Une ontologie est un système de concepts fondamentaux, c'est-à-dire un système de connaissances qui constitue les connaissances d'arrière-plan d'une base de connaissances; une ontologie offre une conceptualisation du monde-cible ainsi qu'une base solide sur laquelle construire des bases de connaissances partageables, et plus largement utilisables qu'une base de connaissances conventionnelle »[MIZOGUCHI, 1995].

Une autre définition avancée par T. GRUBER deux années après la première selon laquelle « **une ontologie est une conceptualisation partagée** » [GRUBER, 1995]. Une conceptualisation partagée inclut les cadres conceptuels pour la modélisation de la connaissance du domaine; les protocoles spécifiques à communication entre agents d'interopération ainsi que le consensus sur la représentation des théories d'un domaine particulier. Dans le contexte de partage des connaissances, les ontologies sont spécifiées sous la forme de définitions d'un vocabulaire représentationnel. Le cas le plus simple est un vocabulaire de type hiérarchique spécifiant des classes et leurs relations de subsomption. Les

schémas de bases de données relationnelles peuvent servir comme ontologies en spécifiant les bases de données partagées et l'intégrité des contraintes qui doivent être vérifiées par elles.

N. GUARINO [GUARINO, 1995], affine la définition de T. GRUBER en considérant les ontologies comme des « spécifications partielles et formelles d'une conceptualisation »

Pour [GUARINO, 1995] et [GRUBER, 1995], la **conceptualisation** est une vue abstraite et simplifiée du monde qu'on cherche à représenter, la **spécification formelle** signifie que les ontologies sont basées sur des théories formelles qui permettent à une machine de vérifier automatiquement certaines propriétés de consistance et/ou de faire certains raisonnements automatiques sur les ontologies et leurs instances. **Explicite** signifie que les catégories de concepts du domaine et leurs différentes propriétés sont décrites explicitement. **Partagée** signifie qu'une ontologie capture une connaissance consensuelle, i.e., admise par tous les experts d'une communauté.

Le schéma de [Guizzardi, 2005] (Figure III.1) clarifie la définition de Gruber, en explicitant les relations existantes entre une conceptualisation, un langage, une abstraction du domaine et une spécification dans le langage qui représente cette abstraction du domaine.

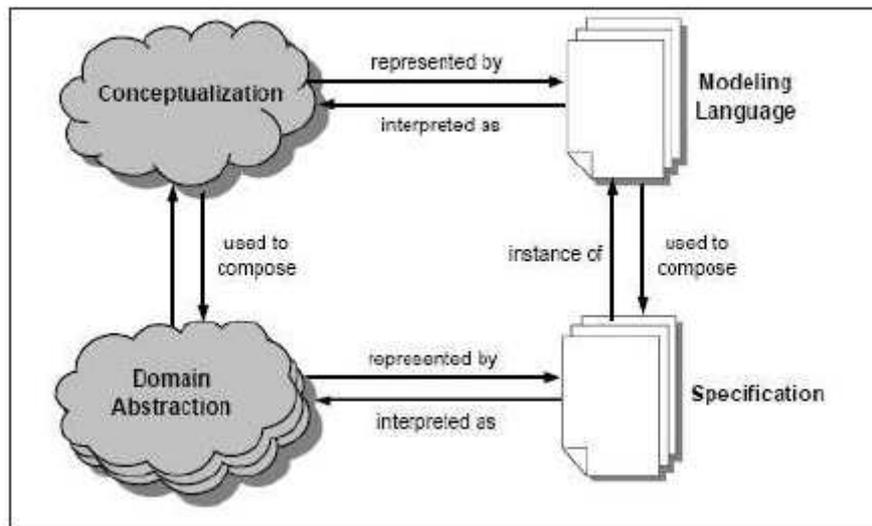


Figure III.1 Les relations entre conceptualisation, abstraction du domaine, langage de modélisation et spécification [Guizzardi, 2005]

Nous retiendrons dans le cadre de ce travail, que les ontologies donnent le cadre formel dans lequel les connaissances sont exprimées. Elles définissent des choix et des contraintes qu'il faut respecter pour que des affirmations aient une signification dans un domaine ou une application.

## III.2.2 Typologies des Ontologies

D'après une synthèse de [Mizuguchi, 2004 a] [Gomez-Perez, 1999] [Lassila et al, 2001] les ontologies sont classifiées suivant plusieurs dimensions, nous en retenons quatre :

- Typologie selon l'objet de conceptualisation
- Typologie selon le niveau de complétude
- Typologie selon le niveau de granularité
- Typologie selon le niveau d'expressivité

### III.2.2 .1 Typologie selon l'objet de conceptualisation :

Il s'agit de la classification la plus courante des ontologies, sept catégories d'ontologies sont distinguées :

**Les ontologies de représentation des connaissances** : les ontologies de représentation des connaissances sont utilisées pour formaliser un modèle de représentation des connaissances. On peut par exemple citer l'exemple de l'ontologie de *frame* , qui définit les primitives de représentation des langages à base de frames (classes, instances, slots, facettes, etc.).

**Les ontologies supérieures (aussi appelées ontologies de haut niveau)** : ces ontologies modélisent le travail réalisé par les philosophes depuis longtemps dans leur travail d'explication de ce que l'on trouvait dans le monde. En particulier, les ontologies de haut niveau modélisent les concepts les plus généraux que l'on puisse définir. Leur objet est l'étude des catégories des choses qui existent dans le monde. Comme les concepts de haute abstraction tels que les entités, les évènements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés, etc. On peut citer ici par exemple les dix catégories d'Aristote (matière, quantité, qualité, relation, position, temps, etc.) ou encore les notions de primalité (*firstness* en anglais – ce qui peut être défini sans condition : humain, forêt, etc.), secondarité (*secondness*, ce qui est défini dans un certain contexte : professeur, mère, etc.) et tertialité (*thirdness*, ce qui donne le contexte) de C.S. Pierce, intégrées dans l'ontologie supérieure de Sowa [SOWA, 1995]. On peut aussi noter l'existence d'efforts pour créer des ontologies standard de haut niveau<sup>37</sup>

**Les ontologies génériques** : appelées aussi, **méta-ontologies** ou **core ontologies**, modélisant des connaissances génériques moins abstraites que celles véhiculées par l'ontologie de haut niveau, mais assez générales néanmoins pour être réutilisées à travers différents domaines [Psyché, 2003].

---

<sup>37</sup> Standard Upper Ontology Working Group, <http://suo.ieee.org>

**Les ontologies des tâches** : ce type d'ontologie sert à modéliser les tâches d'un problème ou d'une activité donnée. Ce type d'ontologie est utile pour décrire la structure d'une tâche de résolution de problème de manière indépendante du domaine concerné.

**Les ontologies de domaine** sont réutilisables à l'intérieur d'un domaine donné et modélisent le vocabulaire à l'intérieur de ce domaine. La plupart des ontologies existantes sont des ontologies de domaine [Psyché, 2003]

**Les ontologies de tâches-domaine** : ce sont des ontologies de tâches spécifiques à un certain domaine. Un exemple d'une telle ontologie est celui d'une ontologie des termes liés à la planification chirurgicale [Gomez-Perez, 1999].

**Les ontologies d'application.** Il s'agit du type d'ontologie le plus spécifique. Les concepts que l'on trouve dans ce genre d'ontologies modélisent les concepts d'un domaine particulier dans le cadre d'une application donnée.

### III.2.2.2 Typologie selon le niveau de complétude

[Bachimont, 2000] définit trois niveaux de complétude :

**Niveau 1 : Sémantique** : tous les concepts, caractérisés par un terme/libellé, doivent respecter les quatre principes différentiels :

- Communauté avec l'ancêtre ;
- Différence, spécification, par rapport à l'ancêtre ;
- Communauté avec les concepts frères, situés au même niveau ;
- Différence par rapport aux concepts frères.

Ces principes correspondent à l'*engagement sémantique* qui assure que chaque concept aura un sens univoque et non contextuel associé. Deux concepts sémantiques sont identiques si l'interprétation du terme/libellé à travers les quatre principes différentiels aboutit à un sens équivalent.

**Niveau 2 : Référentiel** : les concepts référentiels ou formels, se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. L'*engagement ontologique* spécifie les objets du domaine qui peuvent être associés au concept, conformément à sa signification formelle. Deux concepts formels seront identiques s'ils possèdent la même extension. Ce qui permet de définir une ontologie référentielle, constituée de prédicats formels pourvus d'une sémantique référentielle ou extensionnelle.

**Niveau 3 : Opérationnel** : les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des inférences u *engagement computationnel*. Deux concepts opérationnels sont identiques s'ils possèdent le même potentiel d'inférence.

### III.2.2.3 Typologie selon le niveau de granularité

On peut distinguer les ontologies selon le niveau de description utilisé [Psyché, 2003] :

- **Granularité fine** : ce niveau correspond à des ontologies très détaillées, elles possèdent ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche. Ce niveau de granularité peut s'avérer utile lorsqu' il s'agit d'établir un consensus entre les agents qui l'utiliseront ;
- **Granularité large** : ce niveau correspond à des vocabulaires moins détaillés. Par exemple, les Scénarios d'utilisation spécifiques où les utilisateurs sont déjà préalablement d'accord à propos d'une conceptualisation sous-jacente, Les ontologies de haut niveau possèdent une granularité large, compte tenu que les concepts qu'elles traduisent sont normalement raffinés subséquentment dans d'autres ontologies de domaine ou d'application.

### III .2.3. 4 Typologie selon le niveau d'expressivité

Selon cette dimension [MIZOGUCHI, 2003] distingue les ontologies « *lightweight* » (légères) et ontologies « *heavyweight* » (« lourdes »). Les premières contiennent typiquement une simple hiérarchie de concepts ainsi que des relations entre ces concepts. Les secondes sont définies de manière plus précise, en déterminant des propriétés avancées sur ces concepts permettant des inférences.

[Lassila, 2001] présente une échelle des degrés d'expressivité des ontologies. Cette échelle va du vocabulaire contrôlé, simple ensemble de mots appartenant à un domaine, aux ontologies lourdes comprenant les termes désignant les concepts et relations du domaine, leurs propriétés conceptuelles, et toutes les connaissances nécessaires à la description de la sémantique du domaine. Entre les deux, des glossaires aux ontologies légères, les représentations de connaissances offrent des possibilités croissantes d'intégrer la sémantique du domaine, en spécifiant toutefois toujours la terminologie.

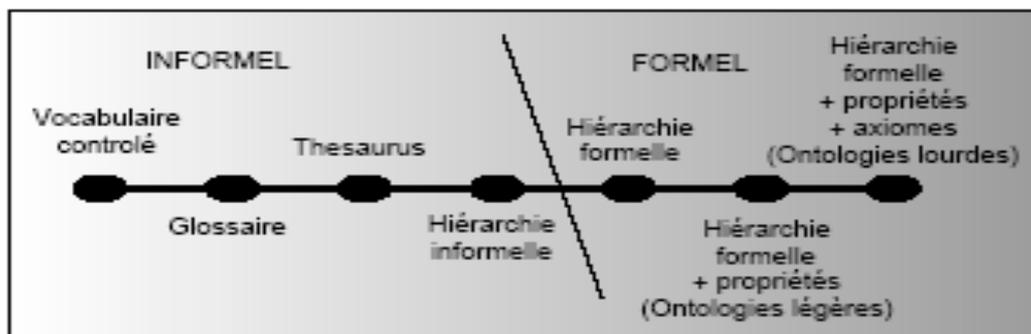


Figure III.2 typologie des ontologies selon le niveau d'expressivité [Lassila, 2001]

[GRUNINGER et al, 19995] expliquent que le degré de formalisation exigé du langage pour l'ontologie dépend étroitement du degré d'automatisation dans les diverses tâches impliquant l'ontologie. Si une ontologie est une aide à la communication entre personnes, alors la représentation de l'ontologie peut être informelle du moment qu'elle est précise et qu'elle capture les intuitions de chacun. Cependant, si l'ontologie doit être employée par des outils logiciels ou des agents intelligents, alors la sémantique de l'ontologie doit être rendue beaucoup plus précise.

La figure III.3 résume les dimensions de classification d'ontologies que nous avons analysées ci-dessus.

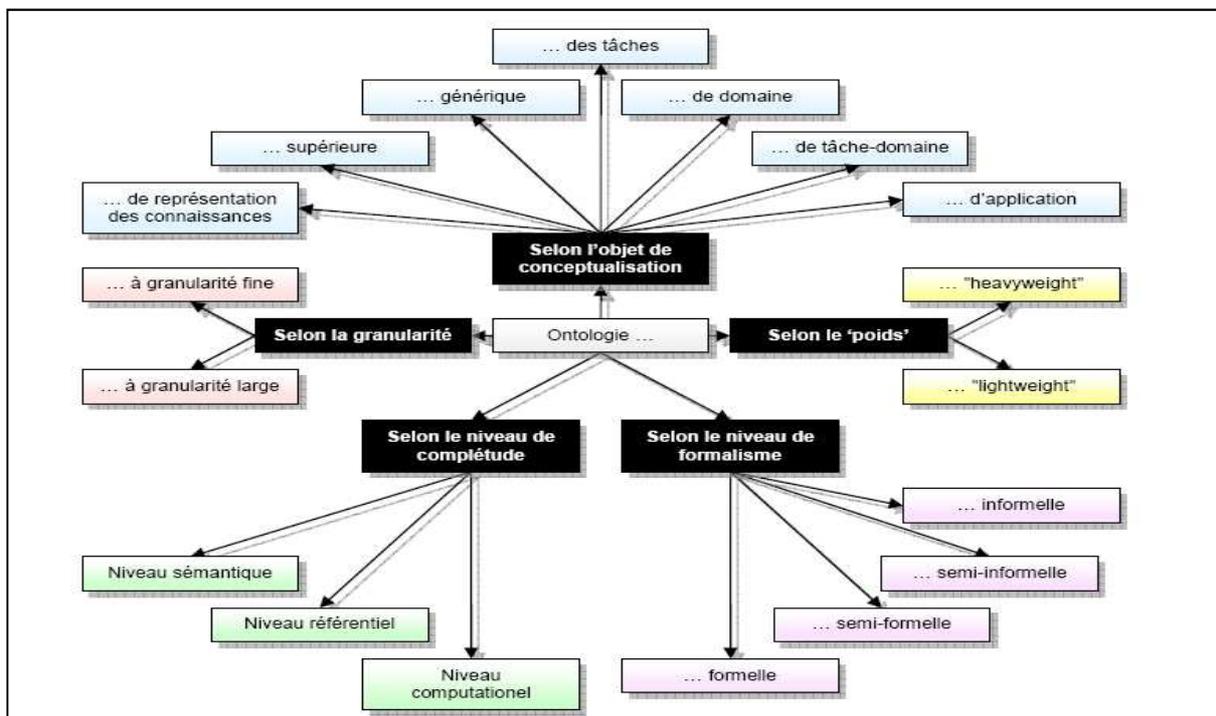


Figure III.3 Synthèse des différentes classifications des ontologies [AUBRY, 2007]

## III.2.3 Processus de développement, méthodes et outils de l'IO

### III.2.3.1 Processus de développement des Ontologies

Le processus de développement général d'une ontologie se déroule en trois principales étapes qui sont : la conceptualisation, la formalisation appelé également ontologisation et l'opérationnalisation. Ces étapes sont généralement précédées d'une étape de d'évaluation de besoins et de délimitation du domaine de connaissances à modéliser.[FURST, 2004]

- la **conceptualisation**, conduit, à partir d'un ensemble de ressources (corpus, normes, experts..) à l'élaboration d'un **modèle conceptuel**, informel ou semi-formel, identifiant les connaissances du domaine au travers des concepts manipulés et de leur sémantique ;
- l'**ontologisation**, conduit du modèle conceptuel à l'**ontologie** proprement dite, représentation formelle des connaissances du domaine aussi indépendante que possible des objectifs applicatifs ;
- l'**opérationnalisation**, conduit de l'ontologie à une **ontologie opérationnelle**, qui est une représentation formelle des connaissances du domaine adaptée à une application particulière. le terme opérationnel à la fois pour caractériser un modèle de représentation doté de mécanismes de raisonnement et d'une sémantique opérationnelle, et pour caractériser tout langage exécutable qui implémente un tel modèle. Plusieurs modèles de représentation de connaissances sont utilisées en IO [Dameron, 2003], parmi lesquels nous citons les modèles à base de Frames [KIFER, 1995], les modèles basés sur les Logiques de Description [BAADER, 1991] et le modèle des Graphes Conceptuels [SOWA, 1984]. Chacun de ces modèles de représentation est implémenté dans un ou plusieurs langages implémentant une partie ou la totalité du modèle, en particulier des langages du Web Sémantique<sup>38</sup> ( comme RDF/RDFS, OWL , SWRL).

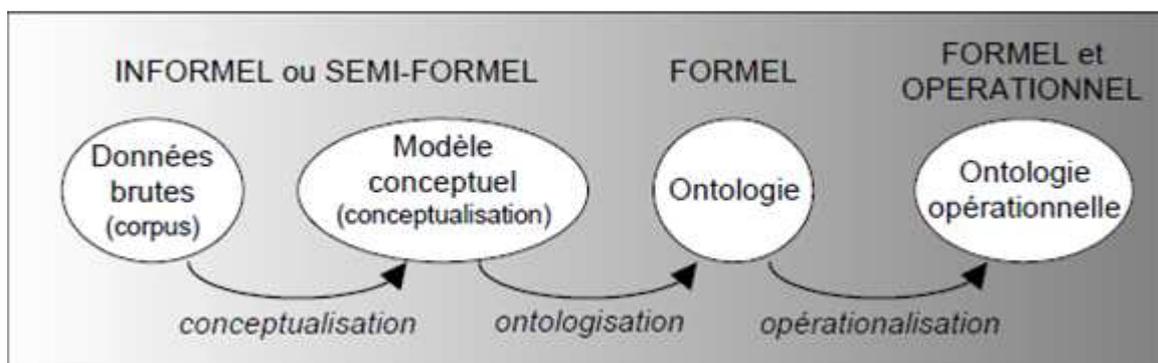


Figure III.4 Processus de développement des ontologies [Furst, 2004]

<sup>38</sup> [www.w3.org](http://www.w3.org)

### III.2.3.2 Méthodes d'IO

La méthode choisie pour construire une ontologie, doit être fortement guidée par le type d'ontologies désiré et les objectifs de son utilisation. À l'heure actuelle, il n'existe pas encore de consensus à propos des meilleures pratiques à adopter lors du processus de développement des ontologies.

Plusieurs méthodes d'IO ont été proposées dans la littérature [Gomez-Perez et al, 1997] [Gruninger et al. 1995], [Mizoguchi R., 1998][ Uschold, 1995]. [Guarino, 1995] qualifie ces méthodologies de "*résultat de mixture d'une créativité ad-hoc et d'introspection naïve*".

Ces méthodes peuvent être analysées selon plusieurs critères ou dimensions, l'un d'entre eux étant le type du processus de construction [Mendes O., 2003][Isaac, 2005]:

- Construction d'Ontologie à partir de zéro : basée majoritairement sur l'extraction d'ontologies à partir de textes. La méthode TERMINAE [Després et al, 2006] est un exemple de cette approche. Elle se base sur les étapes de *Constitution d'un corpus* (documents techniques, comptes rendus, livres de cours, etc.), à partir d'une analyse des besoins de l'application visée, l'*Etude linguistique*, pour identifier des termes et des relations lexicales, en utilisant des outils de traitement de la langue naturelle comme LEXER [Assadi et al., 00], la *Normalisation sémantique*, conduisant à des concepts et des relations sémantiques définis dans un langage semi-formel. La *Formalisation* et intégration des concepts au sein d'une Base de Connaissance formelle.
- Construction d'Ontologie par réutilisation : basée sur l'intégration ou fusion avec d'autres ontologies, Ceci nécessite très souvent une étape d'alignement, qui identifie les concepts et les relations que ces ontologies ont en commun. Des exemples de propositions méthodologiques ou techniques concernant cette approche sont trouvables dans ONIONS [Gangemi et al, 1999] ou PROMPT [Noy , 2003]
- Construction collaborative : ces travaux cherchent davantage à mettre en valeur et assister la nécessaire collaboration entre les concepteurs des ontologies, en mettant à leur disposition des dispositifs de discussion et de gestion de versions différentes des ressources en cours de construction. [Domingue , 1998]

Dans le cadre de nos travaux nous nous positionnons par rapport à une méthode par réutilisation d'ontologies existantes. A cette fin nous nous sommes inspirés de la méthodologie générale proposée par [Uschold, 1995]. Cette méthode est basée sur l'expérience du développement de l'ontologie Enterprise Ontology<sup>39</sup>, et repose sur ces différentes étapes :

- Identification du « Pourquoi » de l'ontologie (sous la forme de question de compétences)
- Construction de l'ontologie :
  - Identification des concepts clefs
  - Modélisation informelle
  - Formalisation et intégration d'ontologies existantes ;
- Evaluation et documentation de l'ontologie

### III.2.3.3 Environnements et outils d'IO

Un ensemble d'environnements d'IO ont été développés afin de systématiser l'IO. Ces environnements offrent des services de construction, d'extraction, d'alignement, d'annotation, de raisonnement et d'évaluation d'ontologies [GÓMEZ-PÉREZ et al., 2003]. Nous présentons un échantillon d'environnements d'IO, ayant fait l'objet d'une étude comparative dans [Furst, 2004].

	Oiled	OntoEdit	OntoLingua	Protégé	WebODE	WebOnto	DOE	TooCoM
Modèle de Connaissances	DL (DAML + OIL)	Frames + FOL	Frames + FOL	Frames + FOL	Frames + FOL	Frames + FOL	Entité/Relation	Entité/Relation syntaxe GC
Langage d'axiomes	Oui (DAML + OIL)	Oui (FLogic)	Oui (KIF)	Oui (PAL)	Oui (WAB)	Oui (OCML)	Non	Oui syntaxe GC
Support méthodologique	Non	Oui	Non	Non	Oui	Non	Oui	Non
Moteur d'inférence	FaCT (interne)	OntoBroker (interne)	ATP (externe)	PAL (interne)	Prolog (interne)	Oui (interne)	Non	Cogitant (externe)
Test de cohérence	Oui	Oui	Non	Oui	Oui	Oui	Non	Oui
Hierarchies graphiques	Non	Non	Oui	Oui	Oui	Oui	Oui	Oui

**Modèle de Connaissances** : paradigme et langage utilisés dans l'outil pour représenter les ontologies

**Langage d'axiomes** : langage utilisé dans l'outil pour représenter les axiomes des ontologies

**Support méthodologique** : guide méthodologique pour la construction et la structuration des ontologies

**Moteur d'inférence** : il peut être spécifique à l'outil (interne) ou développé à part (externe)

**Test de cohérence** : fonctionnalité permettant d'évaluer la cohérence des ontologies

**Hierarchies graphiques** : interface graphique de restitution et de manipulation des hierarchies de concepts et de relations

Tableau III.1 étude comparative des outils d'IO [Furst, 2004]

<sup>39</sup> <http://www.aiai.ed.ac.uk/project/enterprise>

### III.2.4 Applications de l'IO

Plusieurs classes d'applications de l'IO ont été proposées dans les travaux de [Ushold et al, 1999] et [Mizoguchi, 2003]. Nous présentons dans cette section une synthèse de ces applications

#### Type 1: l'Ingénierie Ontologique comme vocabulaire commun

C'est ce que l'on peut qualifier d'ontologie de "surface" car elles ne traitent ni la structure conceptuelle profonde, ni les connaissances fondamentales du monde cible. L'ontologie dans cette classe d'application, est un vocabulaire compréhensible par un ordinateur (commun entre des utilisateurs et des machines). Une ontologie est créée pour permettre à plusieurs applications d'accéder, par le partage ou l'échange, à des données communes. Des traducteurs bidirectionnels sont développés pour faire le lien entre les structures de données propres aux applications et le format commun de l'ontologie. Un premier apport est de réduire le coût des applications multiples en leur donnant un accès à des données communes et de faciliter l'interopérabilité. Un second apport, pour l'utilisateur final, est d'avoir accès, dans un format unique, à des sources d'informations hétérogènes.

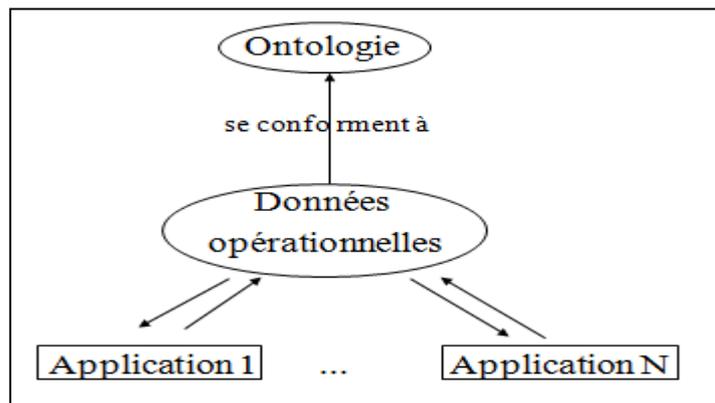


Figure III.5 : l'Ingénierie Ontologique comme vocabulaire commun [Ushold et al, 1999]

#### Type 2: L'Ingénierie Ontologique comme base de partage de connaissances

L'ontologie joue le rôle d'un index pour les ressources recherchées, ce qui renforce l'espoir de retrouver des informations pertinentes. Les connaissances ontologiques permettent de représenter le sens de la requête et d'effectuer des inférences sur les informations décrivant le contenu des ressources (les métadonnées ou annotations sémantiques [Euzenat, 2005]), permettant d'améliorer la qualité de la recherche. En annotant les ressources du Web ou d'une organisation sur la base d'un vocabulaire commun s'adossant à une ontologie, il est possible de rendre l'accès au Web plus performant et mieux adaptés aux besoins des usagers. Le KM dans sa nouvelle génération intègre, à la fois, métadonnées et ontologies.

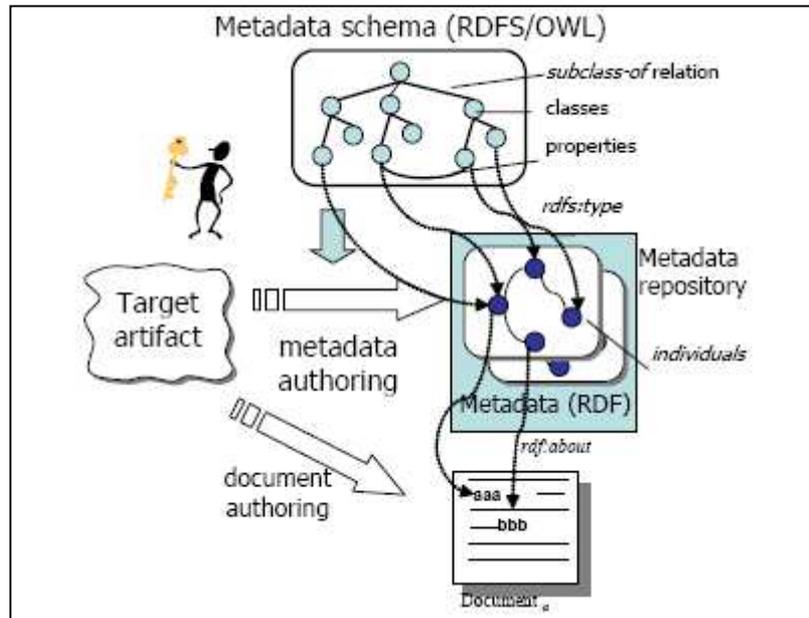


Figure III.6 : l'Ingénierie Ontologique comme base de partage de connaissances [Kimatura, 2006]

### Type 3: L'Ingénierie Ontologique comme spécification

L'ontologie modélise un domaine et fournit un vocabulaire pour spécifier les besoins d'une (ou plusieurs) application(s) cible(s). L'ontologie guide le développement de systèmes opérationnels. Suivant les cas, ces derniers peuvent contenir (ou non) une nouvelle représentation explicite de l'ontologie. Les motivations pour cette approche sont diverses :

- Promouvoir la réutilisation de connaissances dans plusieurs applications,
- Faciliter la maintenance de logiciels grâce à une représentation explicite de l'ontologie sur laquelle ils sont basés,
- Rendre pérennes des connaissances ontologiques, dans une perspective de mémoire organisationnelle.

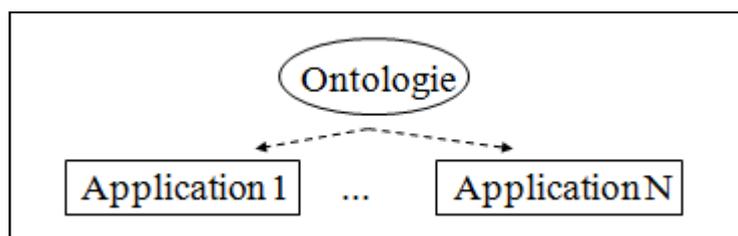


Figure III.7 : l'Ingénierie Ontologique en tant que spécification [Ushold et al, 1999]

### **Type 5: L'Ingénierie Ontologique comme base de la systématisation des connaissances**

L'Ontologie peut constituer le noyau d'une structuration conceptuelle. Cette fonctionnalité est essentiellement significative pour les méta-ontologies (Ontologies de Haut Niveau). La systématisation des connaissances exige :

- La formalisation des concepts basiques;Le positionnement de ces concepts dans la structure conceptuelle tout en exhibant les relations qu'ils peuvent entretenir avec d'autres concepts;
- Une identification plus poussée des relations nécessaires entre concepts;
- Une profonde compréhension des concepts tout en considérant leurs situations et en les compilant en vue d'être prêts à l'emploi au second niveau.

## **III.3 De l'Ingénierie Ontologique en Ingénierie Système**

Les applications de l'Ingénierie Ontologique (IO) dans les domaines d'ingénierie ont été principalement étudiées dans les travaux de [Mizoguchi, 2004 a] [Mizoguchi, 2004 a] et [Kimatura, 2006]

Nous présenterons ici des travaux représentatifs, la liste des investigations qui se font dans ce domaine n'est pas exhaustive vu que ce domaine de recherche est pluridisciplinaire et qu'à ce jour certains travaux sont en cours. Nous nous focaliserons en particulier l'application de l'IO pour la systématisation de connaissances et sur la recherche d'informations basée sur les ontologies. (Cf. section III.2.4).

### **III.3.1 L'Ingénierie Ontologique comme base de systématisation de connaissances métiers**

Deux catégories d'ontologies (selon l'objet de conceptualisation) ont été considérées dans cette application de l'IO :

- Les ontologies de domaine (dénommées aussi ontologie d'artéfact [Kimatura, 2006] et ontologies de produit dans [Lee et al, 2009]) destinées à expliciter les connaissances de fonds d'un domaine d'ingénierie. (*"One of the deep necessities of ontologies of artifacts for engineering design is, we believe, the lack of explicit description of background knowledge of modeling."*) [Kimatura, 2006]
- Les ontologies de tâches [Chandrasekaran, 1999] ou encore d'activités [Gruninger,94] ou d'entreprise [Gandon, 2000] est utilisée pour gérer des tâches spécifiques dans les systèmes, telles que les tâches de diagnostic, de planification, de conception, de configuration, de tutorat, etc. Elle régit un ensemble de vocabulaires et de concepts qui décrit une structure de résolution des problèmes inhérente aux tâches et indépendante du domaine.[Psyché et al, 2003].

### III.3.1.1 Ontologies de domaine pour la systématisation des connaissances d'ingénierie

Trois principales dimensions de modélisation des systèmes d'ingénierie ont été considérées dans la littérature à savoir : la dimension ingénierie d'exigences systèmes, la dimension fonctionnelle et la dimension physique.

#### – IO appliquée à la dimension ingénierie d'exigences systèmes

[Lin et al., 1996] ont conçu une ontologie destinée à supporter le processus générique de gestion des exigences et à formaliser la représentation des exigences d'un système. Selon cette conceptualisation le concept d'exigence est spécialisé en exigences physiques, exigences structurelles, exigences fonctionnelles et exigences de coût. Ces classes d'exigences sont liées par des relations sémantiques de décomposition et de dérivation. Le contenu sémantique des concepts d'exigence sont exprimés par des primitives conceptuelles représentées dans une ontologie dénommée ontologie de produit. Cette ontologie est déclinée en trois dimensions de modélisation :

- Composants (*parts*) : permettant de définir les types de composants d'un système et leurs relations de spécialisation et décomposition
- Caractéristiques d'un système (*feature*) : définissant les caractéristiques associées aux composants d'un système et leurs relations de spécialisation et décomposition. Dans l'ontologie proposée les auteurs considèrent les caractéristiques fonctionnelles et géométriques d'un système.
- Paramètres d'un système « *parameters* » : décrivant les propriétés des composants et des caractéristiques d'un système. Par exemple la propriété diamètre décrit une caractéristique géométrique d'un composant. Chaque paramètre est caractérisé par un type de données (i.e. entier) et une unité de mesure (i.e. kg).

Les travaux de [Lin et al., 1996] ont été intégrés dans le projet TOVE (Toronto Virtual Enterprise) [Fox & Gruninger, 1998]. Une partie de TOVE est considérée comme une ontologie des besoins dans le domaine de la conception en ingénierie. Les primitives ontologiques représentant les exigences ainsi que d'autres plus spécifiques structurés en une hiérarchie décrite par la Figure III.8.

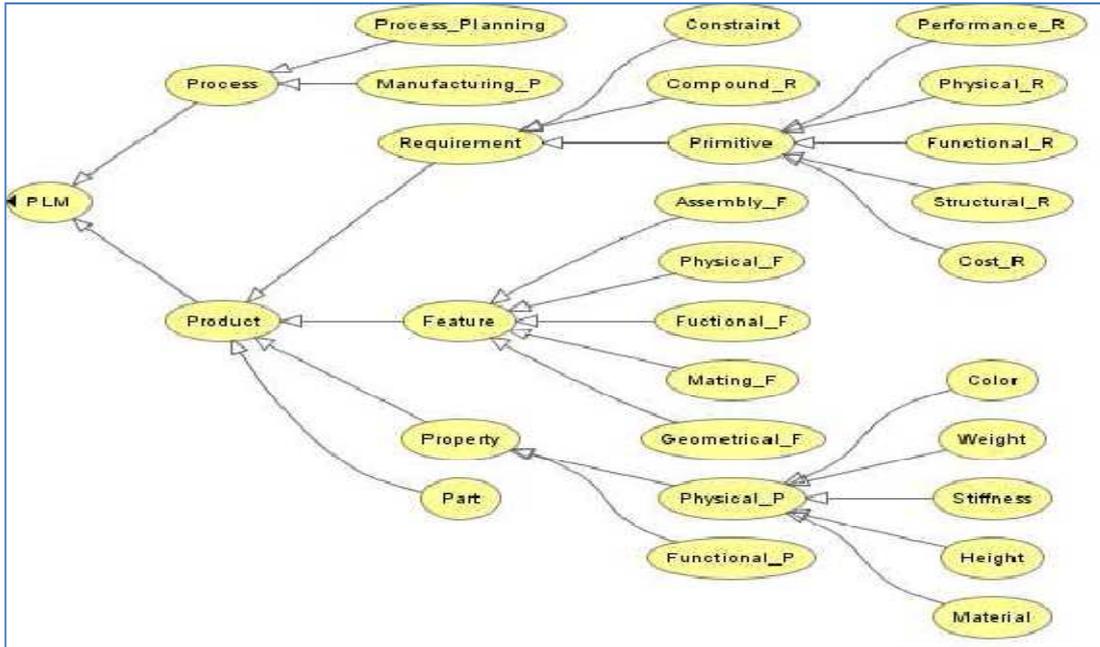


Figure III.8. Extrait de TOVE explicitant les exigences systèmes, [Fox & Gruninger, 1998].

#### – IO appliquée à la dimension fonctionnelle

[Kimatura, 2006] souligne que : « *Knowledge about functionality is at the conceptual level and thus tends to be ad hoc* ». Dans l'objectif de promouvoir le partage des connaissances d'ingénierie selon la dimension fonctionnelle [Mizoguchi et al, 2004 b] proposent un cadre ontologique multi-niveau pour la systématisation de ces connaissances. L'ontologie fonctionnelle fournit une taxonomie des fonctions génériques (appelées concepts fonctionnels). Les fonctions d'un système sont explicitement représentés par des classes de concepts et des relations de type subsomption et agrégation.

Une autre proposition de conceptualisation des fonctions systèmes a été présentée dans le cadre du projet NIST Design Repository Project [Skyman, 2000] sous la forme de catégories de fonctions systèmes et de catégories de flux d'entrées-sorties de ces fonctions. Un exemple de spécialisation d'une ontologie fonctionnelle développée dans le cadre du projet NIST est présenté dans la figure III.9.

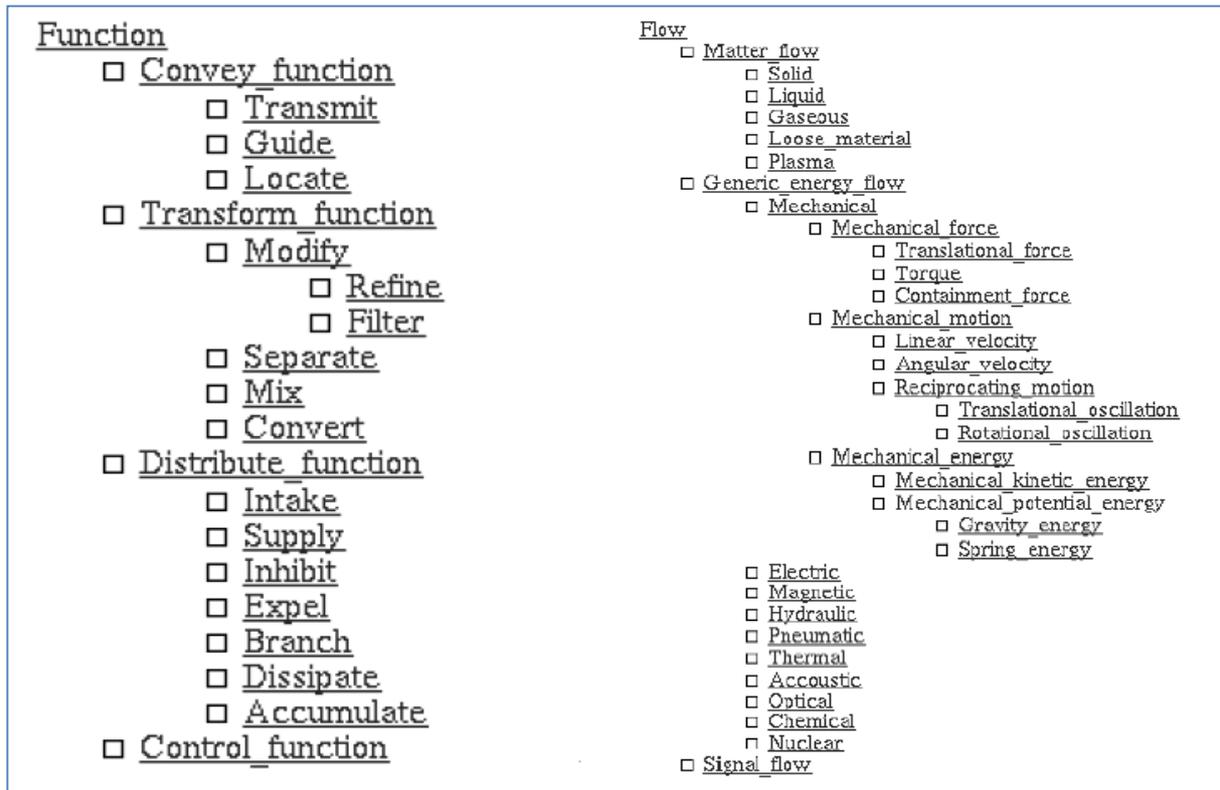


Figure III.9 ontologie fonctionnelle [Skyman, 2000]

#### – IO appliquée à la dimension Physique :

Deux ontologies de références ont été proposées pour expliciter les concepts et les théories sous-jacentes à l'ingénierie d'un système physique à savoir : YMIR [Alberts, 1993] et PhysSys [Borst et al. 1997]. Ces deux ontologies sont basées sur les théories générales de systèmes physiques. Nous détaillons dans ce qui suit la proposition PhysSys.

PhysSys est structurée sous la forme d'un ensemble d'ontologies d'ingénieries supportant de multiples points de vue de modélisation d'un système physique. Les points de vue suivants sont considérés: « meleology », « topology », « systems theory », « component », « process » et « mathematics for engineering domain » (EngMath). La *méréologie* est la théorie concernant les relations entre la partie et le tout. La topologie ajoute des connexions typées entre parties. ( i.e. relation, « Associé-à ».)

L'ontologie de composants définit les concepts terminaux et explicite l'aspect structurel des artefacts sous la forme de composants liées par des relations méronomiques ou topologiques. L'ontologie de processus explicite l'aspect comportemental de l'artefact (i.e. *in electrical systems, dynamic behaviour consists of the change of electrical charge*), l'ontologie EngMath explicite les concepts de quantités, unités et lois physiques. La structure du treillis ontologique et les relations d'interdépendances des ontologies de composants, processus et EngMath sont illustrées dans la figure III.10.

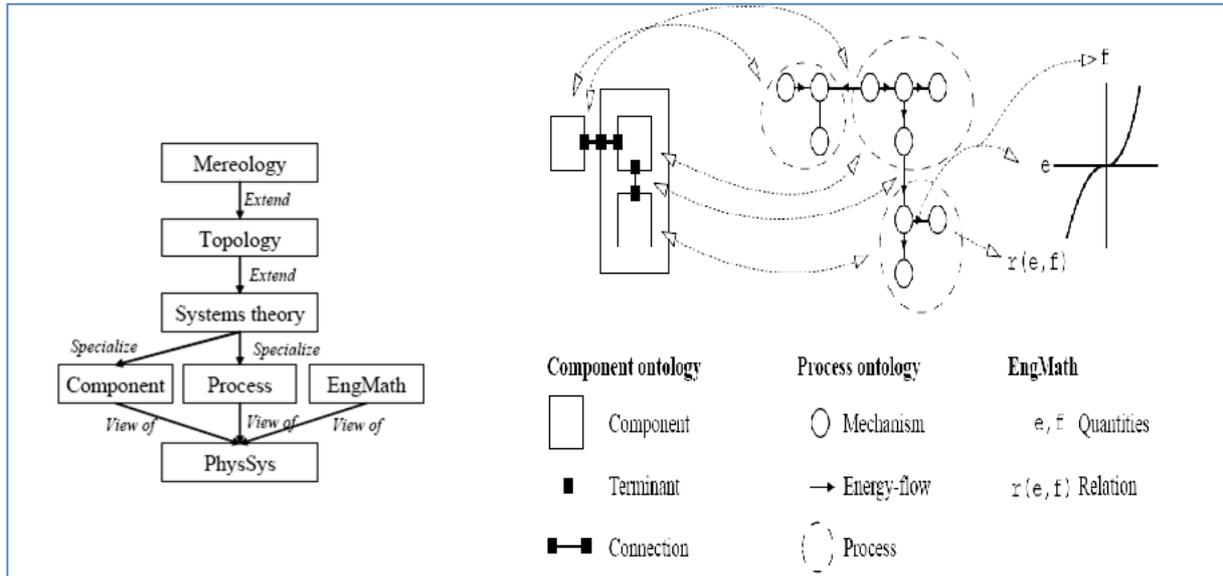


Figure III.10 treillis d'ontologies PhysSys [ Borst et al. 1997]

Dans cette perspective de conceptualisation des connaissances de domaines d'ingénierie, [Ahmed et al., 2005] propose une ontologie légère ( organisée sous forme de taxonomie de concepts) dénommée EDIT : Engineering Design Integrated Taxonomy, visant à indexer les documents d'ingénierie. Cette proposition couvre les dimensions ingénierie d'exigences, fonctions et composants et les relie à une ontologie de tâche. EDIT intègre quatre ontologies : l'ontologie de processus : explicitant les activités d'un processus d'ingénierie, l'ontologie de produit : explicitant les constituants physiques et leurs composants, l'ontologie de fonction fonctions explicitant les fonctions et les flux fonctionnels, l'ontologie *issues* : explicitant les besoins fonctionnels/non fonctionnels d'un système ainsi que les caractéristiques et l'environnement opératoire du système. (c.f. figure III.11)

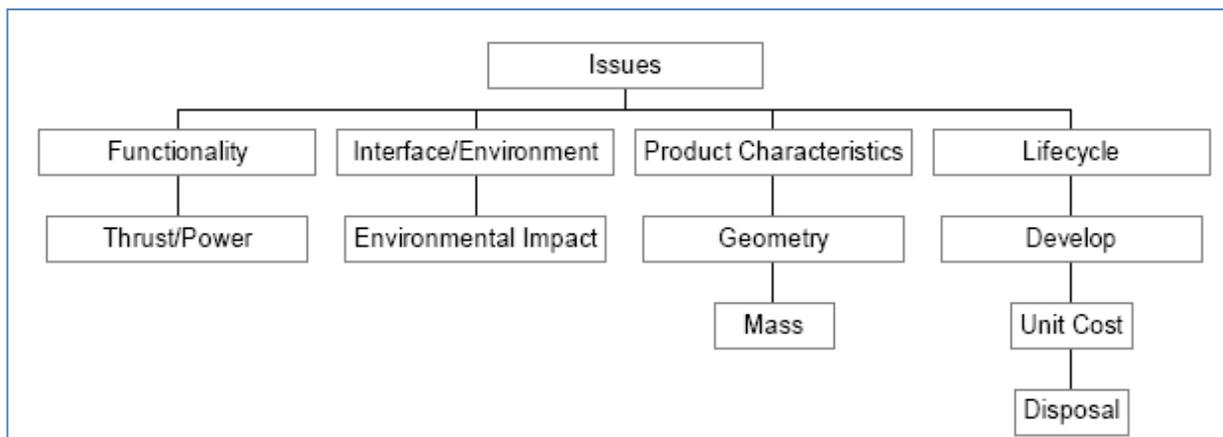


Figure III.11 : ontologie légère des exigences systèmes [Ahmed et al., 2005]

### III.3.1.2 Ontologies de tâches pour la systématisation des connaissances d'ingénierie

Cette classe d'ontologie a fait l'objet de plusieurs travaux du domaine de la gestion de connaissances organisationnelles [Gandon, 2000] [Dieng, 2000]. Dans ce contexte, ces ontologies explicitent la structure de l'organisation, les processus et les ressources informationnelles servant de support de connaissances. Selon [Gandon, 2000], « un modèle organisationnel est une représentation explicite de la structure, des activités, des processus, des ressources, des personnes, des comportements, des objectifs et des contraintes de l'organisation. L'ontologie correspondante capture les caractéristiques essentielles des entités modélisées et les formes de relations existantes entre elles dans un vocabulaire consensuel, non ambigu ». Ces ontologies agissent comme un modèle de contextualisation des connaissances organisationnelles.

Dans ce contexte plusieurs ontologies modélisant l'organisation ont été développées. Nous citons deux exemples représentatifs de cette classe d'ontologie : TOVE et Enterprise.

Le projet TOVE (Toronto Virtual Enterprise) [Fox & Gruninger, 98] vise à fournir une terminologie d'entreprise qui soit partagée par plusieurs applications développées dans différentes unités de l'entreprise (bureau d'études, fabrication, marketing, etc.) sous la forme d'un ensemble d'ontologies intégrées afin de modéliser à la fois les entreprises commerciales que les entreprises publiques. Un aperçu de la partie supérieure de l'ontologie TOVE est présenté dans la figure III.12.

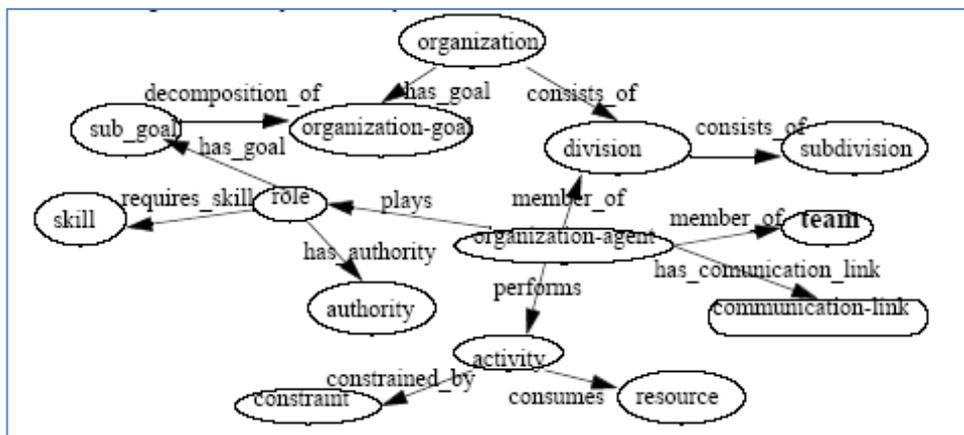


Figure III.12 Extrait de TOVE selon le point de vue tâche [Fox & Gruninger, 98]

Le projet Enterprise [Ushold et King., 1995] a pour objectif de promouvoir l'utilisation des systèmes à base de connaissance dans la modélisation de l'entreprise (The Enterprise partners). Ce projet ciblait l'innovation dans la gestion des entreprises et l'utilisation stratégique des TICs pour l'aide à la gestion du changement. Le contenu de l'ontologie est un ensemble de cinq sous ontologies ou sections: (1) une méta-ontologie (exemple : entité, relation, rôles) et une section consacrée au temps (exemple : intervalle de temps), (2) une section activité (exemple activité de planification), (3) une section organisation (exemple

unité d'organisation, partenaire) (4) une section stratégie ( exemple : mission, facteur de succès) et (5) une section marketing ( par exemple : vente, client).

Dans le cadre de notre contexte d'IS, les travaux de [Sarder et al, 2007] ont abordé la problématique de conceptualisation d'ontologies orientées processus d'IS. L'objectif étant de définir une ontologie noyau de la communauté d'IS à partir des normes, des standards d'IS comme ISO-15288 (ISO, 2002), EIA-632 (GEIA 1999) et des modèles de maturités comme CMMI (SEI 2006). Dans sa version actuelle, cette ontologie donne une conceptualisation possible des processus techniques, des processus de gestion techniques de projets, des processus de gestion d'entreprise, des acteurs et des produits d'IS. (c.f. figure III.13)

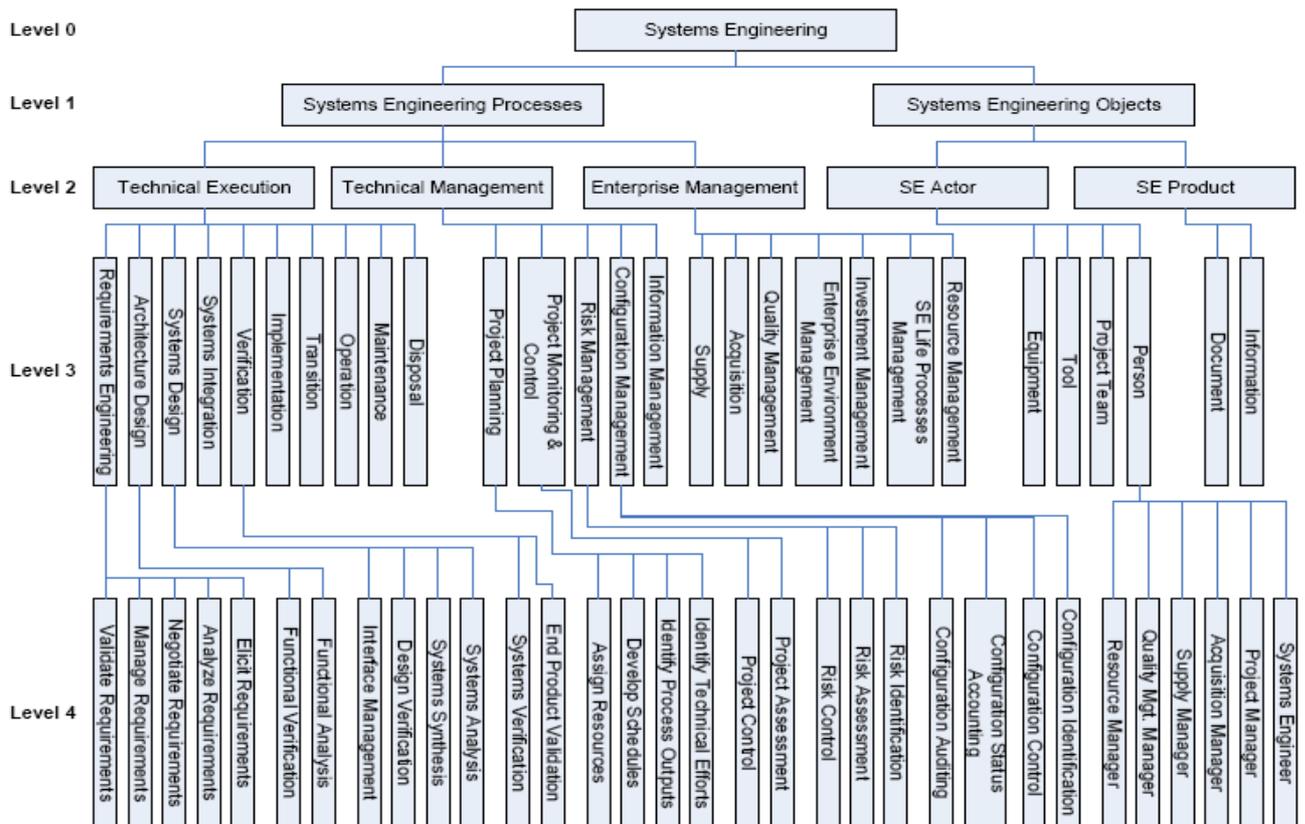


Figure III.13 Ontologies de tâches en IS [Sarder et al, 2007]

Travaux étudiés	Couverture du métier d'IS		Fondements de l'ontologie	Modèle de représentation et de raisonnement		Support méthodologique
	Ontologie de domaine	Ontologie de processus		Représentation de connaissances	Capacités inférentielles	
[Lin et al., 1996]	Exigences, composants, Caractéristiques produits, paramètres	Non	Cycle en V d'IS	Logique de premier ordre	Oui	Non
TOVE [Fox & Gruninger, 98].	Exigences, composants, Caractéristiques produits, paramètres	Activité, organisation, ressources	Théorie organisationnelle	KIF	Oui	Oui
[Mizoguchi,2004 b]	Fonctions, comportement	Non	Théorie de systèmes	RDF/RDFS	Non	Par réutilisation d'ontologie
[ Skyman, 2000]	Fonctions, flux fonctionnels	Non	"models based on inputs and outputs of flows" [Pahl et Beitz, 1988]	Non renseigné		Non
[Saema, 2005]	Issues, product, function	Processus	Interview concepteurs systèmes	Non renseigné		Non
PHYSIS [Borst et al. ,97]	Composants, comportement, modèles mathématiques	Non	Théorie de système	Ontolingua	oui	Non
[ Ushold et al., 1998]	Non	Organisation, activité	Théorie organisationnelle		Non	Oui
[Sarder et al, 2007, 2008]	Non	Processus d'IS, acteur, produits	Normes d'IS	IDEF5	Non	(DKAP) ontology modeling methodology (Sarder 2006)

Tableau III.2 : Etude comparative des travaux d'IO comme systématisation de connaissances métiers

### III.3.2 L'Ingénierie Ontologique comme base de partage de connaissances

Cette application de l'IO est à la base des travaux du domaine du Web Sémantique et des Systèmes de Gestion de Connaissances de nouvelles générations<sup>40</sup>. Le concept fondateur de cette classe d'application est l'annotation de ressources décrites de manière informelle, par des descriptions formelles de leurs contenus.[Euzénat, 2005]. Dans le cadre du web sémantique, [Euzénat 2005] formalise l'annotation sémantique sous forme d'une fonction de l'ensemble des documents vers l'ensemble des représentations formelles de la connaissance exprimées dans des ontologies. Par analogie entre les ressources du web et les ressources d'une entreprise, le projet O'COMMA [Gandon, 2000] propose de matérialiser un système de gestion de connaissances à travers un «web sémantique d'entreprise (WSE)» ou « web sémantique d'organisation (WSO)» ( figure III.14) en utilisant les ontologies qui fournissent un cadre formel pour décrire les différentes sources de connaissances de l'organisation et qui guident la création d'annotations sémantiques facilitant la description, le partage et l'accès à ces sources. Les principales composantes de ce web sémantique d'entreprise sont les suivantes :

- Les ressources : peuvent être des bases de données, des personnes, des documents (dans tous les formats), des services/logiciels, etc.
- Les ontologies : décrivant le vocabulaire partagé par les différentes communautés de l'entreprise.
- Les annotations sémantiques : décrivant des métadonnées sur les ressources en se basant sur les concepts et les relations de l'ontologie.

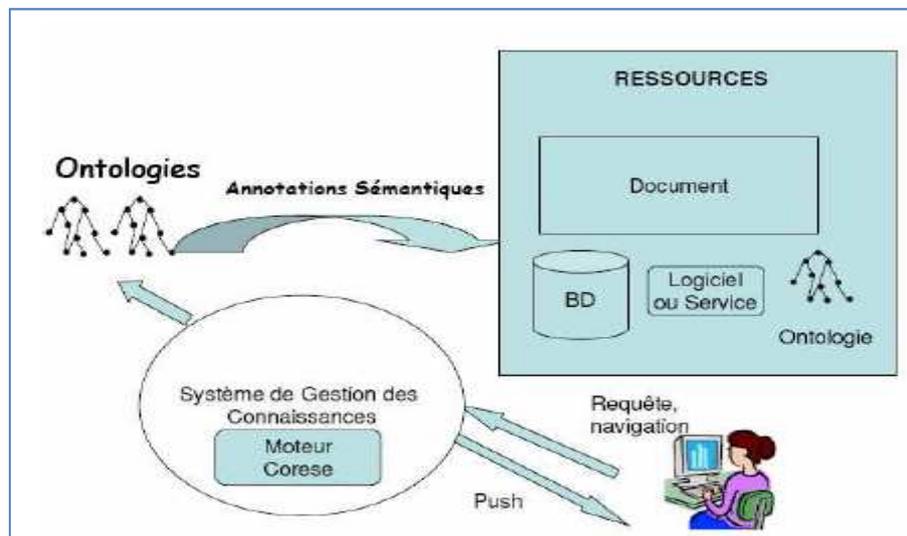


Figure III.14 : Web Sémantique Organisationnel [Gandon ,2000]

<sup>40</sup> Ou web sémantique organisationnel

L'opérationnalisation d'un Web Sémantique Organisationnel nécessite l'expression des ontologies et des annotations sémantiques dans un formalisme de représentation de connaissances. Dans le contexte des travaux du web sémantique une pyramide de langages de représentation à été proposée. RDF fournit une première réponse en proposant un langage de représentation de métadonnées à base de triplets (sujet, prédicat, objet). RDFS étend ce premier langage en proposant des primitives spécifiques d'expression de connaissances ontologiques séparées des connaissances factuelles. OWL tend lui-même RDFS en proposant un formalisme plus complet d'expression de connaissances factuelles et ontologiques. Enfin, SWRL vise à étendre OWL par la prise en compte d'un langage de règles. [Leclère et al., 2005]. Pour exploiter (retrouver) les connaissances décrites par ces langages, plusieurs raisonneurs sur le web sémantiques ont été proposés [Gandon , 2008]. Citons par exemple Sesame [Broekstra,2005] qui est une architecture générique pour le stockage persistant de RDF(S) dans une base de données et l'interrogation de RDF(S) avec le langage RQL et Corese [Corby,2008] qui est un moteur de recherche sémantique basé sur les graphes conceptuels implantant RDF/S et SPARQL. Avec la standardisation de OWL DL, les moteurs à base de logiques de descriptions ont pris une importance particulière, nous Citons par exemple Racer et, Pellet. Ces moteurs offrent les opérations de raisonnement de base des logiques de descriptions (identification ou test d'instanciation d'un concept, classification, subsomption [Périé, 2008])

L'intérêt de la représentation formelle du contenu sous forme d'annotations sémantiques a été dans plusieurs travaux d'ingénierie.

En Ingénierie des Systèmes d'Information, le système SRS (Semantic Reuse System) [Antunes et al, 2007] est basé les standards de web sémantique pour représenter les artefacts produits en cours d'un processus de développement. Ces artefacts (i.e. Documents de spécification, modèles de conception, composants logiciels) sont annotés par les primitives conceptuelles de deux ontologies : une ontologie de catégorie d'artefacts et une ontologie de domaine. Les annotations sont capitalisées dans un référentiel de *SDKE (Software Development Knowledge Element)*

[Regli et al, 2003] ont développés une ontologie dans le formalisme des Logiques de Description pour annoter les artefacts d'ingénierie selon un point de vue fonctionnel. Ces travaux ont été appliqués dans le domaine des produits électromécaniques

Dans le domaine de l'ingénierie automobile [Gao et al. 2003] ont proposés une intégration d' système de gestion de produit (PDM) avec une ontologie de domaine. L'objectif est d'offrir des capacité de gestion de connaissances pour la phase de conception de produits.

[Mizoguchi, et al, 2004] ont proposés un système d'annotation sémantique d'artefacts d'ingénierie base sur leur cadre ontologique fonctionnel (c.f. III.3.1).

[Zhang et al, 2006] ont étudié l'applicabilité des technologies du web sémantique, en particulier le langage OWL au domaine d'ingénierie de la conception. Ils proposent une approche générique d'annotation de documents d'ingénierie à base d'ontologies explicitant les connaissances fonctionnelles et structurelles de produits.

Ces travaux ont montré l'apport de l'annotation sémantique dans le processus de recherche d'artefacts d'ingénierie. Mais chaque approche propose une méthodologie d'annotation spécifique et dépendante du métier d'ingénierie dans lequel elle s'applique. Les ontologies utilisées dans ce contexte sont difficilement réutilisables.

## III.4 Conclusion

Nous avons étudié dans ce chapitre les fondements de la discipline de l'Ingénierie Ontologique. Nous nous sommes ensuite focalisé sur deux applications potentielles de cette discipline pour la gestion des connaissances dans différents secteurs d'activités d'Ingénierie Système à savoir : la systématisation des connaissances métiers et le partage de connaissances.

- **Positionnement par rapport à l'Ingénierie Ontologique comme base de systématisation de connaissances métiers** : les travaux étudiés présentent des résultats significatifs en termes de représentation formelle et consensuelle des connaissances de domaine et de processus d'ingénierie. Mais la plupart des propositions offrent une couverture partielle des dimensions de modélisation de connaissances métiers d'ingénierie (tableau III.2). Ces propositions serviront de base à la définition d'un cadre ontologique générique unifiant les modèles ontologiques étudiés tout en élargissant la couverture des processus d'IS et le champ d'applicabilité de ces modèles dans le cadre de différents secteurs d'activités de l'IS.
- **Positionnement par rapport à l'Ingénierie Ontologique comme base de partage de connaissances** : les travaux étudiés sont spécifiques à des métiers d'ingénierie et les ontologies utilisées dans le processus d'annotation sémantique sont difficilement réutilisables (à l'exception des travaux de [Mizoguchi, 2003]). De plus ces travaux sont principalement orientés vers l'annotation de modèles d'ingénierie. Dans le cadre de nos travaux nous nous intéressons plus particulièrement à la formalisation des expériences qui émergent au cours des processus techniques d'IS. Ces expériences résultent de la mise en œuvre effective des connaissances métiers dans des situations de résolution de problèmes d'ingénierie. Nous étudierons les apports du concept d'annotation sémantique comme base de formalisation et de réutilisation d'expériences d'ingénierie tout en nous basant sur la systématisation des connaissances métiers d'IS.

Ainsi, la contribution majeure de nos travaux consiste à combiner et à tirer profit des deux applications d'Ingénierie Ontologique susmentionnées.

Pour répondre aux objectifs de Gestion de connaissances en IS (cf. Chapitre I, section I.4.2) nos travaux se situent à l'intersection de ces deux applications de l'IO, et s'articulent autour de deux volets :

- Modélisation de connaissances métiers d'IS (objet du chapitre IV) : Ce chapitre étudie le rôle de l'IS comme base de systématisation de connaissances métier d'IS. Nous proposons un cadre ontologique générique multi-facette et multi-strate pour la modélisation de connaissances d'ingénierie de systèmes. Ce cadre ontologique a pour objectif de couvrir les diverses dimensions décrivant les processus d'IS à savoir : la dimension « système à faire » capturant explicitement une conceptualisation du domaine du problème et des exigences, du domaine fonctionnel, et du domaine physique (ou organique (assemblage, composants matériels ou logiciels.) système étudié. La dimension « système pour faire » capturant explicitement le contexte d'un projet d'ingénierie sous la forme de processus, ressources humaines et matérielles, et artefacts d'ingénierie.
- Capitalisation et partage d'expériences d'ingénierie (objet du Chapitre V) : Nous analysons dans ce chapitre les apports de l'annotation sémantique [Euzenat, 2005] comme base de capitalisation et de partage d'expériences d'IS. Ces expériences sont dénommées Connaissances Métiers Situées parce qu'elles émergent dans des situations de résolutions effectives de problèmes d'Ingénierie et tiennent compte du contexte métier, organisationnel et décisionnel des processus d'ingénierie. Pour gérer ces connaissances nous proposons deux modèles :
  - Un modèle conceptuel de capitalisation de Connaissances Métiers Situées (CMS) : Une CMS est une représentation des expériences de projets d'Ingénierie sous forme d'annotations sémantiques multidimensionnelle basées sur le cadre ontologique que nous proposons. Quatre dimensions d'annotation sont proposées : la dimension situationnelle, la dimension téléologique (but d'ingénierie), la dimension argumentative et la dimension décisionnelle.
  - Un modèle de partage de CMS : Le processus de partage de connaissances consiste en l'appariement des représentations des dimensions situationnelles et téléologiques des CMS avec celles des représentations des situations de projets en cours. Cet appariement se base sur des critères à la fois sémantiques et pragmatiques (dépendante de raisonnements typiques dans une situation d'IS). Les résultats du processus d'activation par l'appariement sont ensuite organisés sous forme de treillis de CMS traduisant une hiérarchisation des connaissances nécessaires et/ou potentiellement utiles dans une situation d'ingénierie.

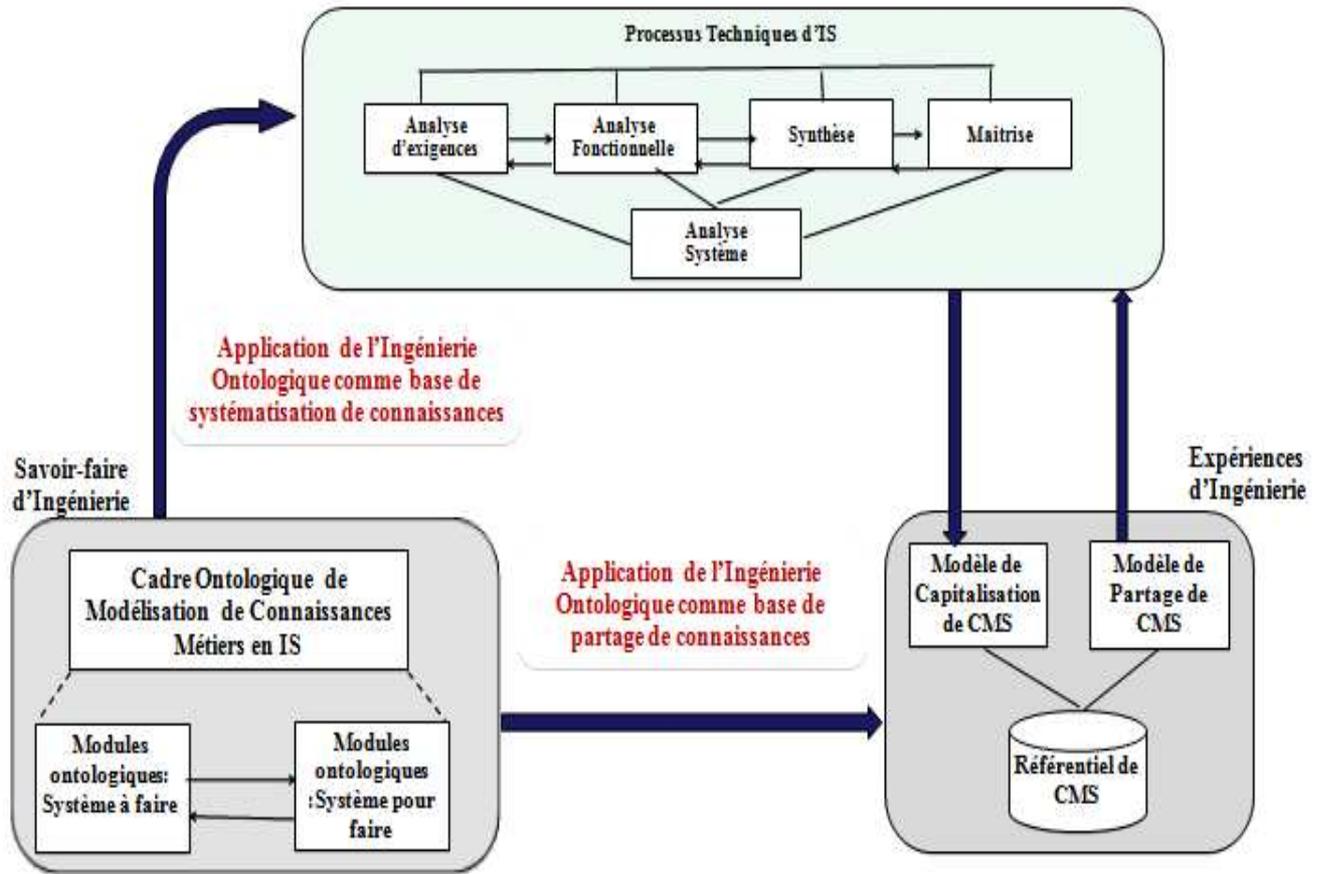


Figure III.15 : Proposition d'un cadre conceptuel de Gestion de Connaissances en IS couvrant deux applications de l'Ingénierie Ontologique

## **Deuxième partie**

# **Proposition d'un cadre ontologique générique de Modélisation, de Capitalisation et de Partage de Connaissances Métiers situées en Ingénierie Système**

## Chapitre IV

# Proposition d'un Cadre Ontologique de Modélisation de Connaissances Métiers en IS : OntoIS

---

### Plan du chapitre

---

IV.1 Introduction

IV.2. Méthodologie d'ingénierie du cadre ontologique OntoIS

IV.3. Architecture générale du cadre ontologique OntoIS

IV.4 Conceptualisation du cadre ontologique OntoIS

IV.4.1 Strate Fondationnelle

IV.4.2 Strate Noyau

IV.4.2.1 Facette système à faire

IV.4.2.2 Facette système pour faire

IV.4.3 Exemple de spécialisation de Patrons Conceptuels Ontologiques

IV.5 Conclusion : Applications potentielles du cadre Ontologique

## **IV 1. Introduction**

Dans ce chapitre nous abordons la problématique de modélisation conceptuelle des connaissances métiers en IS en nous basons sur les concepts de l'Ingénierie Ontologique.

L'ingénierie Ontologique appliquée à l'IS est un problème complexe, lié au nombre de primitives conceptuelles devant être considérées, les différents secteurs d'activités de l'IS, et l'interdisciplinarité de chacun de ces secteurs d'activité. Pour rendre compte de cette complexité, nous proposons un cadre générique de modélisation de connaissances métiers d'IS intégrant plusieurs modules ontologiques. L'objectif est d'offrir une modélisation consensuelle et partagée des savoir-faire métiers de l'IS (Engagement sémantique selon [Bachimont, 2000]) en couvrant les différentes dimensions de descriptions des processus d'IS ainsi que les différents secteurs d'activités de ce métier.

Nous présentons dans ce chapitre les différents modules ontologiques du cadre proposé, en mettant l'accent sur la méthodologie adoptée pour la conceptualisation des connaissances métiers d'IS.

## **IV.2. Méthodologie d'Ingénierie du cadre ontologique OntoIS**

Dans le souci de concevoir un cadre ontologique générique explicitant les connaissances métiers d'IS, nous avons adopté une méthodologie basée d'une part sur la spécialisation d'ontologies fondationnelles [Guarino, 1998] et d'autre part sur la conception de patrons conceptuels ontologiques noyaux [Gangemi et al., 2003]

Une telle méthodologie de conception vise à maîtriser plusieurs sources de complexité liées à l'interdisciplinarité de l'IS:

- Une complexité de modélisation conceptuelle, en permettant de modéliser des systèmes complexes comme le sont les systèmes à différents niveaux d'abstraction.
- Une complexité de modélisation consensuelle en permettant de réutiliser des modules ontologiques déjà conçus et évalués dans le cadre d'autres domaines d'Ingénierie.

Nous analysons dans ce qui suit les apports des ontologies fondationnelles et des patrons conceptuels ontologiques.

## IV.2.1 Rôle des ontologies fondationnelles

Les ontologies fondationnelles sont des ontologies de référence de haut niveau, basées sur des théories ontologiques étudiées dans l'ontologie formelle [Temal, 2008]. Ces théories se basent sur des principes provenant d'analyses relevant d'autres domaines, principalement la philosophie, la linguistique, la logique et les mathématiques. Les avantages majeurs de l'utilisation des ontologies fondationnelles dans les processus d'ingénierie ontologique ont été recensés dans les travaux de [Guarino, 1998][Oberle et al, 2006][Guizzardi, 2006] :

- Base de modélisation : les ontologies fondationnelles fournissent un point de départ pour construire de nouvelles ontologies. Au lieu de modéliser à partir de zéro, l'utilisation des ontologies fondationnelles fournit aux concepteurs un ensemble d'entités ontologiques prédéfinies, qu'ils peuvent réutiliser pour les ontologies qu'ils veulent construire.
- Clarté conceptuelle : les ontologies fondationnelles fournissent un point de référence pour une comparaison rigoureuse entre plusieurs approches ontologiques possibles, et un cadre de travail pour analyser, harmoniser, et intégrer des ontologies existantes.
- Partage d'ontologies : En cas d'incompatibilité entre modèles sémantiques de domaines, les ontologies fondationnelles offrent une plate-forme appropriée pour la négociation des significations et l'aboutissement à une sémantique consensuelle.

Plusieurs ontologies fondationnelles ont été proposées dans la littérature, parmi lesquelles nous citons : SUMO (Suggested Upper Merged Ontology) [Niles et al., 2001], DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering).[Masolo et al., 2003], BFO (Basic Formal Ontology) [Grenon, 2004] et BWW (Bunge, Wand, Weber) [Wand, 1996].

Dans le contexte de nos travaux nous avons choisi l'ontologie DOLCE comme cadre de référence décrivant les connaissances métiers de l'IS. Ce choix est motivé par la disponibilité de nombreuses extensions de cette ontologie dans plusieurs domaines comme la neuroimagerie [Temal, 2008], l'audiovisuel [Isaac, 2005], les systèmes mécatroniques [Damjanović et al., 2007]. De plus, les travaux de [Graves, 2008] montrent l'adéquation de DOLCE à la description des concepts de l'IS en spécialisant un sous ensemble<sup>41</sup> des concepts de haut niveau de cette ontologie dans le domaine des systèmes avioniques.

---

<sup>41</sup> Un sous ensemble relatif aux concepts de DOLCE décrivant les *artefacts physiques, qualité et région*.  
<http://www.w3.org/2005/Incubator/w3pm/wiki/images/4/49/OWLED2008-presentation.pdf>

## IV.2.2. Ingénierie ontologique et patrons de conception

La notion de patron de conception, importée depuis la communauté du génie logiciel, est une idée de plus en plus populaire dans la communauté de l'Ingénierie Ontologique. Les Patrons de conception [Gamma, 1995] du génie logiciel ont pour ambition de présenter des solutions attestées à des problèmes de conception logicielle courants dans le domaine de la programmation orientée objet. Servant de guides pratiques pour les développeurs, ils décrivent des conceptions logicielles consensuelles à un niveau plus abstrait que celui des classes et des instances normalement utilisées dans les systèmes implémentés. Un tel effort d'abstraction est en effet nécessaire au consensus recherché, et donc autorise une plus grande réutilisabilité. [Isaac, 2005]

Ces objectifs généraux sont de fait relativement proches de ce que l'on recherche en représentation de connaissances ontologiques : des spécifications de conceptualisation partagées et réutilisables, mais aussi exploitables dans des systèmes concrets. Inspirés par ces similarités, des chercheurs comme [Blomqvist et al., 2005], [Gangemi, 2006] ont proposé d'adapter ces concepts de génie logiciel à l'Ingénierie Ontologique. Plusieurs classes de patrons ontologiques<sup>42</sup> ont été proposées : patrons d'application, patrons architecturaux, patrons conceptuels, patrons sémantiques et patron syntaxiques.

La classe de patron la plus proche de nos objectifs de conceptualisation de connaissances métiers d'IS est celle des patrons ontologiques conceptuels de contenu (CODeP) [Gangemi, 2006].

Les CODePs<sup>43</sup> visent à apporter des solutions réutilisables à des problèmes récurrents de modélisation d'un domaine. Parmi les principales caractéristiques de cette classe de patron nous citons :

- Ils encodent les savoir-faire et les bonnes pratiques des experts d'un domaine.
- Ils permettent de capturer les concepts et les relations sémantiques consensuelles d'un domaine donné indépendamment d'un langage de représentation de connaissances.

---

<sup>42</sup> Application Patterns - Purpose, scope, usage and context of the implemented ontology or ontologies, including interfaces and relations to other systems.

• Architecture Patterns - A description of how to combine or arrange implemented Design Patterns in order to fulfill the overall goal of the ontology.

• Design Patterns - A small collection of Semantic Patterns that together create a common and generic construct for ontology development.

• Semantic Patterns - Language independent description of a certain concept, relation or axiom. A meta-description of a Syntactic Pattern.

• Syntactic Patterns - Language specific ways to arrange representation symbols, to create a certain concept, relation or axiom. [Blomqvist et al., 2005]

<sup>43</sup> <http://ontologydesignpatterns.org/wiki/Category:ContentOP>

Afin d'offrir une plus grande réutilisabilité, des CODEPs génériques constitués de concepts et de relations de l'ontologie fondationnelle DOLCE ont été définis<sup>44</sup> [Gangemi, 2006]. Ces CODEPs capturent des représentations neutres de problèmes de modélisation pouvant être spécialisées pour obtenir les connaissances ontologiques au niveau d'un domaine ou d'une application. [Gangemi et al., 2004] montrent la spécialisation d'un patron conceptuel générique : description et situation [Gangemi et al., 2003] dans le domaine biomédical.

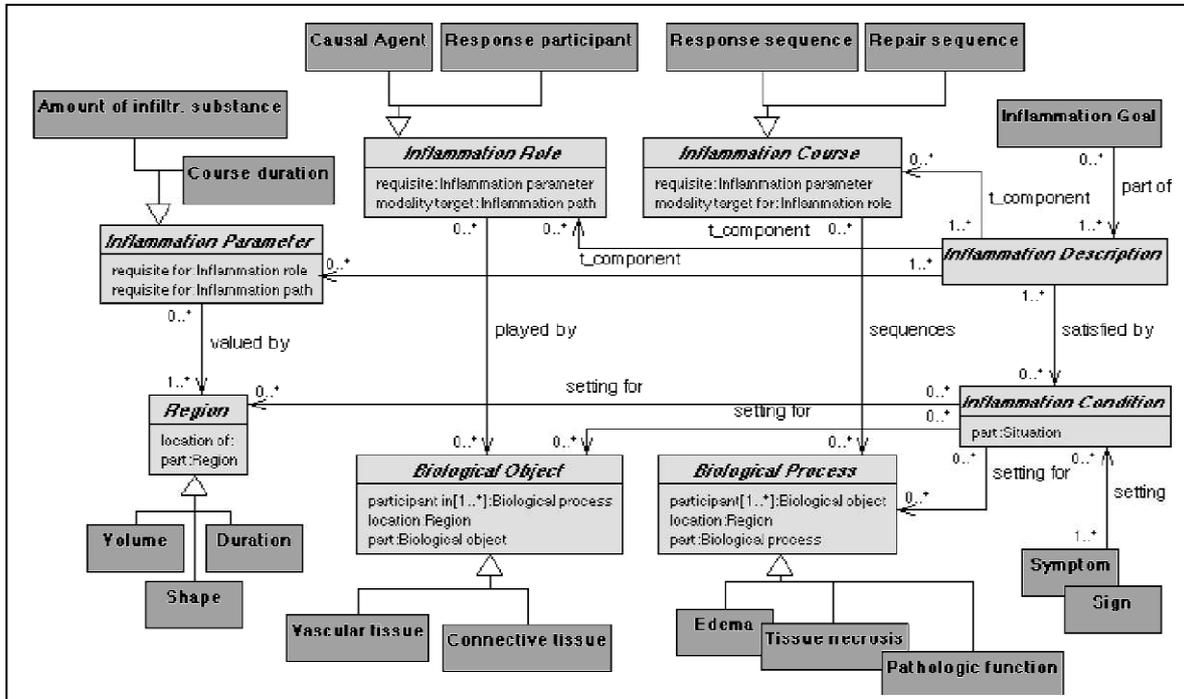


Figure IV.1 Spécialisation du patron ontologique description et situation (D&S) pour s'adapter au domaine de la description de processus d'inflammation et de leurs résultats [Gangemi et al., 2004]

Dans le cadre de nos travaux, nous adaptons cette notion de CODEPs<sup>45</sup> pour capturer les savoir-faire invariants du métier de l'IS : nous proposons d'expliciter ces savoir-faire sous forme d'un ensemble de patrons génériques répondants aux objectifs de modélisation communs à la communauté de l'IS. Ces patrons visent à alléger le processus complexe de construction d'ontologies de différents secteurs d'activités d'IS, en offrant les concepts et les relations sémantiques à spécialiser dans un domaine d'ingénierie.

<sup>44</sup>[http://wiki.loa-](http://wiki.loa-cnr.it/index.php/LoaWiki:Ontologies#DOLCE.2B_Ultralite_for_lightweight_applications_and_related_plugins)

[cnr.it/index.php/LoaWiki:Ontologies#DOLCE.2B Ultralite for lightweight applications and related plugins.](http://wiki.loa-cnr.it/index.php/LoaWiki:Ontologies#DOLCE.2B_Ultralite_for_lightweight_applications_and_related_plugins)

Ces patrons sont indépendants des langages de représentations de connaissances ontologiques. Dans le cadre des travaux de cette équipe sur le web sémantique, des exemples d'expression de ces patrons dans le langage OWL est fourni.

<sup>45</sup> Patrons ontologiques conceptuels dans la suite de ce document.

### IV.3. Architecture générale du cadre ontologique OntoIS

L'Architecture générale du cadre ontologique OntoIS est organisée autour de trois vues :

- **Vue connaissances métiers** : dont l'objectif est de couvrir deux dimensions de description d'un système :
  - la dimension « système à faire » capturant explicitement une conceptualisation du domaine du problème et des exigences, du domaine fonctionnel, et du domaine organique (assemblage, composants matériels ou logiciels.) d'un système.
  - la dimension « système pour faire » capturant explicitement le contexte d'un projet d'ingénierie sous forme de processus, rôles, ressources technologiques, et produits. Ces connaissances métiers seront formalisées sous la forme de patrons conceptuels ontologiques.
  
- **Vue domaine d'ingénierie** : dont l'objectif est de couvrir l'aspect pluridisciplinaire du métier d'IS et de permettre aux différents secteurs d'activités d'ingénierie de concevoir des modules ontologiques par spécialisation de la vue connaissances métiers. Cette vue est organisée en quatre strates
  - Une strate supérieure : qui correspond à la spécialisation de l'ontologie fondationnelle DOLCE au métier de l'IS
  - Une strate noyau : qui correspond aux concepts invariants du métier de l'IS
  - Une strate domaine : qui correspond à la spécialisation de la strate noyau aux concepts spécifiques des différents secteurs d'activités de l'IS (Telecom, automobile, Aéronautique..) et des différents métiers de l'IS (génie logiciel, génie électrique, génie mécanique...);
  - Une strate applicative : spécialisant la strate domaine pour tenir compte des bonnes pratiques et des systèmes (Par exemple, une ligne de produit) développés par une organisation particulière.
  
- **Vue représentation de connaissances** : cette vue constitue une dimension orthogonale aux vues connaissances métiers et domaine d'ingénierie. Elle correspond aux choix d'un modèle de représentation et de raisonnement sur les connaissances ontologiques offrant les primitives logiques permettant de créer des ontologies de domaine. cette vue peut s'organiser en quatre niveaux d'abstraction. Une méta méta ontologie offrant les primitives de représentation de connaissances minimaux ( concepts, relations et axiomes). Une méta ontologie (ou ontologie de représentation [Furst, 2004] qui correspond au choix d'un formalisme (exemple : Graphes Conceptuels) ou un langage (exemple : OWL) de représentation de connaissances. Un niveau ontologique correspondant à l'instanciation des vues connaissances métiers et domaine d'ingénierie à l'aide d'un formalisme de représentation. Un niveau instance qui correspond à l'expression de faits du monde réels en instanciant le modèle ontologique.

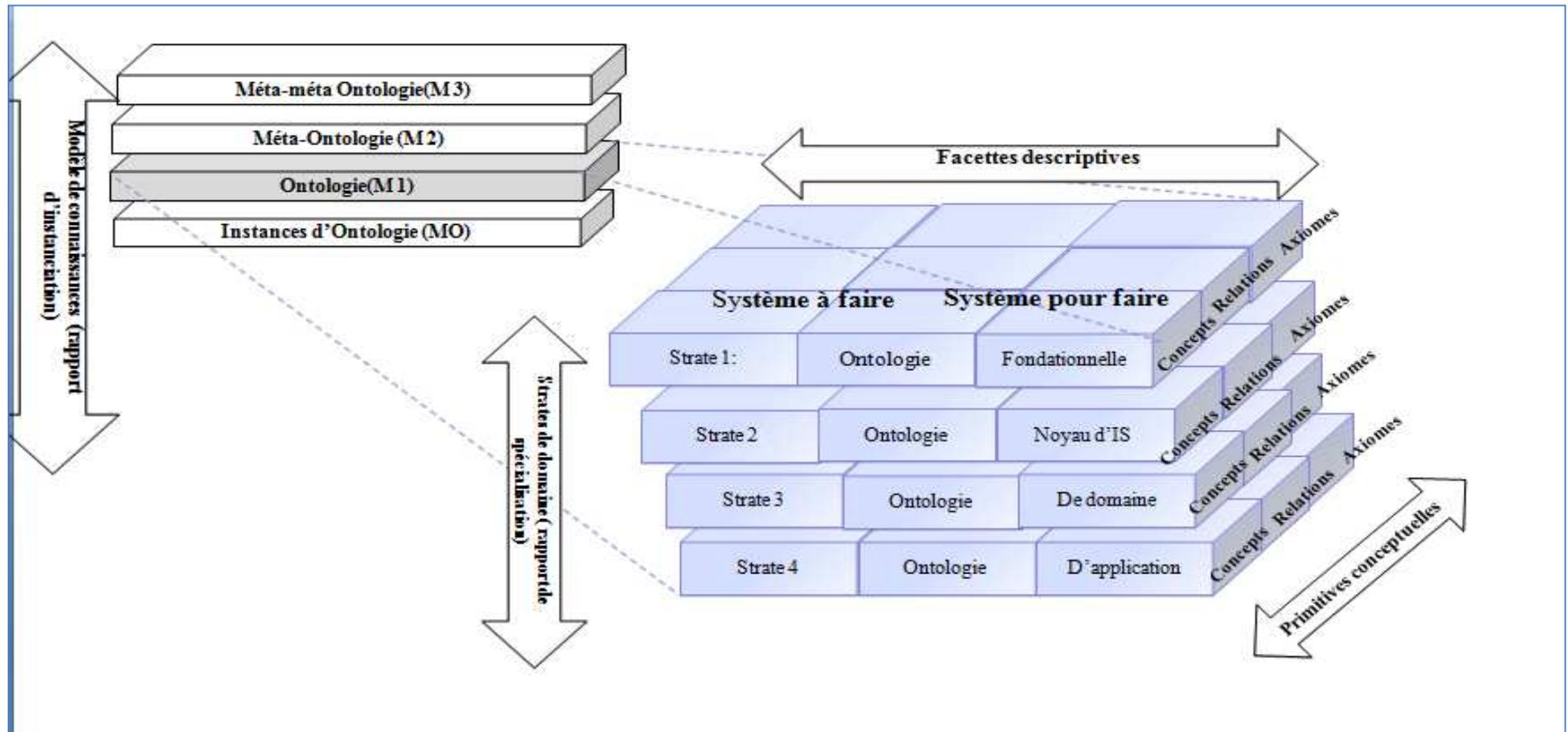


Figure V.2 Architecture générale du cadre ontologique OntoIS

## IV.4. Conceptualisation du cadre ontologique OntoIS

Dans cette section nous appliquons les principes méthodologiques de spécialisation d'ontologie fondationnelle et de conception de patrons ontologiques noyaux pour formaliser les modules ontologiques du cadre OntoIS.

### IV.4. 1. Strate fondationnelle : extension de l'ontologie DOLCE

Face à la multiplicité et à la complexité des concepts décrivant le métier d'IS, l'extension l'ontologie fondationnelle DOLCE nous a permis de clarifier le cadre général des catégories que nous voulons représenter dans le cadre ontologiques OntoIS. Nous présentons dans ce qui suit les principaux concepts de haut niveau de l'ontologie DOLCE et l'alignement des concepts de l'IS (cf. chapitre I) par rapport à cette dernière (cf. figure IV.3) (cf. figure IV.4)

Le domaine de DOLCE [Masolo et al., 2003], i.e. l'ensemble des entités classées par les concepts de l'ontologie (appelé ensemble de particuliers « Particulars »), est partitionné en quatre sous-domaines :

- Les Endurants sont des entités qui existent dans le temps, et qui ont une présence totale à un instant donné. Parmi les Endurants sont distingués les *Physical Objects*, spécialisés par des *physical agentive agent* et les *Non-Physical Objects*. Le domaine des *Non-physical Objects* recouvre les entités sociales et les entités cognitives (objets mentaux). En IS, les objets physiques correspondent par exemple aux constituants d'un système. Les agents physiques correspondent aux ressources humaines ou acteurs qui sont engagés dans un processus d'ingénierie. Ces processus mettent en œuvre des activités cognitives pour générer de multiples artefacts qui correspondent à des objets mentaux.
- Les Perdurants : Ce sont des entités qui « occurent dans le temps » et auxquelles participent des Endurants. Parmi les Perdurants sont distingués les événements et les processus. Comme exemple de spécialisation d'événement en IS, nous pouvons considérer les besoins des parties prenantes d'un système déclenchant une mise en œuvre de processus techniques d'ingénierie pour répondre à ces besoins.
- Endurants et Perdurants sont caractérisés par des Qualités inhérentes que nous percevons et/ou mesurons. Ces Qualités prennent leurs valeurs dans des Régions (intervalles ou zone) de valeurs. En IS, nous disposons de plusieurs exemples d'entités mesurables comme les contraintes de coût et de délais de projets, ou encore les paramètres physiques et les performances d'un système.

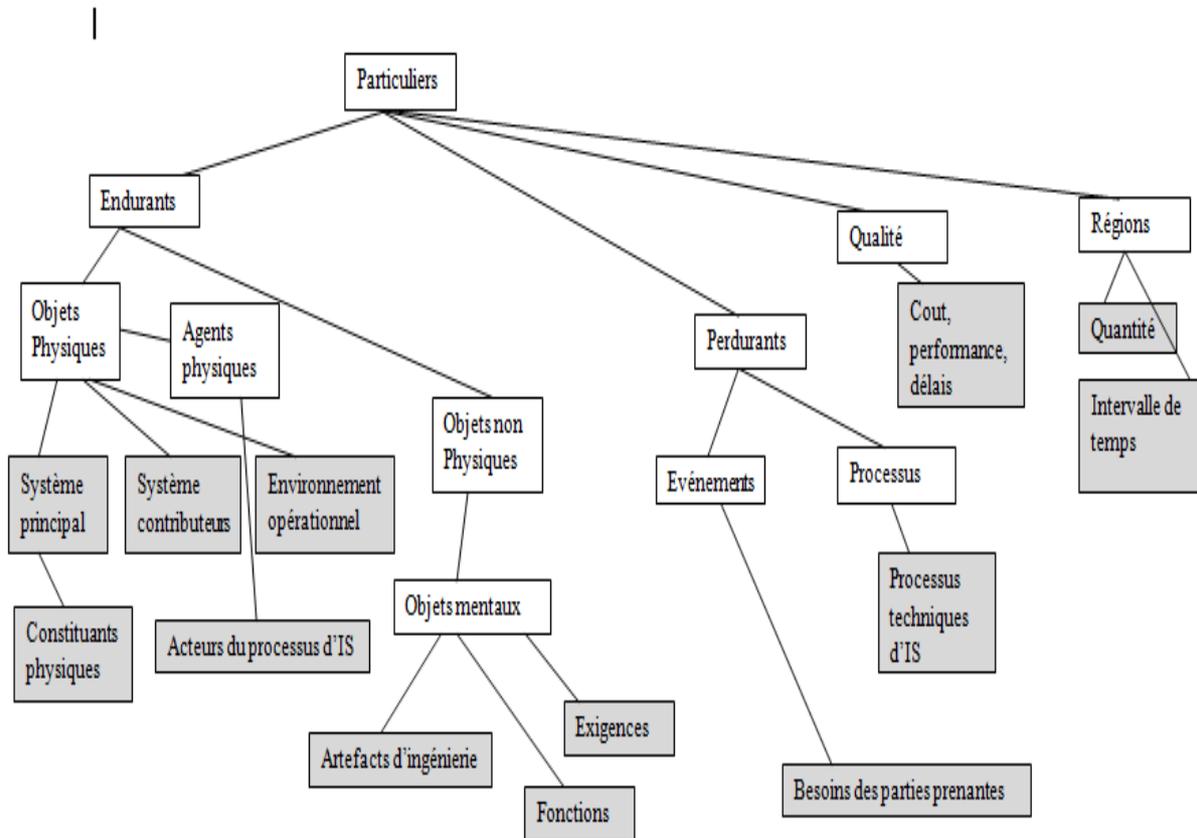


Figure IV.3 : Extension de l'ontologie DOLCE pour représenter les concepts de L'IS (concepts grisés)

Les relations sémantiques génériques liant les quatre sous domaines de l'ontologie DOLCE ont été formalisées dans le patron « Descriptions et Situations » [Gangemi et al., 2003]. Ce patron présente des descriptions de séquences d'événements qui ordonnent des entités temporelles (les Perdurants), des rôles que durants peuvent jouer pour répondre à ces événements, ainsi que des paramètres qui sont utilisés pour décrire rôles et événements, et prennent leurs valeurs dans des régions plus ou moins abstraites. Cette structure générique cherche à s'abstraire d'un domaine particulier pour décrire une situation du monde réel sous forme de primitives ontologique de DOLCE. Ce patron est adaptable à la conceptualisation d'un domaine particulier au travers d'une spécialisation des concepts et des relations qu'il représente. La figure IV.4 montre comment les concepts de D&S ont été spécialisés pour représenter une situation d'IS.

Les besoins des parties prenantes d'un système correspondent à une suite d'événement à prendre en considération lors de la mise en œuvre d'activité d'ingénierie comme la décomposition fonctionnelle d'un système. Les exigences et les fonctions systèmes sont allouées aux constituants d'un système : chaque constituant joue un rôle particulier pour satisfaire les besoins et les contraintes des parties prenantes. Et, dans la description de ces concepts, nous trouvons des paramètres comme le coût d'un projet ou les performances requises par un système, évalués par des unités de mesure.

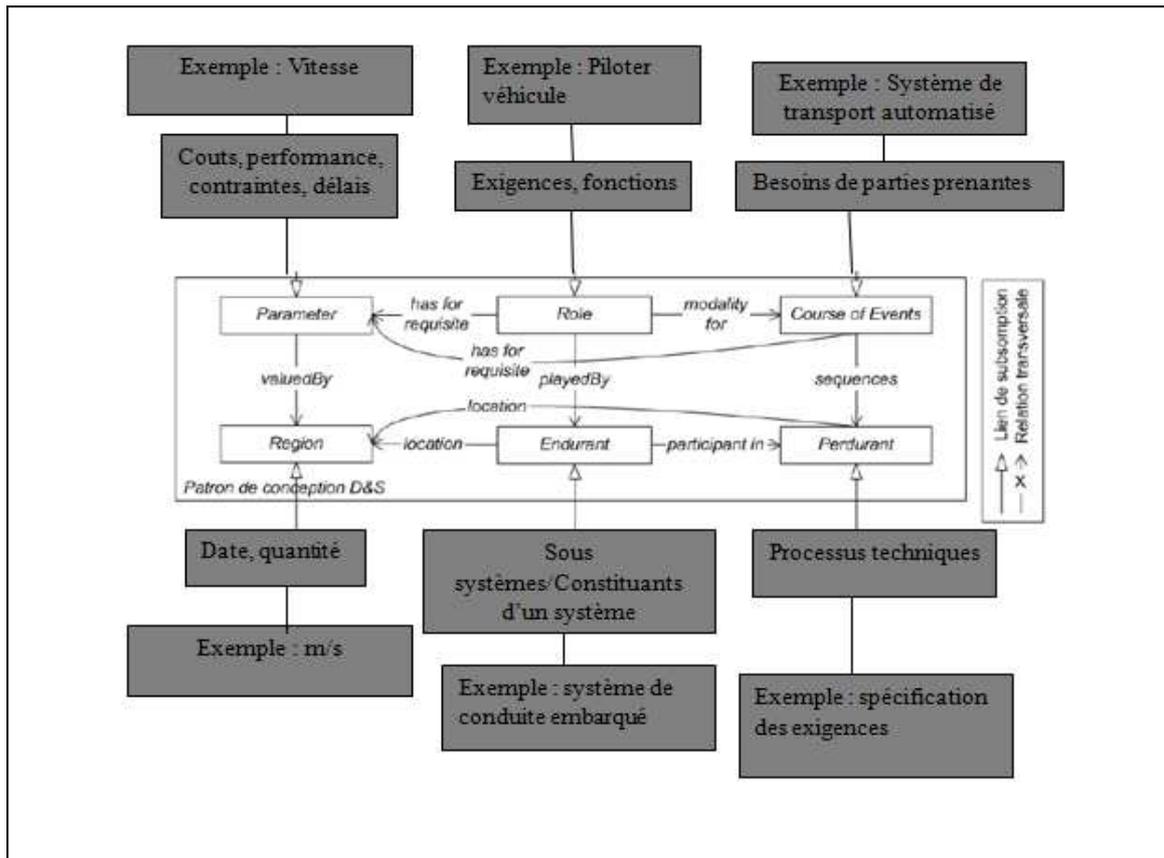


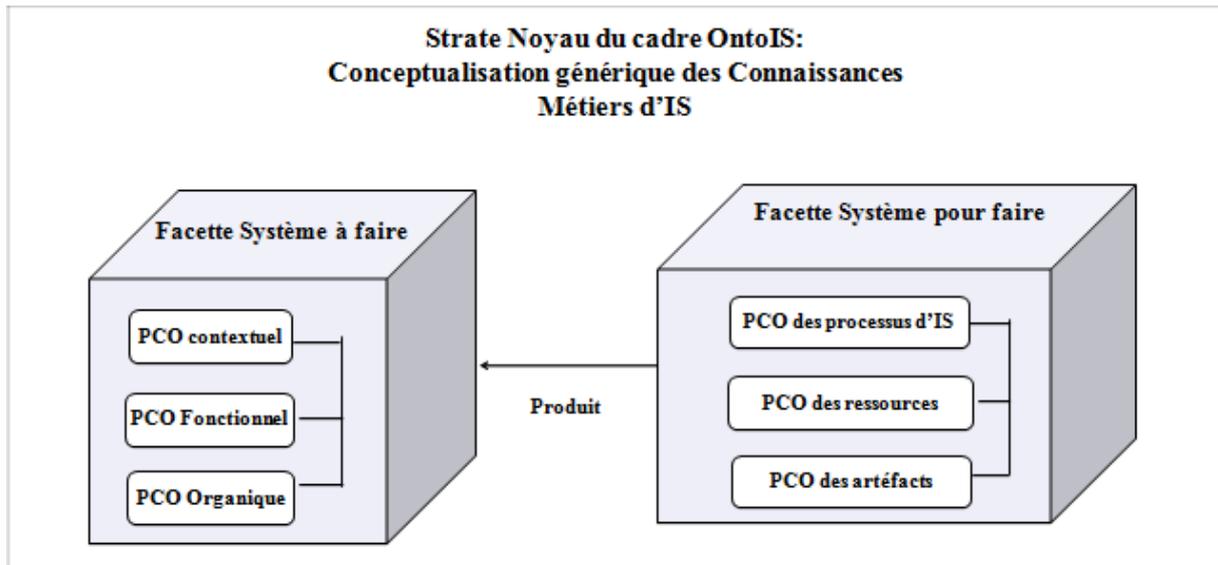
Figure IV.4 Spécialisation du patron D&S pour la représentation d'une situation d'Ingénierie Système

## IV.4. 2. Strate noyau du cadre OntoIS

L'alignement des concepts du métier de l'IS avec les concepts de haut niveau de l'ontologie DOLCE confirme le besoin de relier deux classes de connaissances métiers pour décrire les processus d'ingénierie : d'une part les connaissances liées à la dimension *système à faire* décrivant les besoins des parties prenantes (événements), les fonctions (objets mentaux) et les constituants (objets physiques) d'un système et d'autre part les connaissances liées à la dimension *système pour faire* représentant les processus d'ingénierie (endurants de type processus), les produits ou artefacts d'ingénierie (objets mentaux) et les ressources matérielles et humaines (agents physiques) intervenant dans un processus d'ingénierie.

Nous présentons dans cette section une modélisation de ces connaissances métiers sous la forme de Patrons Conceptuels Ontologiques (PCO). L'objectif est d'obtenir une modélisation couvrant les concepts génériques du métier de l'IS, dite noyau. Ces patrons peuvent être considérés comme le plus grand dénominateur conceptuel commun à toutes les ontologies relatives à un secteur d'activité ou à un métier d'IS.

La figure IV.5 présente l'organisation générale de la strate noyau du cadre OntoIS dédiée à conceptualisation des connaissances métiers d'IS.



IV.5 Strate noyau du cadre OntoIS

#### IV.4. 2. 1 Facette Système à faire : le Système étudié

Nous détaillons dans cette partie le contenu des patrons ontologiques destinées à modéliser les connaissances nécessaires pour décrire un système. Pour chaque patron nous présentons sa portée et les objectifs de modélisation, les intrants : représentant les ressources que nous avons utilisées pour conceptualiser les connaissances de domaine et le PCO proposé. Ces patrons qui sont au nombre de trois, sont les suivants :

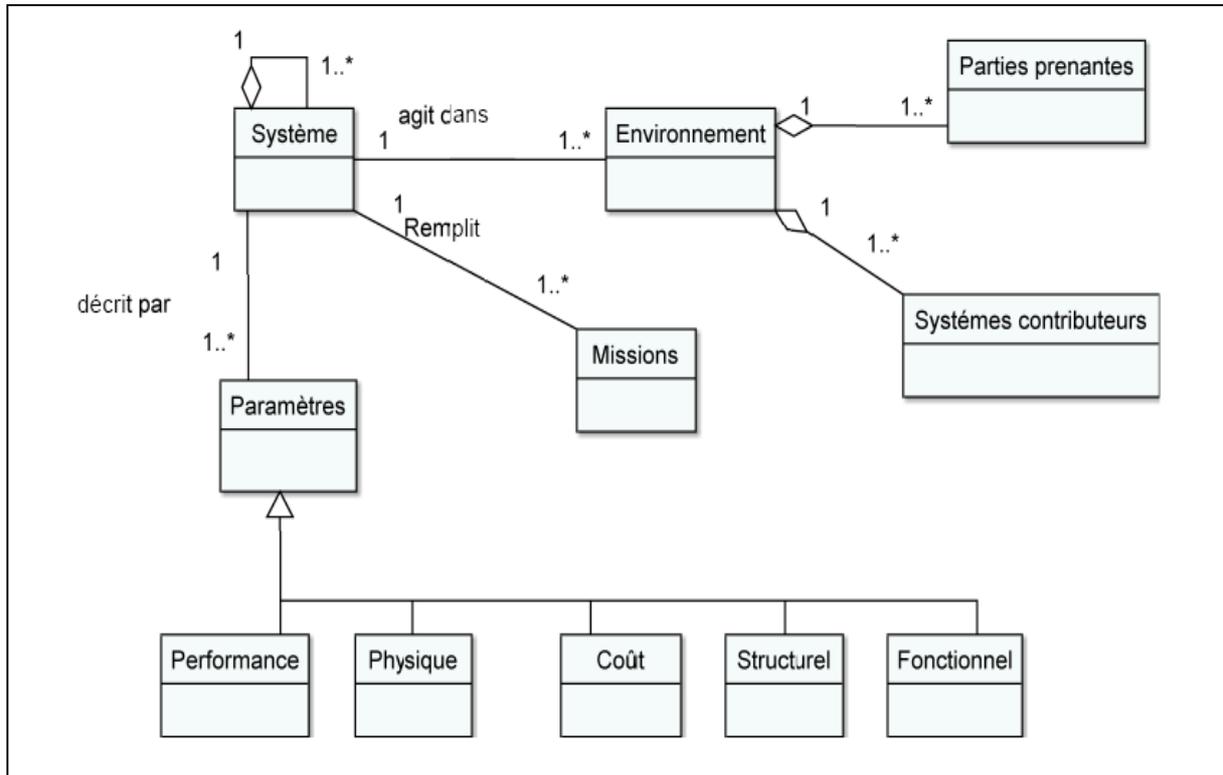
- Patron conceptuel ontologique (PCO) contextuel
- Patron conceptuel ontologique (PCO) fonctionnel
- Patron conceptuel ontologique (PCO) organique

##### IV.4. 2. 1.1 Patron conceptuel ontologique (PCO) contextuel

- **Portée et objectifs de modélisation :** expliciter les besoins des parties prenantes, la relation du système avec son environnement et les systèmes contributeurs. Les primitives conceptuelles du patron ontologique sont destinées à exprimer formellement les exigences d'un système.
- **Intrants :** réutilisation du fragment ontologique TOVE [Fox & Gruninger, 98], modèle de données [AFIS, 2007], [Meinadier, 2002]

Le PCO contextuel est schématisé par le diagramme suivant :

– PCO contextuel

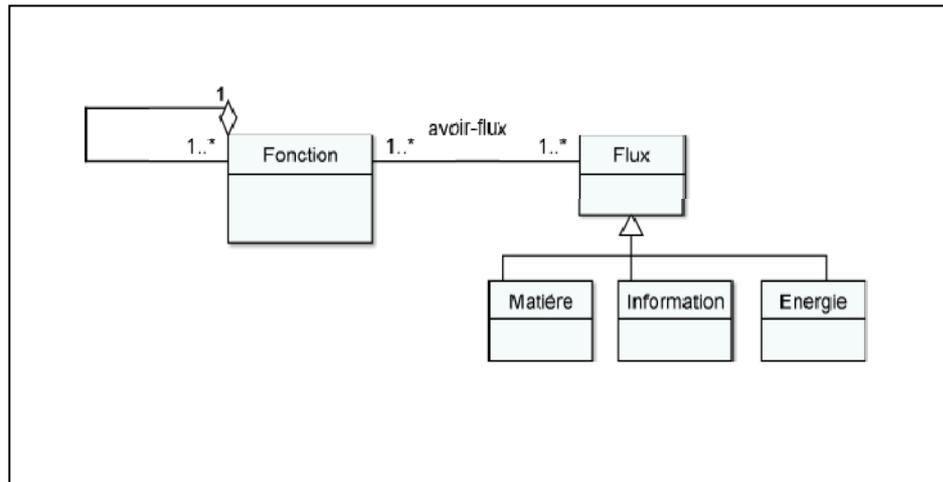


**IV.4. 2. 1.2 Patron conceptuel ontologique (PCO) fonctionnel**

- **Portée et objectifs de modélisation :** expliciter des fonctions de services et des fonctions internes d'un système sous la forme de catégories de fonctions systèmes et de catégories de flux d'entrées-sorties de ces fonctions. Les primitives conceptuelles (concepts et relations) de ce patron servent à formaliser le contenu de modèles fonctionnels d'un système exemple donner une sémantique partagée aux éléments d'un digramme de flux fonctionnels.
- **Intrants :** Ontologie fonctionnelle du projet NIST Design Repository Project [ Skyman, 2000], Cadre de systématisation des fonctions-système de [Mizoguchi, 2003]

Le PCO fonctionnel est schématisé par le diagramme suivant

– **PCO fonctionnel**

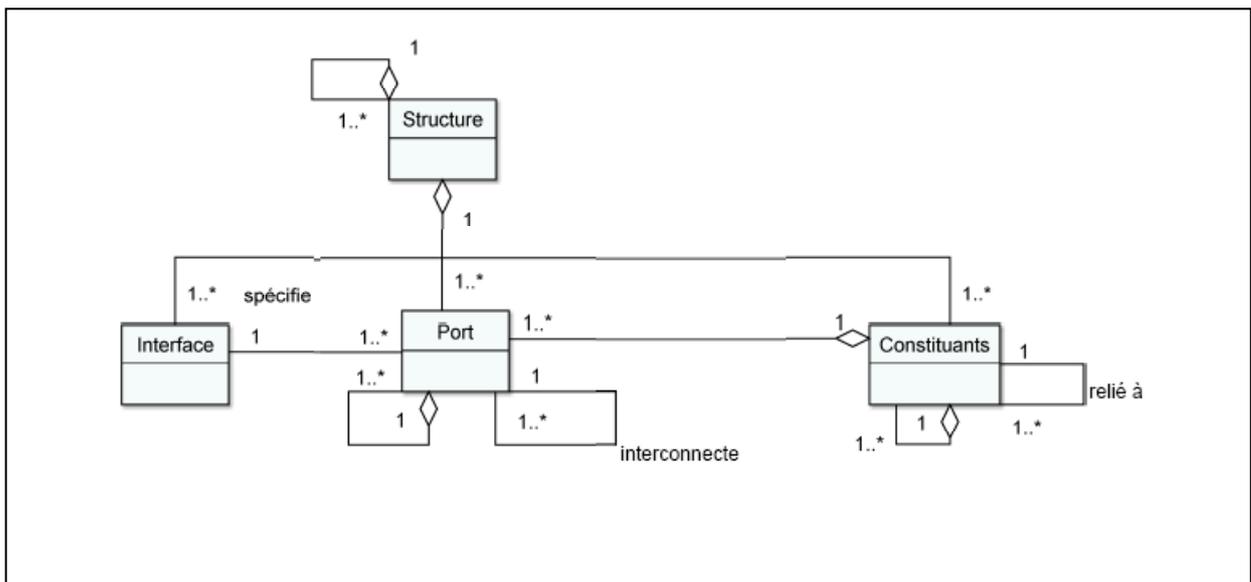


**V.4. 2. 1.3 Patron conceptuel ontologique (PCO) organique**

- **Portée et objectifs de modélisation :** expliciter les connaissances relatives aux composants matériels et aux composants logiciels d'un système. Les primitives ontologiques de ce patron servent à formaliser le contenu de modèles d'architecture physique d'un système.
- **Intrants :** fragment de l'ontologie Physis [ Borst et al. 1997], fragment de l'Ontologie proposée par [Chung, 2003]<sup>46</sup>

Le PCO organique est schématisé par le diagramme suivant

– **PCO organique**



<sup>46</sup> a port ontology for automated model composition [Chung, 2003]

## IV.4. 2. 2 Facette Système pour faire : le projet

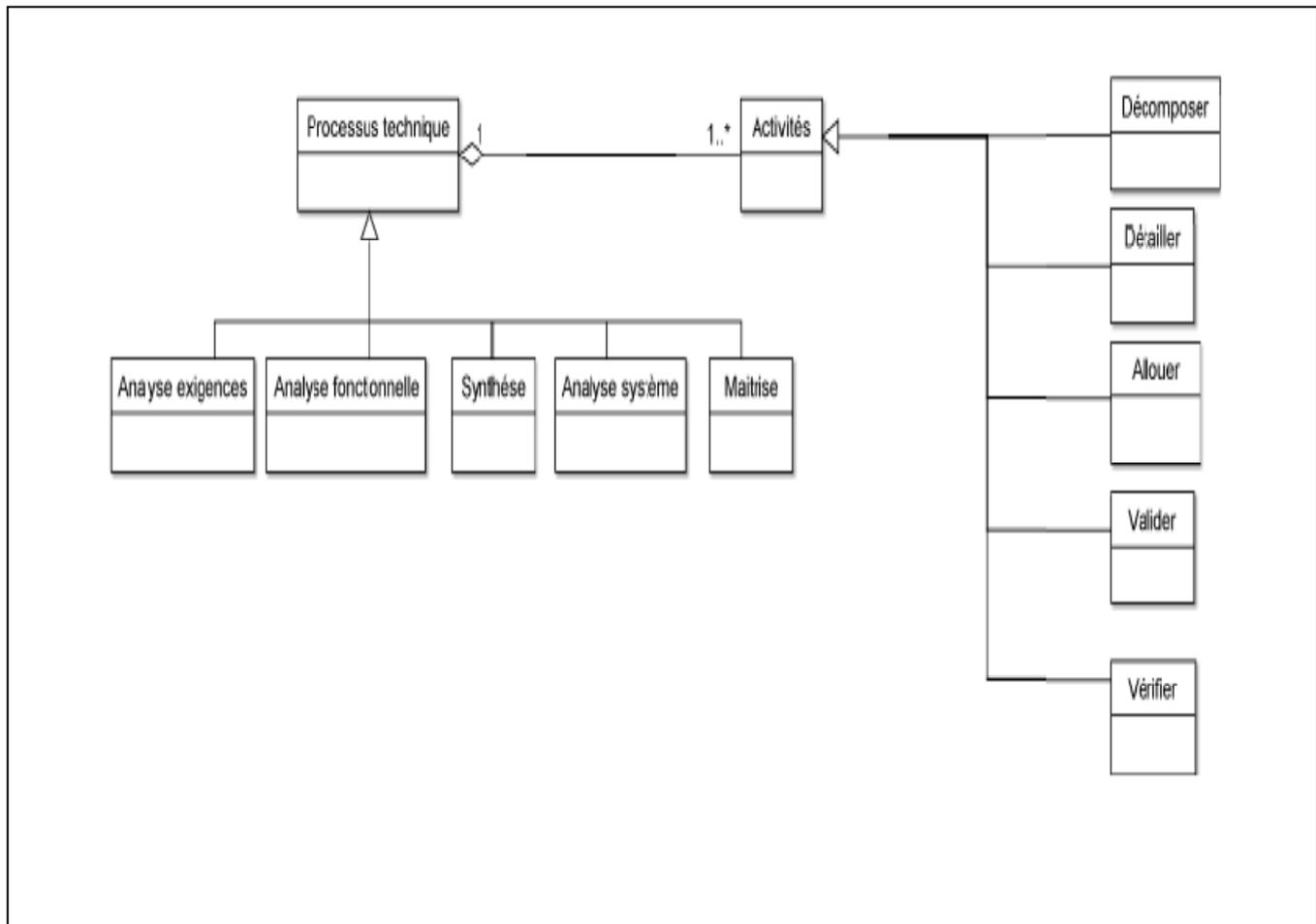
Les modules ontologiques de cette facette ont pour objectif de fournir une description consensuelle du contexte d'un projet d'ingénierie. Cette facette correspond aux ontologies de tâches (comme montré dans le chapitre III) servant à décrire la structure d'une tâche de résolution de problème de manière indépendante du domaine concerné.

### IV.4.2.2.1 Patron conceptuel ontologique (PCO) des processus d'IS :

- **Portée et objectifs de modélisation** : expliciter l'organisation générale d'un processus en termes de phases et d'activités d'ingénierie. les primitives ontologiques de ce patron servent à décrire le contexte de création des artefacts d'ingénierie et à prescrire à une granularité large<sup>47</sup> les tâches à suivre pour résoudre un problème d'ingénierie.
- **Intrants** : norme IEEE1220, fragment de l'ontologie d'IS [Sarder et al., 2007]

Le PCO processus est schématisé par le diagramme suivant

- **PCO processus**



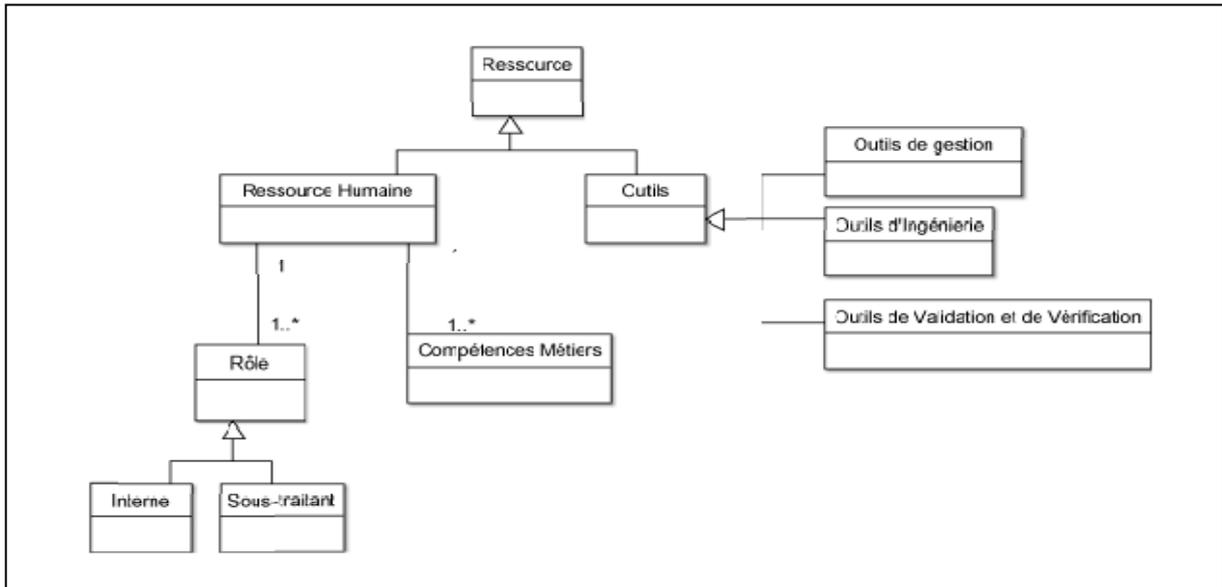
<sup>47</sup> Coarsegrained process models

#### IV.4.2.2 Patron conceptuel ontologique (PCO) des ressources

- **Portée et objectifs de modélisation** : expliciter la typologie des ressources supportant un processus d'IS. Ces ressources incluent les rôles, les profils métiers des ingénieurs systèmes et les outils supportant un processus d'ingénierie.
- **Intrants** : fragment de l'ontologie de modélisation des communautés de pratique O' COP [Mirbel et al., 2008], modèles de l'AFIS [AFIS, 2009b], fragment de l'ontologie d'IS [Sarder et al, 2007].

Le PCO ressources est schématisé par le diagramme suivant

- **PCO ressources**



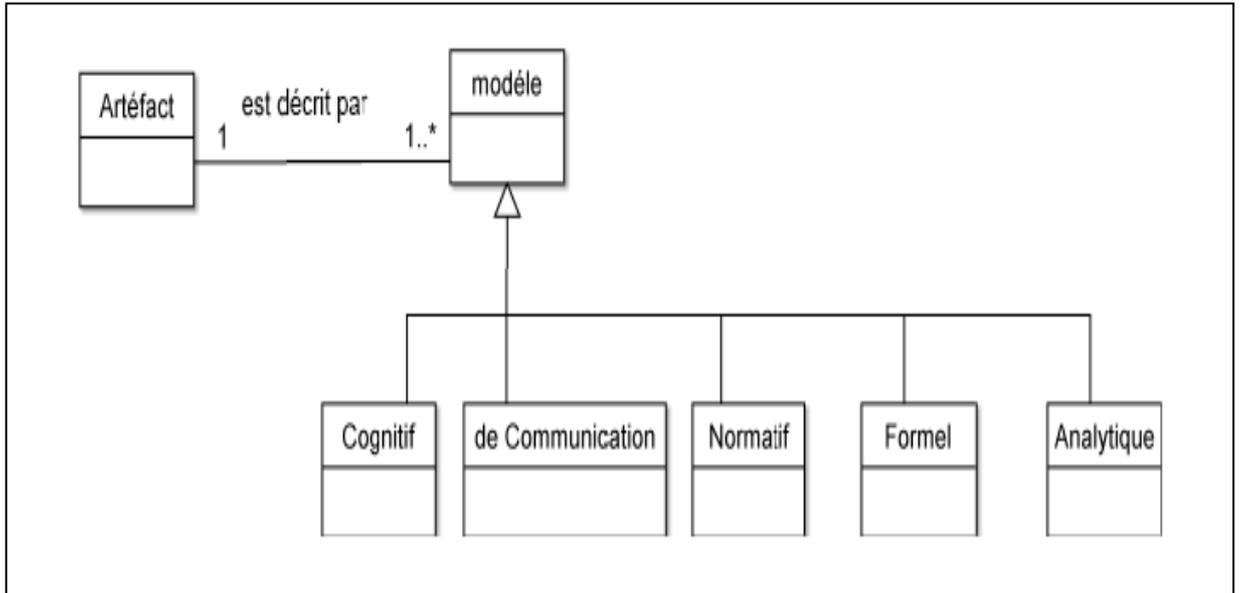
#### IV.4.2.2.3 Patron conceptuel ontologique des artefacts (PCO)

- **Portée et objectifs de modélisation** : expliciter la typologie des artefacts (ressources informationnelles) générées dans un processus d'IS. Les diagrammes SysML<sup>48</sup> constituent un exemple de spécialisation possible de modèles d'artéfact.
- **Intrants** : modèles de l'AFIS [AFIS, 2009b].

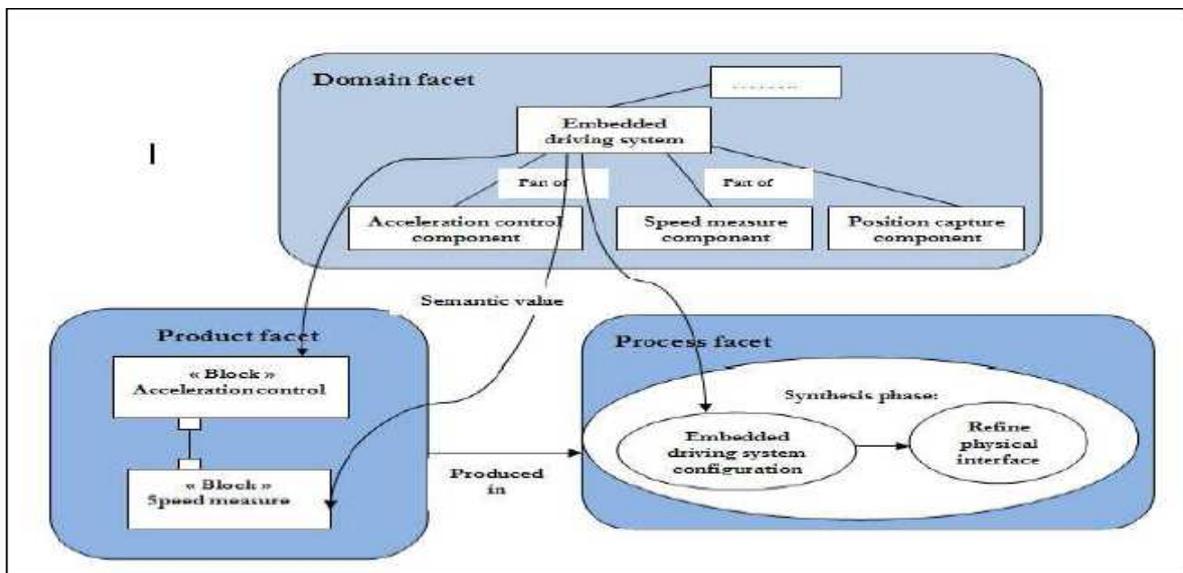
Le PCO Artéfact est schématisé par le diagramme suivant

<sup>48</sup> www.SysML.org

– PCO Artéfact



La figure IV.6 illustre les relations et la complémentarité de trois patrons conceptuels : organes, artéfacts et processus pour conceptualiser une situation d'IS.

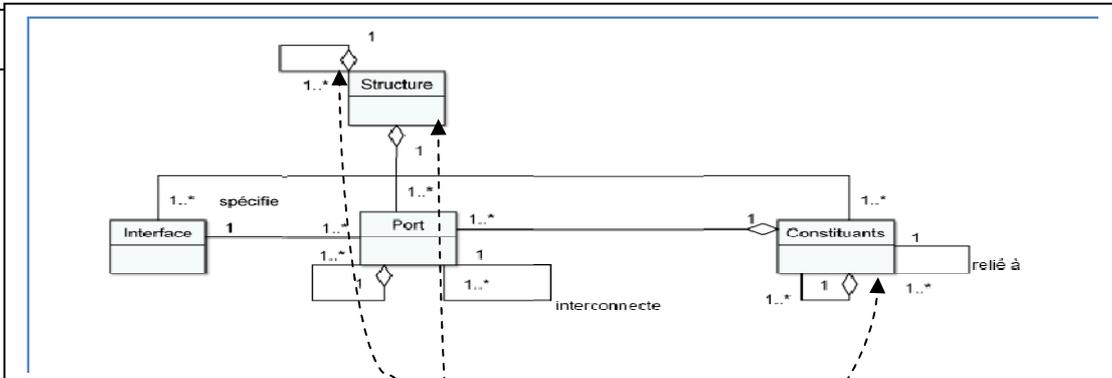


IV. 6 Relations entre les facettes système à faire et système pour faire [Chourabi et al, 2009]

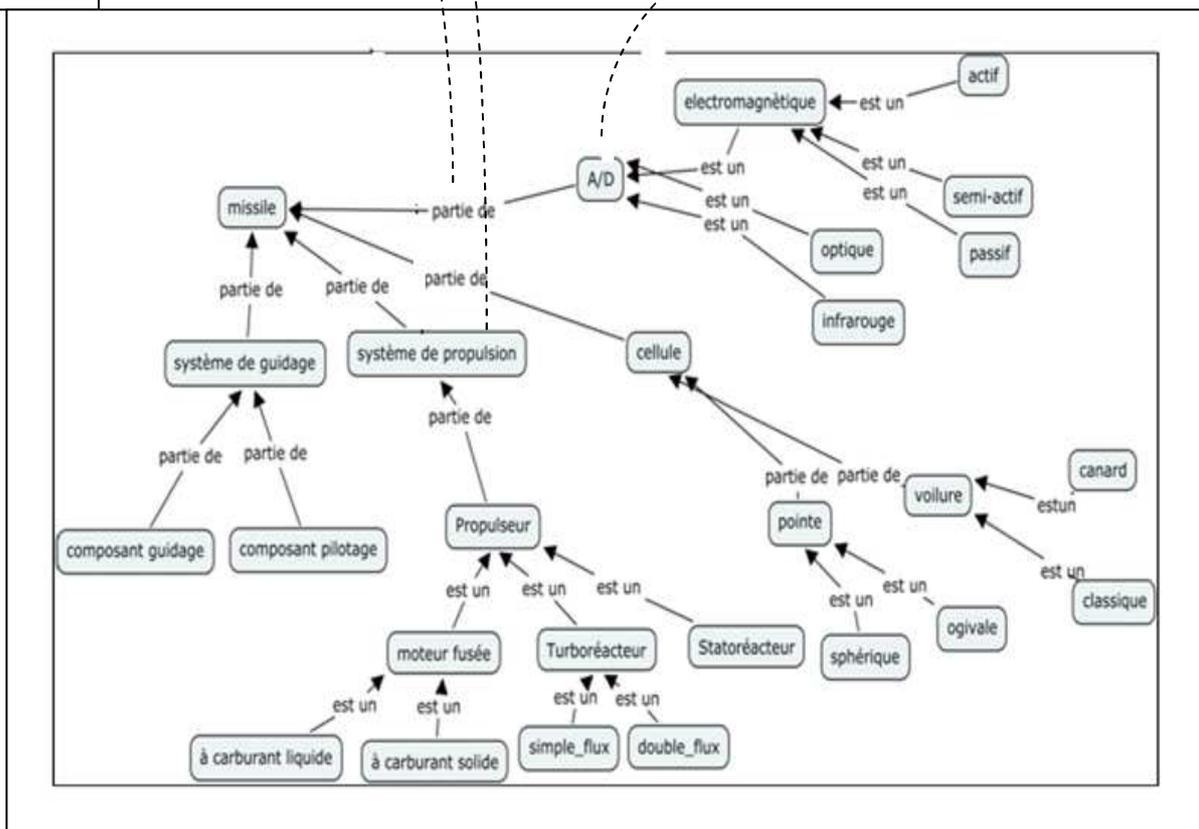
## IV.5. Exemple de spécialisation de Patrons Conceptuels Ontologiques (PCO) noyaux

Nous présentons dans cette section un exemple de spécialisation de patron ontologique conceptuel organique dans le secteur d'activité des systèmes de défense<sup>49</sup>.

Strate noyau



Strate application



<sup>49</sup> Cet exemple est tiré de l'étude de cas du chapitre VII de la présente thèse

## IV.6. Conclusion : Classes d'Applications potentielles du cadre OntoIS

Le cadre ontologique proposé est destiné à offrir une représentation consensuelle et unifiée des connaissances métiers d'IS. Il constitue une base générique permettant la définition de plusieurs classes de systèmes de gestion de connaissances. Nous analysons dans ce qui suit des applications potentielles de ce cadre dans un système de gestion de connaissances d'IS :

- Base de systématisation de connaissances de domaine d'IS : la spécialisation des patrons ontologiques de la facette système à faire permet de donner une vision consensuelle des concepts et des relations d'un domaine d'ingénierie. La navigation dans les composants ontologiques peut constituer d'une part un moyen d'apprentissage d'un domaine d'ingénierie, et d'autre part une base de spécification de nouveau système par réutilisation de connaissances métiers.
- Partage de solutions d'ingénierie basée sur le contenu : Les acteurs de projets d'IS sont en mesure de rechercher les solutions existantes en spécifiant les fonctions, les composants ou des sous-systèmes. La sémantique des solutions est formellement représentée par les ontologies ; le système de gestion de connaissance est donc apte à comprendre la sémantique d'une requête en se basant, par exemple, sur les relations de subsomption ou de composition.

*Exemple de partage de solution d'ingénierie basée sur le contenu pour un système de moteur d'avion* : nous considérons que la modélisation de ce système repose sur un fragment d'ontologie d'application (facette système à faire). L'association d'une connaissance formelle au modèle d'ingénierie de la figure VI.7 permet de le retrouver par une recherche sémantique. Le contenu de ce modèle est représenté par des instances des concepts « pompe de moteur d'avion », « moteur à réaction » et « pompe à carburant ». Un scénario de recherche constitué du concept « pompe hydraulique » permet de retrouver ce modèle en se basant sur la relation de subsomption entre « pompe hydraulique » et « pompe de moteur d'avion » (pompe de moteur d'avion « is a » pompe hydraulique).

- Navigation sémantique dans un référentiel de connaissances d'ingénierie : en explicitant les relations d'allocations de la facette système à faire un ingénieur système peut explorer les choix d'ingénierie capitalisés dans un référentiel de connaissances, par exemple : visualiser l'architecture organique correspondante à une architecture fonctionnelle donnée.
- Exploration du contexte de création des artefacts : À partir de la vue processus associée à chaque élément de connaissances capitalisé dans un référentiel de connaissances, un acteur de projet d'IS peut retrouver la trace du contexte de projet qui a conduit à choisir une solution donnée.

- Base de formalisation des expériences de projets : ce scénario concerne la représentation explicite et formelle relatives à des situations de résolution de problèmes d'ingénierie. Cette application est étudiée dans le chapitre V de cette thèse.

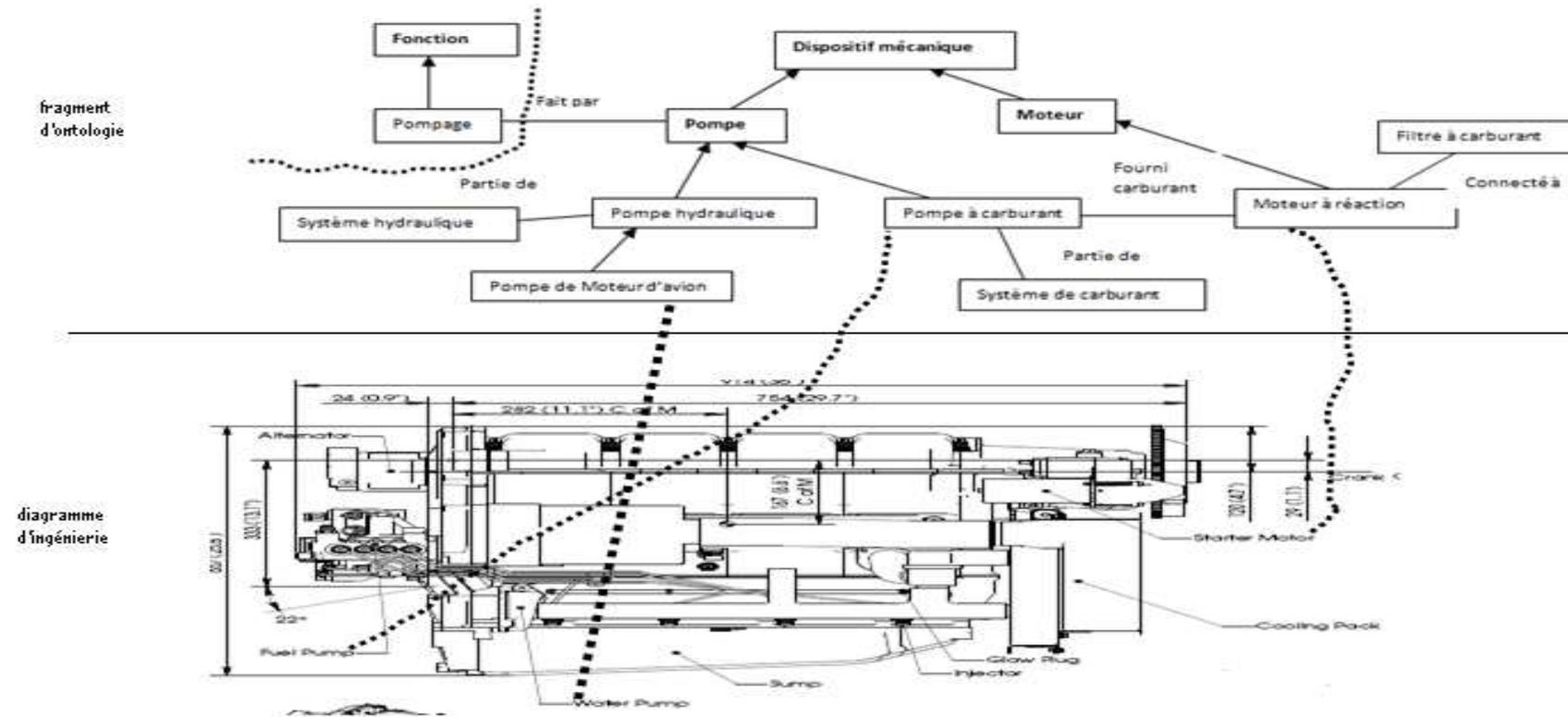


Figure IV. 7 Exemple de partage de solution d'ingénierie basée sur le contenu pour un système de moteur d'avion

# Chapitre V:

## Capitalisation et Partage De Connaissances Métiers Situées en IS

---

### Plan du chapitre

---

V.1. Introduction

V.2. Proposition d'un modèle de Connaissances Métiers Situées (CMS)

V.2.1. Position du problème

V.2.2 Exemple illustratif

V.2.3 Modèle conceptuel d'une Connaissances Métiers Située

V.3. Formalisation de la proposition

V.3.1. Modèles ontologiques

V.3.2. Modèle de capitalisation de CMS

V.3.3. Modèle de partage de CMS

V.4 Conclusion

## **V.1. Introduction**

Ce chapitre est dédié à la présentation détaillée d'une application possible du cadre Ontologique OntoIS : gestion des expériences qui émergent au cours des processus techniques d'IS. Ces processus sont perçus dans le cadre de nos travaux comme un espace de création continu de connaissances mettant en œuvre de multiples activités intellectuelles faisant passer progressivement d'un besoin d'ingénierie à la définition rigoureuse d'un système. La pratique actuelle montre que les ingénieurs systèmes cherchent à adapter, à s'inspirer ou à réutiliser des expériences de projets antérieurs (à différents niveaux de granularités parton, fragments de spécification, bonnes pratiques, architecture détaillées). Nous tentons de répondre à ces besoins en proposant un modèle de capitalisation et un modèle de partage d'expériences de projets d'IS.

## **V.2. Proposition d'un modèle de Connaissances Métiers Situées en IS**

### **V.2.1. Position du problème**

L'utilisation de techniques d'ingénierie simultanée et l'organisation des activités par projets sont désormais courantes dans les organisations d'IS [AFIS, 2007]. Dans ce contexte, des acteurs provenant de métiers différents et participant à des projets indépendants sont conduits à travailler ensemble. En situation de résolution de problèmes, l'investissement intellectuel des ingénieurs systèmes est considérable : proposition d'une solution d'ingénierie réalisant un compromis équilibré entre besoins et possibilités technologiques dans le contexte d'engagement de fonctionnalité, coût, délais, qualité [Meinadier, 2002].

En effet, au niveau cognitif le processus d'IS n'est pas entièrement planifiable, il s'agit d'un processus d'essai et erreur (en anglais, trial and error process) [Zadrahah, 2006]. Plutôt que de parcourir un ensemble d'activités séquentielles, le processus d'ingénierie est où les traits d'une solution sont reconnus, explorés, révisés et optimisés jusqu'à l'obtention d'une solution convenable (la plus satisfaisante) (sous-optimale et répondant aux exigences du système).

Dans ces situations de résolution de problèmes, les ingénieurs systèmes ne sont généralement pas motivés pour tracer leurs connaissances et ne souhaitent généralement pas être contraint par la documentation de leurs décisions et choix d'ingénierie. Ces connaissances décisionnelles constituent un capital intellectuel crucial : situé dans l'action i.e. dans des situations effectives de résolution de problèmes. ( figure V.1).

La valorisation de cette catégorie de connaissances dans des projets d'ingénierie ultérieurs pose les défis suivants : quelles sont les dimensions de modélisation des décisions de projets en IS ?, comment fournir aux ingénieurs Système un support en termes de connaissances potentiellement utiles dans une situation de projet ?

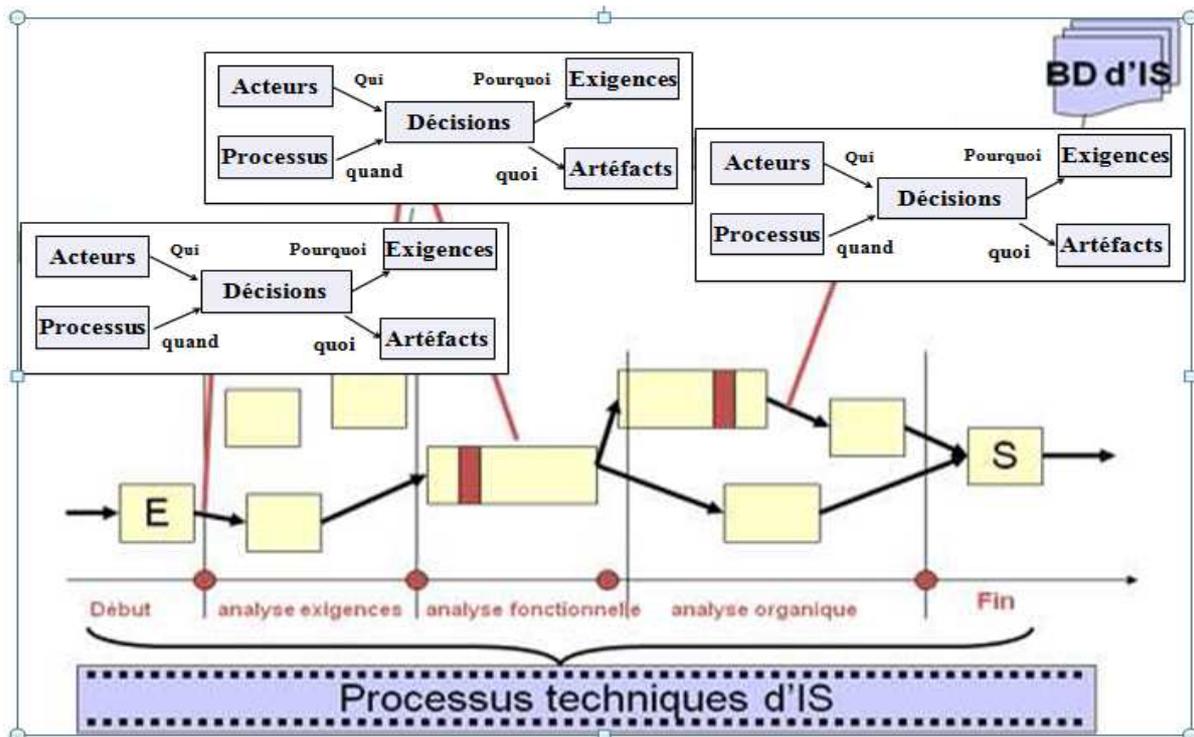


Figure V.1 : Dimension décisionnelle d'un processus d'Ingénierie

## V.2.2. Exemple illustratif : cas d'ingénierie d'un système de transport automatique

Prenons l'exemple d'un système de transport. Une analyse fonctionnelle montre qu'il faut prendre en compte les fonctions suivantes : capturer la vitesse, estimer la distance, commander le véhicule (à partir des vitesses et distances effectives, et de la loi de mouvement requise), propulser, freiner, contenir les voyageurs.

Cette vue fonctionnelle est déduite des exigences fonctionnelles du service à rendre (transporter des voyageurs d'un point à un autre). Il faut également prendre en compte les exigences non fonctionnelles comme par exemple : la performance (quantifiée par un temps de transport, le confort, décomposée en espace, qualité de l'assise, d'une part et accélération maximum d'autre part, la fiabilité du système, etc. L'architecture physique qui sera solution possible du problème d'ingénierie sera un ensemble d'organes choisis ou spécifiés : implémentant l'ensemble des fonctions, respectant les exigences non fonctionnelles, respectant des contraintes de coût et de délai imposées.

Les exigences non fonctionnelles imposées globalement se répercutent sur les exigences affectées aux composants. Cela peut être directement (l'espace alloué aux passagers et la qualité de l'assise s'affectent directement à l'organe devant assurer la fonction « contenir », c'est-à-dire l'habitacle), D'autres exigences se répartissent entre les constituants (le poids total est la somme des poids des composants), d'autres exigences portent sur l'ensemble et

devrons être vérifié par simulation (performance en temps de transport dans différents scenarii, fiabilité, etc.).

Chaque choix d'organe donne lieu à des alternatives d'ingénierie, avec différents choix possibles, et une décision conséquente qui doit prendre en compte non seulement les exigences spécifiques imposées à la fonction (ou aux fonctions) à assurer, mais aussi les exigences globales, que l'on ne peut vérifier globalement que sur la solution physique finalisée

Il existe de nombreuses interactions entre les choix. Par exemple un choix de constituant qui en exclut un autre (par exemple on voit que l'on peut implémenter les captures de vitesse et de position de nombreuses manières. Si l'on prend une centrale à inertie qui délivre à la fois une vitesse et une position, cela exclu des capteurs spécifiques). Des interactions au niveau d'un paramètre commun. Par exemple les performances en temps dépendent du poids total du véhicule, lui-même somme des poids des différents organes. On voit bien donc que la performance de temps ne peut se reporter seulement sur l'organe qui assurera la propulsion. Par ailleurs, dans les exigences, il y a des exigences « dures » (elles sont imposées) et des exigences plus ou moins flexibles (avec une valeur à atteindre si possible, mais d'autres valeurs admissibles, mais dotée d'une préférence moindre).

Dans la pratique de l'ingénierie d'un système réel un tant soit peu complexe, il y aurait ainsi de nombreuses hypothèses à gérer, qui correspondraient aux choix que l'on ne peut opérer que globalement sur un sous-ensemble (voire sur l'ensemble du système) de constituants.

### **V.2.3. Modèle conceptuel de CMS**

Nous introduisons le concept de Connaissances Métiers Situées (CMS) pour représenter les connaissances implicites liées à la dimension décisionnelle d'un processus d'IS comme illustré dans l'exemple du système de transport automatique.

Pour ce faire, il est nécessaire de clarifier les dimensions de modélisation des décisions de projets en IS.

Dans notre étude de l'état de l'art (Chapitre II section II.4) nous avons analysé les travaux liés aux systèmes de gestion d'expériences, en particulier les mémoires de projets [Matta, 2002] et leurs applications dans le contexte de processus de conception. Ces approches préconisent de modéliser toutes les dimensions décrivant le contexte (i.e. organisationnel, méthodologique, technique.. comme mentionné dans la section II.4.3) de projet et la logique de conception (i.e. les choix et les alternatives de solutions) sous la forme d'un unique schéma conceptuel. Les connaissances de projet sont ensuite mémorisées en instanciant ce schéma. En nous positionnant par rapport à cette approche dans un contexte d'Ingénierie Système, nous trouvons qu'elle n'offre pas le niveau de granularité requis pour modéliser les décisions de projets en IS, i.e. ces décisions ne sont pas dépendantes de tout le contexte d'un projet. Les exigences globales d'un système sont décomposées et ensuite allouées (récursivement) aux divers sous- systèmes à ingénieriser. Pour offrir une modélisation plus adaptée aux besoins de l'IS nous introduisons le concept de CMS comme structure capturant à la fois une décision et sa justification. Par ailleurs, nous pensons qu'il est irréaliste de tout mémoriser et qu'il faut

laisser à l'ingénieur système la possibilité de choisir les décisions les plus pertinentes à mémoriser.

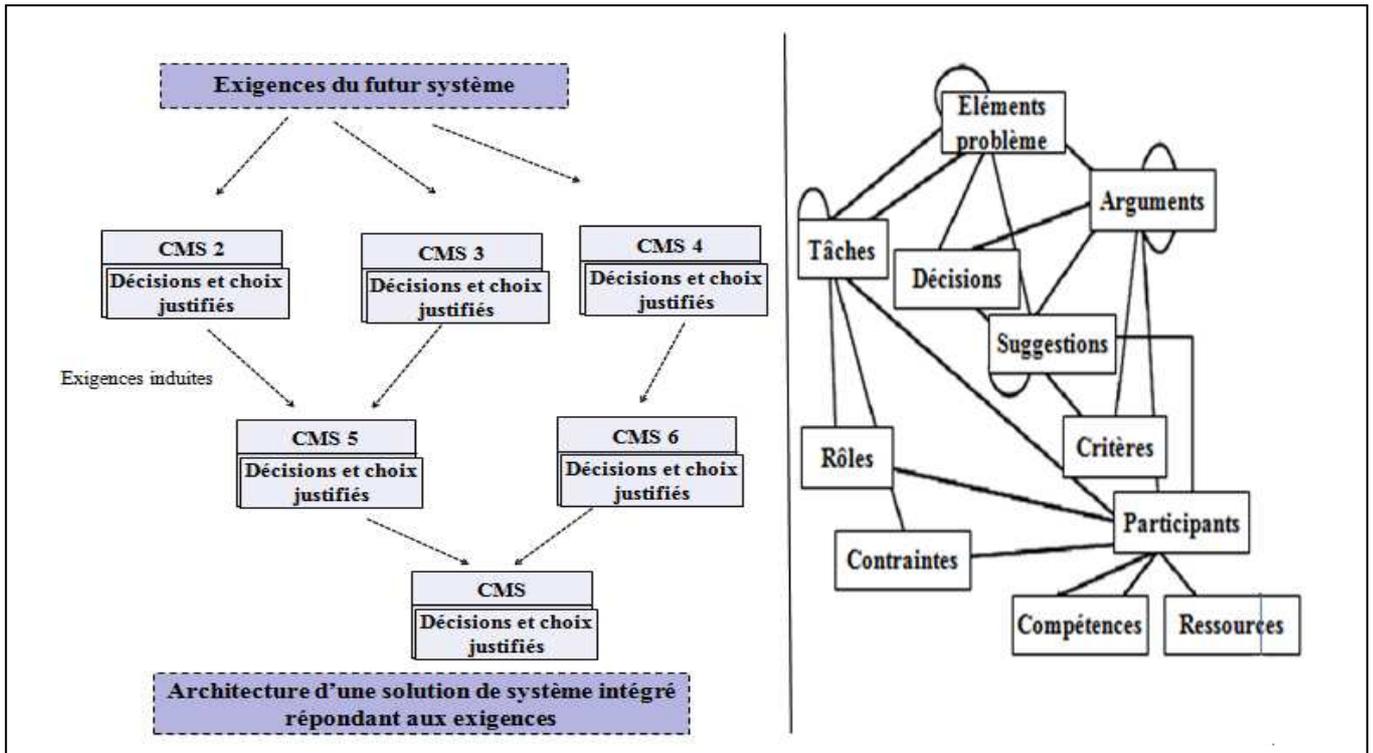


Figure V.2 : CMS vs. Mémoires de projets

Le modèle conceptuel de CMS proposé a pour objectif d'offrir une représentation plus adaptée au métier d'IS, sous la forme d'un modèle multidimensionnel basé sur le cadre ontologique OntoIS. Quatre dimensions sont proposées :

- La dimension situationnelle: spécifie le contexte de création de l'expérience : phase/activité, intrants, contraintes de projet.
- La dimension Téléologique : spécifie le but de production à développer : raffiner, détailler, allouer ou réifier, fonction, sous-système, constituant
- La dimension argumentative: décrit les alternatives de solutions considérées dans une situation de projet
- La dimension décisionnelle : décrit la solution adoptée pour résoudre un problème d'ingénierie



Figure V.3: Modèle conceptuel d'une CMS

### V.3. Formalisation de la proposition

Cette section est consacrée à la formalisation du modèle de CMS proposé. Cette formalisation est constituée de trois modèles :

- **Un modèle d'ontologie** : pour bénéficier des plines possibilités d'une approche ontologique, nous avons retenu un modèle de description basé un modèle d'ontologie lourde<sup>50</sup> permettant d'offrir un niveau expressivité satisfaisant et de bonnes capacités inférentielles.
- **Un modèle de capitalisation de CMS** : constitué d'une structure multidimensionnelle dont chaque dimension est une expression formelle spécifique basée sur le modèle de l'ontologie.
- **Un modèle de partage de CMS**: fondé sur une approche logique d'appariement d'une situation d'ingénierie et d'une base de CMSs conforme au modèle de capitalisation.

Nous nous plaçons ainsi dans un cadre formel de gestion de connaissances représentant une classe d'application de L'IO : l'annotation /recherche à base d'ontologie [Euzenat, 2005] par opposition aux approches textuelles utilisant des comparaisons entre les contenus d'une ressource informative et de requêtes, aux approches numériques – telles que les calculs probabilistes de similarité entre descriptions et requêtes ou aux approches thésaurales qui ne reposent ni sur une interprétation ni sur des calculs formels et qui sont en en définitive, des approches terminologiques [Isaac, 2005].

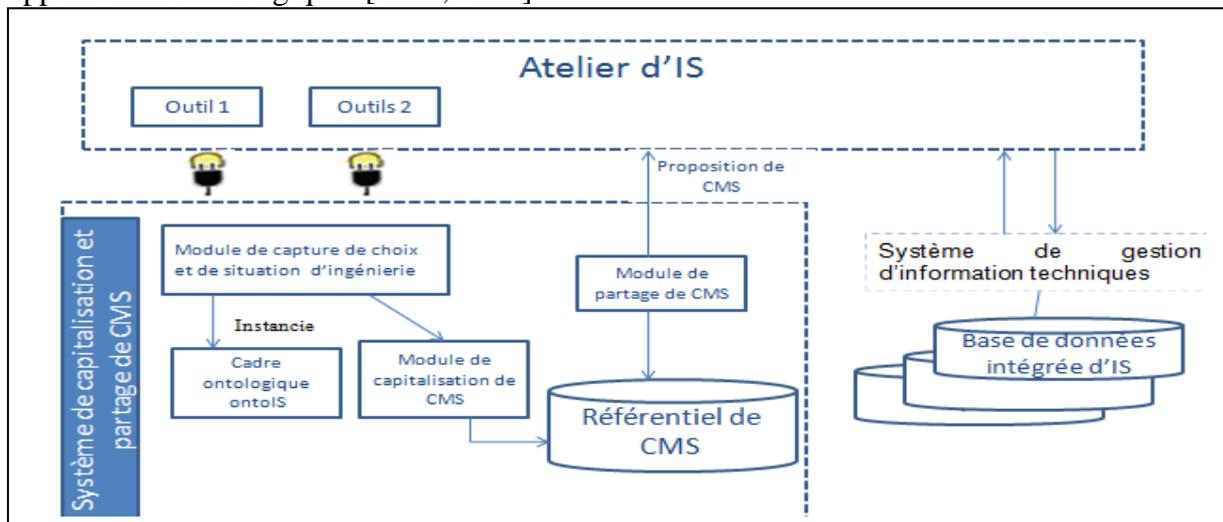


Figure V.4 Vue d'ensemble de la proposition de gestion de Connaissances Métiers Situées

<sup>50</sup> Des ontologies intégrant l'ensemble des axiomes permettant de fixer toute la sémantique du domaine considéré, en mettant en exergue les connaissances inférentielles, en comparaison aux ontologies dites légères qui elles n'incluent pas d'axiomes et sont uniquement fondées sur des hiérarchies de concepts et de relations

### V.3.1. Modèle d'ontologie

Dans cette section nous proposons une formalisation des modèles d'ontologies composant le cadre OntoIS. Pour définir le modèle d'ontologie nous nous sommes principalement inspirés du modèle PLIB [G.Pierra et al, 2003] et du modèle ontologique de Karlsruhe [G.Stumme et al, 2003] que nous étendons et adaptant aux besoins de formalisation de connaissances métiers d'IS. Nous visons la proposition d'un modèle d'ontologie abstrait pouvant être ensuite réifié selon les possibilités offertes par les langages d'opérationnalisation des ontologies (comme les Logiques de Description, les Graphes Conceptuels ou les langages, les langages et outils du web sémantique). Ce modèle d'ontologie doit être suffisamment riche en primitives conceptuelles parce qu'il constitue le fondement sur lequel nous définissons les modèles de capitalisation et de partage de CMS.

Le modèle d'ontologie proposé est décliné en deux niveaux : Un niveau terminologique, Un niveau axiomatique<sup>51</sup>

**Définition 1 : (Modèle d'ontologie)**

Formellement un modèle d'ontologie est défini par un couple  $(\mathbf{T}, \mathbf{Ax})$  où  $\mathbf{T}$  représente le niveau terminologique,  $\mathbf{Ax}$  représente le niveau axiomatique.

#### V.3.1.1 Le niveau terminologique

**Définition 2 : (Niveau Terminologique)** composé d'un ensemble de primitives conceptuelles permettant de décrire les connaissances métiers d'IS.

Le niveau terminologique est défini par un n-uplet

$\mathbf{T} := (\mathbf{C}, \leq_{\mathbf{C}}, \mathbf{R}, \leq_{\mathbf{R}}, \mathbf{sign}, \mathbf{P}, \leq_{\mathbf{P}}, \mathbf{TD}, \mathbf{UM}, \mathbf{Papplic}, \mathbf{TDapplic}, \mathbf{UMapplic})$  où

- $\mathbf{C}$  et  $\mathbf{R}$  sont deux ensembles disjoints dont les éléments sont respectivement appelés *Concepts* et *Relations*.
- $\leq_{\mathbf{C}}$  un ordre partiel sur  $\mathbf{C}$  ( $\mathbf{C} \rightarrow 2^{\mathbf{C}}$ ), appelé hiérarchie de concepts ou relation de subsomption de concepts ;
- $\leq_{\mathbf{R}}$  un ordre partiel sur  $\mathbf{R}$ , appelé hiérarchie de relations ou relation de subsomption de relations
- La fonction **Sign** définit des relations sémantiques entre concepts :  

$$\mathbf{Sign} : \mathbf{R} \rightarrow \mathbf{C} \times \mathbf{C}..$$
- $\mathbf{P}$  l'ensemble des paramètres<sup>52</sup> décrivant les concepts ontologiques  $\mathbf{C}$ .

---

<sup>51</sup> Il existe plusieurs définitions de ce niveau, dans notre modèle, nous considérons les propriétés algébriques de relations et les règles de domaine. Les axiomes terminologiques sont inclus dans le niveau terminologique

<sup>52</sup> Dénommés aussi dans la littérature (attribut/propriétés)

- $\leq P$  un ordre partiel sur  $P$ , appelé hiérarchie de relations ou relation de subsumption de paramètres (l'organisation en treillis de l'ensemble des paramètres constitue un choix fondamental sur lequel nous basons notre modèle de partage de CMS. Ce point est détaillé dans la section V. 3.3)
- $TD$  : l'ensemble des types de données décrivant les paramètres
- $UE$  : l'ensemble des unités de mesures décrivant les paramètres
- La fonction  $PAPPLIC: C \rightarrow 2^P$  qui associe à tout concept les paramètres qui lui sont applicables.
- La fonction  $TDAPPLIC: P \rightarrow 2^{TD}$  qui associe à tout paramètre les types de données qui lui sont applicables.
- La fonction  $UMAPPLIC: P \rightarrow 2^{UE}$  qui associe à tout paramètre les unités de mesure qui lui sont applicables.

Considérons par exemple un cas simplifié d'Ingénierie d'un système aéronautique pour illustrer les primitives conceptuelles du niveau terminologique.

- $C = \{\text{aéronef, avion, propulseur, moteur à réaction, système de carburant, pompe à carburant}\}$
- $\leq c = \{\text{avion} \sqsubseteq \text{aéronef, moteur à réaction} \sqsubseteq \text{propulseur, pompe à carburant} \sqsubseteq \text{système de carburant}\}$
- $R = \{\text{partie-de, connecté-à, fournit-carburant-à}\}$
- $\leq R = \{\text{fournit-carburant-à} \sqsubseteq \text{connecté-à}\}$
- $Sign = \{\text{partie-de (propulseur, avion), partie-de (système de carburant, avion), connecté-à (système de carburant, propulseur), fournit-carburant-à (pompe à carburant, moteur à réaction)}\}$
- $P = \{\text{caractéristiques-physiques, poids, portée}\}$
- $\leq P = \{\text{poids} \sqsubseteq \text{caractéristiques-physiques}\}$
- $TD = \{\text{réel}\}$
- $UM = \{\text{Kg, Km}\}$
- $Papplic = \{(\text{poids, avion}), (\text{portée, avion})\}$
- $TDapplic = \{(\text{poids, réel}), (\text{portée, réel})\}$
- $UMapplic = \{(\text{poids, kg}), (\text{portée, km})\}$

Cette formalisation ontologique indépendante du modèle peut être traduite en plusieurs langages d'opérationnalisation d'ontologies. Nous présentons des exemples de traduction du modèle proposé dans le langage OWL.

**R** → ObjectProperty

**P** → DatatypeProperty,

**≤R, ≤P** → rdfs:subPropertyOf,

**Sign** → rdfs:domain, rdfs:range

Les concepts qui apparaissent au le niveau terminologique **T** peuvent être organisés selon les deux catégories suivantes [Gruber, 1995] :

- les concepts **primitifs** : Les concepts primitifs sont la fondation sur laquelle d'autres concepts de l'ontologie pourront ensuite être définis.
- les concepts **définis** sont les concepts « pour lesquels une ontologie fournit une définition axiomatique complète au moyen de conditions nécessaires et suffisantes exprimées en termes d'autres concepts primitifs ou eux-mêmes définis ».

L'introduction de concepts définis amène à une notion d'équivalence conceptuelle qui permet d'exprimer de plusieurs manières différentes les mêmes concepts.

C'est le modèle d'ontologie ou le langage associé qui définit les différents constructeurs permettant de composer les concepts primitifs et/ou définis pour construire d'autres concepts définis. Les constructeurs de concepts définis de base sont l'Union, l'Intersection et la Restriction. Ces constructeurs peuvent s'appliquer aux primitives ontologiques (concepts, relations, paramètres).

Par exemple, avec un constructeur de Restriction sur les paramètres d'un concept primitif nous pouvons définir un concept définissant l'exigence de poids d'un aéronef comme :

**Exigence\_poids\_Aéronef**  $\equiv$  aéronef  $\cap$  (poids < 3300 kg).

Nous notons **T\*** le niveau terminologique constitué de **T** augmenté des concepts définis moyennant des constructeurs de primitives ontologiques.

### V.3.1.2 Le niveau Axiomatique

Le niveau axiomatique permet d'enrichir la sémantique des primitives conceptuelles spécifiées au niveau terminologique. Il fournit un ensemble de propriétés étendant l'ontologie pour augmenter son expressivité.

Selon [Furst, 2004] ce niveau se décompose en deux parties :

- Un ensemble de schémas d'axiome portant sur les primitives conceptuelles et exprimant, au moins partiellement, la sémantique de ces primitives dans le domaine ; ces axiomes se déclinent en axiomes portant sur les concepts (par exemple : le lien sorte-de, la disjonction de concepts), des axiomes portant sur les relations (par exemple : la propriété de transitivité d'une relation), des axiomes d'instances (le type d'une instance est le concept le plus spécifique auquel elle appartient)
- Un ensemble d'axiomes de domaine complétant si besoin l'expression de la sémantique des primitives du domaine.

Dans notre modèle, nous nous focalisons principalement sur les propriétés algébriques portant sur les relations pour définir les schémas d'axiome, car les axiomes portant sur les concepts ont été formalisés au niveau terminologique T. En particulier, nous considérons les propriétés de symétrie, de transitivité, de réflexivité, d'anti-réflexivité et d'anti-symétrie d'une relation binaire.

Par exemple, la relation « partie-de » liant des constituants d'un système aéronautique est :

**Transitive:**  $(\forall p1, p2, p3), \text{partie-de}(p1, p2) \wedge \text{partie-de}(p2, p3) \rightarrow \text{partie-de}(p1, p3).$

**Non-réflexive :**  $(\forall p) , \neg \text{partie-de}(p, p)$

**Anti-symétrique :**  $(\forall p1, p2), \text{partie-de}(p1, p2) \wedge \text{partie-de}(p2, p1) \rightarrow p1 = p2$

La deuxième catégorie d'axiomes est totalement spécifique au domaine considéré et, contrairement aux schémas d'Axiomes, ne correspondent pas à des propriétés classiques attestées sur les relations. Ces axiomes se matérialisent généralement sous la forme de règle logique décrivant des inférences possibles dans un domaine de connaissances. Nous considérons particulièrement dans cette catégorie les contraintes, de type nécessité et/ou incompatibilité, pouvant être exprimées sur les fonctions et/ou les constituants d'un système. Ces contraintes restreignent les combinaisons possibles des primitives conceptuelles du niveau terminologique.

- Exemple de contrainte exprimant une nécessité « *require* » pour un système aéronautique : la présence du constituant moteur à réaction dans une architecture organique implique la présence d'un constituant pompe à carburant.

Ces axiomes peuvent être utilisés soit pour inférer de nouvelles connaissances soit pour valider l'adéquation d'une connaissance par rapport au domaine considéré. [Furst, 2004]

Un modèle d'ontologie permet de définir aussi un niveau assertionnel. Ce niveau décrit les connaissances factuelles d'un domaine et représente une configuration précise [Temal, 2008]

**Définition 3 :** (*niveau assertionnel*) permet l'expression des extensions (ou instances, ou encore individus) des primitives conceptuelles du niveau terminologique

**ASS = (T\*, I, V, fC, fTd, fUM, fR, fp)** où :

- **T\*** est le niveau terminologique **T := (C, ≤C, R, ≤R, Sign, P, ≤P, TD, UM, P<sub>applic</sub>, TD<sub>applic</sub>, UM<sub>applic</sub>)** augmenté des concepts définis :
- **I** est l'ensemble des extensions de concepts
- **V** est l'ensemble des valeurs des types de données **TD**
- **fC** : **I** → **C** est la fonction d'instanciation de concepts
- **fTD** : **V** → **TD** est la fonction d'instanciation de type de données
- **fUM** : **V** → **UM** est la fonction d'instanciation d'unités de mesure
- **fR** : **I** × **I** → **R** est la fonction d'instanciation de relations
- **fp** : **I** × **V** → **P** est la fonction d'instanciation de paramètres

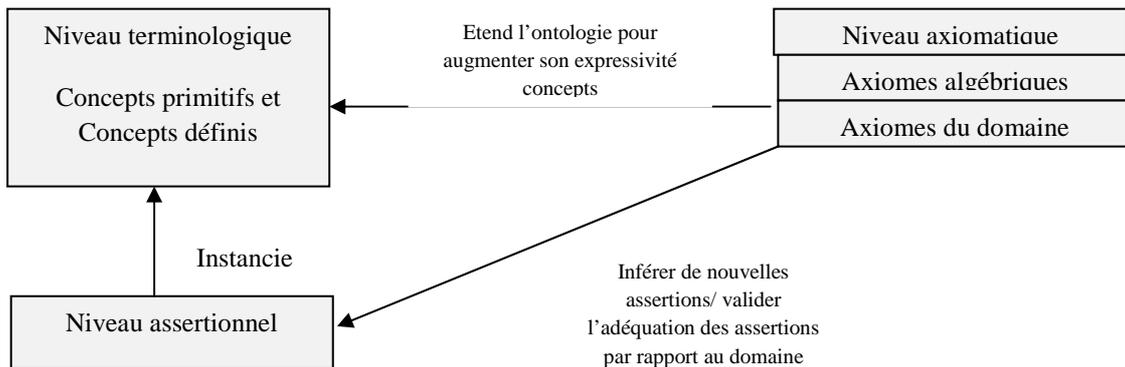


Figure V.6 Relations entre niveaux terminologique, assertionnel, et axiomatique

La figure V.7 présente un exemple d'assertion instanciant le niveau terminologique de la section V.3.1.1

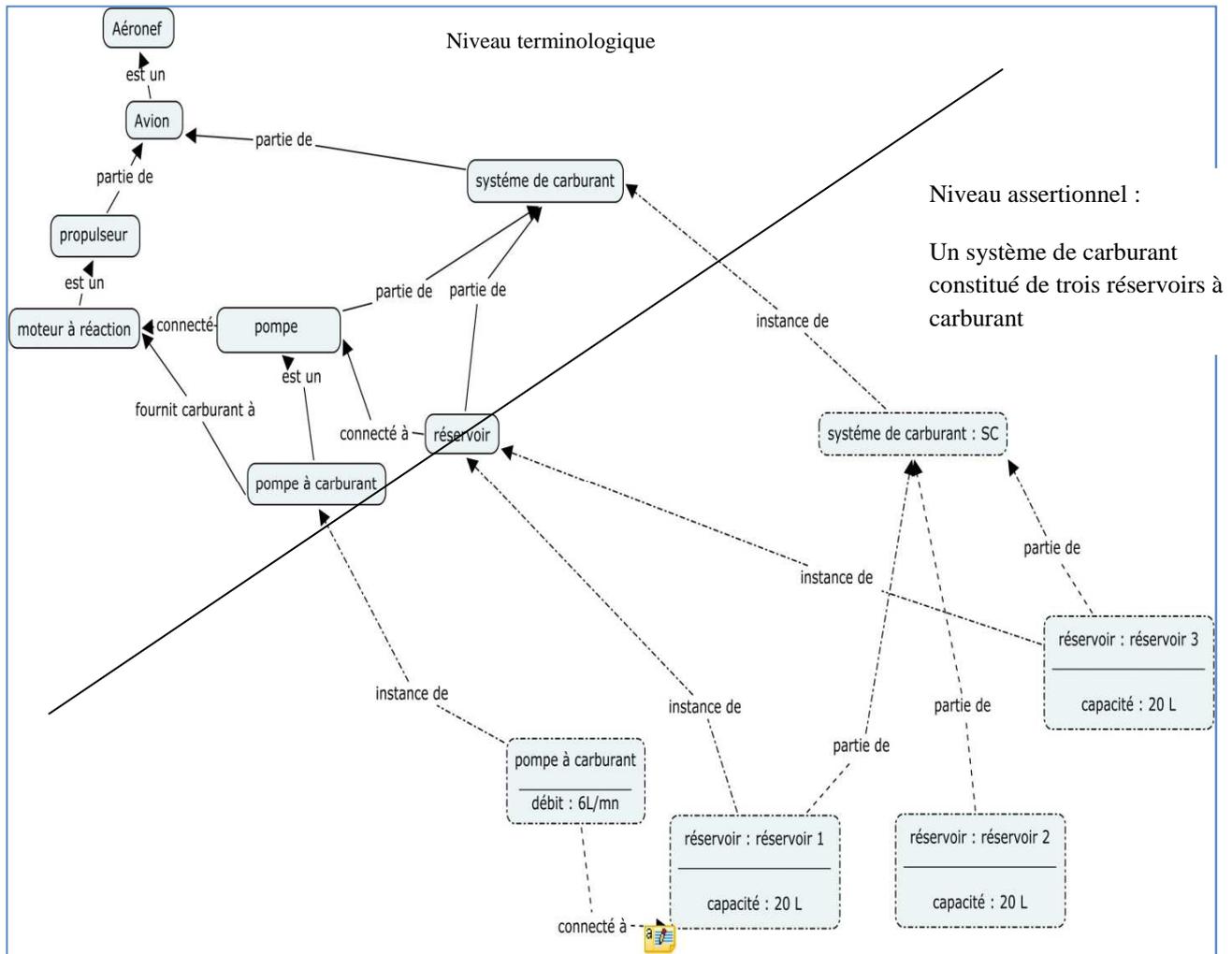


Figure V.7 Exemple d'assertion

### V.3.2. Modèle de capitalisation de CMS

Le modèle de capitalisation de CMS formalise les dimensions situationnelle, téléologique, argumentative et décisionnelle du modèle conceptuel de CMS proposé dans la section V.2.2.

Nous nous inspirons du concept d'annotation sémantique de modèles d'ingénierie tel que proposé par [Kimatura, 2006], [Zdrahal et al., 2003] et [Saeki et al., 2006]. Dans le cadre de ces travaux l'annotation sémantique est définie sous forme d'une fonction d'un ensemble de modèles vers un ensemble d'assertions instanciant les primitives conceptuelles d'une ontologie. L'objectif est d'affecter une sémantique précise à un modèle pour permettre sa réutilisation. ( Figure V.8)

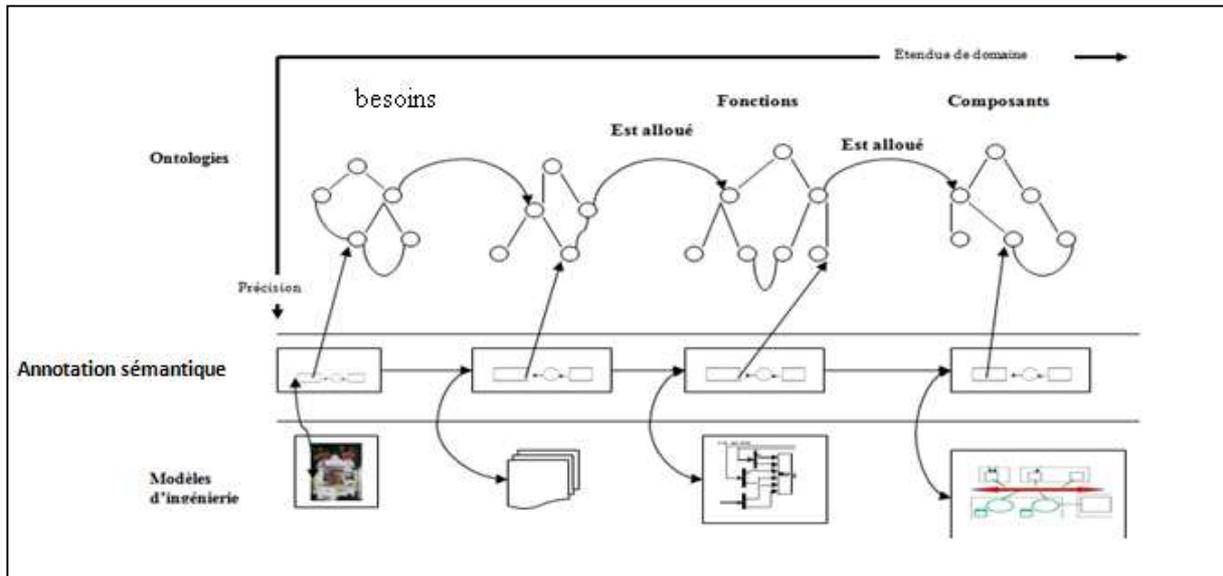


Figure V.8 Annotations sémantiques de modèles en IS [Chourabi et al., 2007]

Nous adoptons ce concept d'annotation sémantique pour formaliser les dimensions décrivant une CMS. A chaque dimension  $D_i$  nous faisons correspondre une annotation sémantique formée d'assertions instanciant les ontologies des facettes *système à faire* et *système pour faire* du cadre ontologique OntoIS.

L'ontologie de la facette *système à faire* modélise les primitives conceptuelles du domaine d'ingénierie selon les axes fonctionnels, organiques et contexte du système

$$O_{\text{Système à faire}} = (O_{\text{contexte système}}, F_{\text{fonctions}}, O_{\text{organes}})$$

Pour l'ontologie de la facette *système pour faire* nous retenons les deux axes processus et rôles pour décrire le contexte du choix d'ingénierie

$$O_{\text{Système pour faire}} = (O_{\text{processus}}, O_{\text{rôles}})$$

Nous notons :

- Annotation *Système à faire* pour désigner un ensemble d'assertions instanciant  $O_{\text{Système à faire}}$ , par exemple le choix d'une fonction de service d'un système
- Annotation *conditionemploi* pour désigner un ensemble d'assertions relatives aux exigences du système à développer. Par exemple le niveau de sécurité des voies dans un système de transport
- Annotation *système pour faire* pour désigner un ensemble d'assertions instanciant  $O_{\text{Système pour faire}}$ , par exemple l'étape du processus ayant donné lieu à une décision de projet.

Ces annotations sont combinées pour formaliser le modèle de capitalisation de CMS.

**Définition 4 :** (*modèle de capitalisation de CMS*)

Le modèle de capitalisation de CMS est un n-uplet = (annotation<sub>Situation</sub>, annotation<sub>But</sub>, annotation<sub>Alternatives</sub>, annotation<sub>Décision</sub>, avoir-but, avoir-alternatives, justifié-par, génère) où :

- Annotation<sub>Situation</sub> = (Annotation<sub>système pour faire</sub>, Annotation<sub>système à faire</sub>, Annotation<sub>condemp</sub>)
- Annotation<sub>But</sub> = (Annotation<sub>système pour faire</sub>)
- Annotation<sub>Alternatives</sub> = (Annotation<sub>système pour faire</sub>)
- Annotation<sub>Décision</sub> = (Annotation<sub>système pour faire</sub>)
- avoir-but : est une relation d'une Annotation<sub>Situation</sub> et d'une Annotation<sub>But</sub>
- avoir-alternatives : est une relation d'une Annotation<sub>But</sub> et d'une Annotation<sub>Alternatives</sub>
- génère : est une relation d'une Annotation<sub>Alternatives</sub> et d'une Annotation<sub>Décision</sub>
- justifié-par : est une relation d'une Annotation<sub>Situation</sub> et d'une Annotation<sub>Décision</sub>

**Exemple**

Nous considérons un scénario de capitalisation de CMS appliqué au système de transport automatique présenté dans la section V.2.2. L'ingénieur système est dans la phase d'analyse d'exigences, il a pour objectif d'allouer la fonction capturer position sachant que le système de transport doit parcourir une voie complexe comprenant une partie souterraine. Ce contexte d'ingénierie l'oriente à allouer la fonction de capture de position à un système GPS car celui-ci est plus précis qu'une centrale inertielle. Ce scénario est formalisé par le modèle de capitalisation de CMS sous cette forme :

- Annotation<sub>Situation</sub> =
  - o Annotation<sub>système pour faire</sub> = (analyse fonctionnelle : a)
  - o Annotation<sub>système à faire</sub> = (capturer vitesse : c)
  - o Annotation<sub>condemp</sub> = (parcours : complexe)
- Annotation<sub>But</sub> = réifier : r
- Annotation<sub>Décision</sub> = GPS : g
- Annotation<sub>Alternatives</sub> = centrale inertielle : c

La figure V.9 illustre ce scénario, avec les fragments d'ontologies des facettes *système à faire* et *système pour faire*

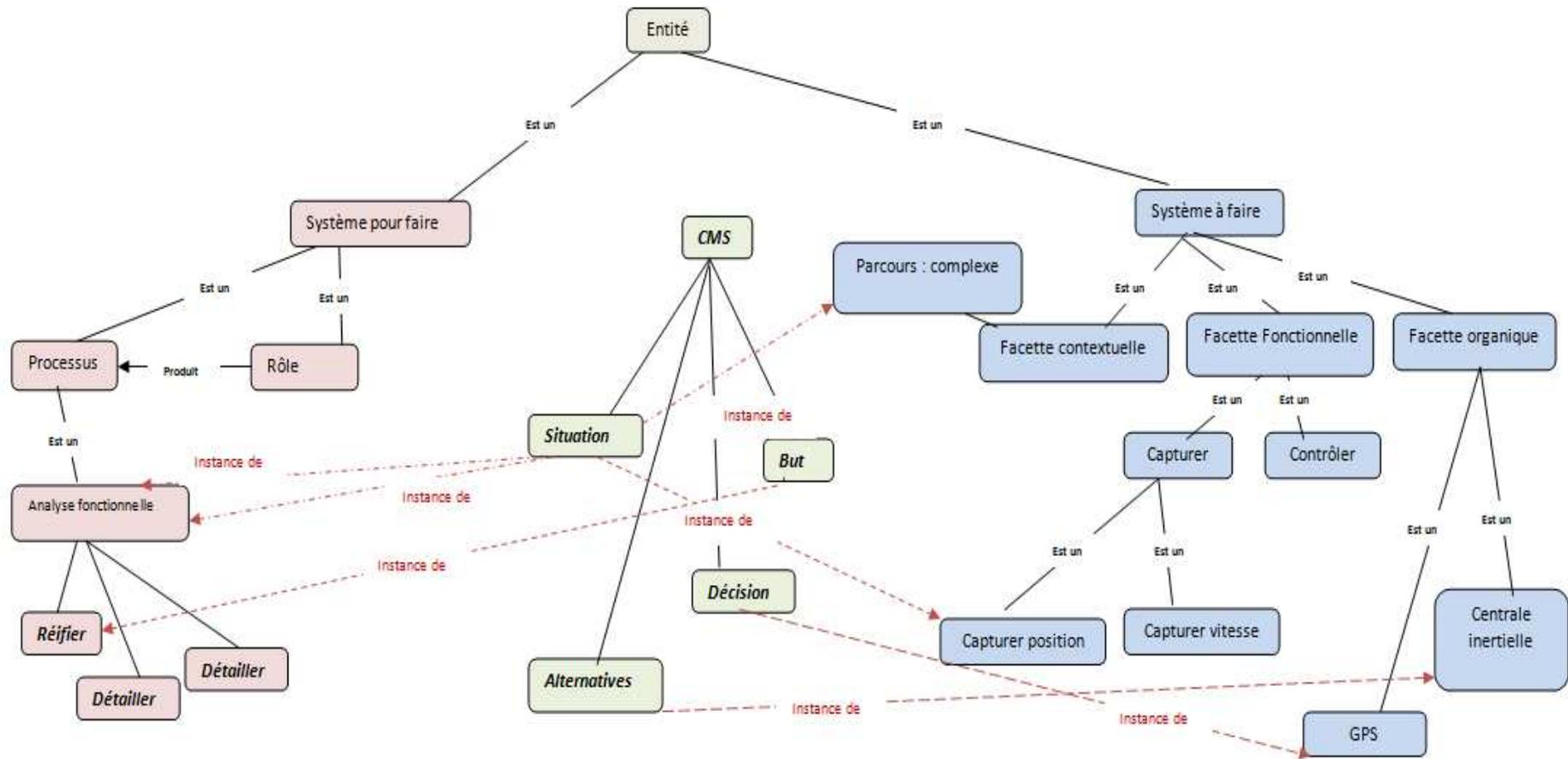


Figure V.9 exemple de capitalisation de CMS pour un système de transport automatique

### V.3.3. Modèle de partage de CMS :

Notre objectif est de définir un modèle de partage de CMS selon une approche proactive i.e. suggérant les CMS les plus appropriées à un contexte d'ingénierie d'un projet en cours. Pour ce faire, nous formalisons le modèle de partage de CMS sous la forme d'un appariement sémantique des dimensions situationnelles et téléologique du modèle de capitalisation CMS avec celles des situations de projets en cours.

Ces dimensions étant formalisées sous la forme d'annotations sémantiques reposant sur des modules ontologiques d'IS, nous nous plaçons alors dans le contexte général de la problématique de recherche à base d'ontologie<sup>53</sup>, fondée sur une approche logique [Gandon, 2009].

Nous formalisons le problème de partage de CMS par un quadruplet :  $(O, C_P, C_{cms}, F)$  où :

- **O** : est le modèle d'ontologie utilisé pour l'annotation.  $O = (T^*, AX)$
- **C<sub>P</sub>** : est le contexte d'un projet en cours, représentant les besoins en termes de CMS d'un ingénieur système. Le contexte du projet est représenté par l'ensemble des annotations sémantiques relatives à ses dimensions situationnelle et téléologique.

**C<sub>P</sub> = (annotation<sub>situation</sub>(annotation<sub>syspourfaire</sub>, annotation<sub>sysàfaire</sub>, annotation<sub>condemp</sub>), annotation<sub>but</sub>).**

- **C<sub>cms</sub>** : est le contexte d'une CMS.

**C<sub>cms</sub> = (annotationsituation (annotationsyspourfaire, annotationsysàfaire, annotationcondemp), annotationbut).**

- **F** : Une fonction d'appariement sémantique définissant les CMS potentiellement utiles dans un contexte de projet en cours. Dans le cadre d'un partage de connaissances basé sur les ontologies un appariement de  $C_P$  avec  $C_{cms}$  existe ssi  $C_{cms}$  et  $O$  impliquent logiquement  $C_P \cdot C_{cms} \wedge O \rightarrow C_P$  (Figure V.10)

La fonction d'appariement implémente une stratégie de résolution de problème dépendante des mécanismes d'inférences opérationnelles du formalisme de représentation de connaissances choisi pour opérationnaliser le modèle d'ontologie et les annotations sémantique. Par exemple, le test de satisfiabilité par la méthode de tableau pour les Logiques

---

<sup>53</sup> Le principe est celui de l'implication logique entre une requête et une description [Rijsbergen, 1986]: soit un modèle de descriptions ( $D$ ) et de requêtes ( $R$ ) alors une description  $D$  répond à une requête  $R$  si la description  $D$  implique logiquement la requête  $R$  noté  $D \rightarrow R$ . Il s'agit alors de trouver un algorithme d'appariement de ces descriptions logiques le plus efficace possible. Il changera d'un système à un autre en fonction des formalismes utilisés, de leur expressivité, et des caractéristiques des réponses que l'on veut assurer. Le principe d'une représentation et d'une recherche à base d'ontologies est que l'implication logique dans la fonction de recherche prend en compte les connaissances ontologiques. Soit un modèle de descriptions et de requêtes alors une description  $D$  répond à une requête  $R$  en considérant une ontologie  $O$  si la description  $D$  et l'ontologie  $O$  impliquent logiquement la requête  $R$  noté  $D \wedge O \rightarrow R$ .

de Descriptions [Prié, 2008], le mécanisme de projections pour les Graphes Conceptuels [Chein et al, 1996]

L'appariement sémantique à base d'ontologies présente une première solution au problème de partage de CMS. Même si cette solution est satisfaisante, nous suggérons d'étendre cet appariement en proposant directement à un ingénieur système les CMS pouvant être retenues comme solutions approchées ou potentiellement satisfaisantes dans le contexte de son projet. Nous présentons dans la suite un exemple motivant cette extension.

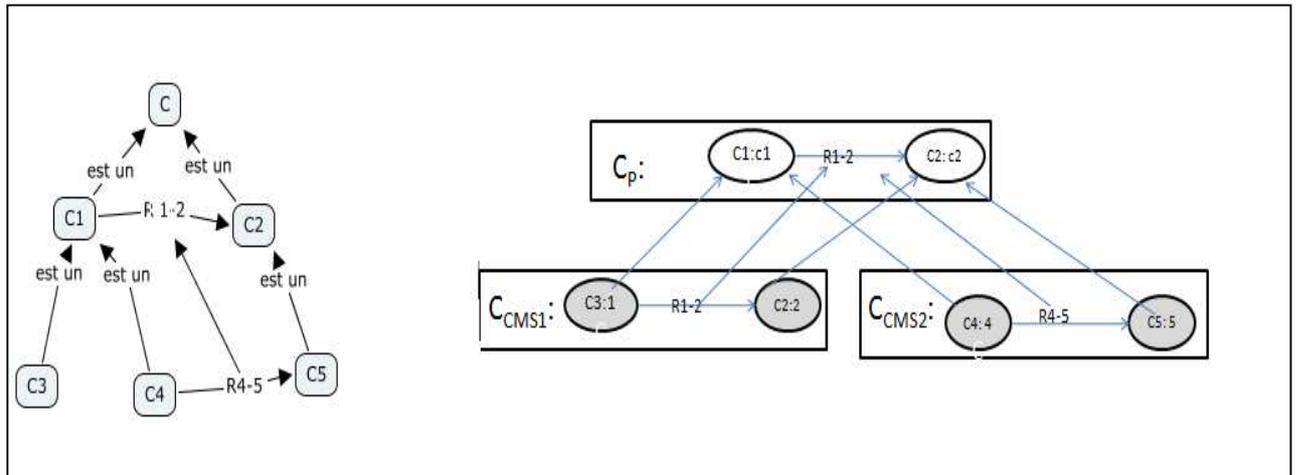


Figure V. 10 Appariement sémantique de  $C_p$  et de  $C_{CMS}$  basé sur les ontologies (test de subsomption entre  $C_p$  et  $C_{CMS}$ )

Nous illustrons nos motivations d'un appariement étendu en nous basant sur l'exemple du système de transport automatique.

Soit un contexte de projet en cours  $C_p$  exprimant le besoin suivant : un but *d'allocation de la fonction guider* à des *organes* dans des conditions d'emplois : parcours de 150 km.

Et soit le contexte d'une CMS exprimant un choix d'allocation des fonctions *capturer vitesse* et *capturer position* dans des conditions d'emplois : parcours de 300 km.

Cette CMS est une solution potentiellement utile dans ce contexte de projet car le choix d'ingénierie répond à des exigences plus sévères : i.e. si l'organe choisi est valable pour un parcours de 300 km il est à fortiori valable pour un parcours de 150 km. Ce type de raisonnement est applicable aux exigences exprimant les capacités (ou capabilité) d'un système.

Avec un appariement sémantique basé sur les ontologies la CMS n'est pas proposée à l'ingénieur système dans le contexte du projet que nous avons considéré, elle ne constitue pas une conséquence logique de  $C_p$ , car il n'existe pas de relation explicite liant ces deux conditions d'emplois. L'extension du modèle de partage de CMS a pour objectif d'intégrer dans le résultat de l'appariement les CMS potentiellement possibles présentant des exigences satisfaisantes dans un contexte de projet en cours. Cette extension se base sur l'introduction d'une nouvelle classe de paramètres de type capacité d'un système. L'appariement de la situation d'ingénierie avec la situation du CMS selon cette classe de paramètres exprime que

la valeur d'un paramètre de capabilité est considérée comme étant un seuil : il est possible de proposer un appariement pour un paramètre exprimant des exigences plus sévère que la situation en cours.

La réification de cette extension dépend du formalisme de représentation de connaissances utilisé pour opérationnaliser modèle de capitalisation et de partage de CMS. Nous formalisons cette extension dans le chapitre VI de la présente thèse.

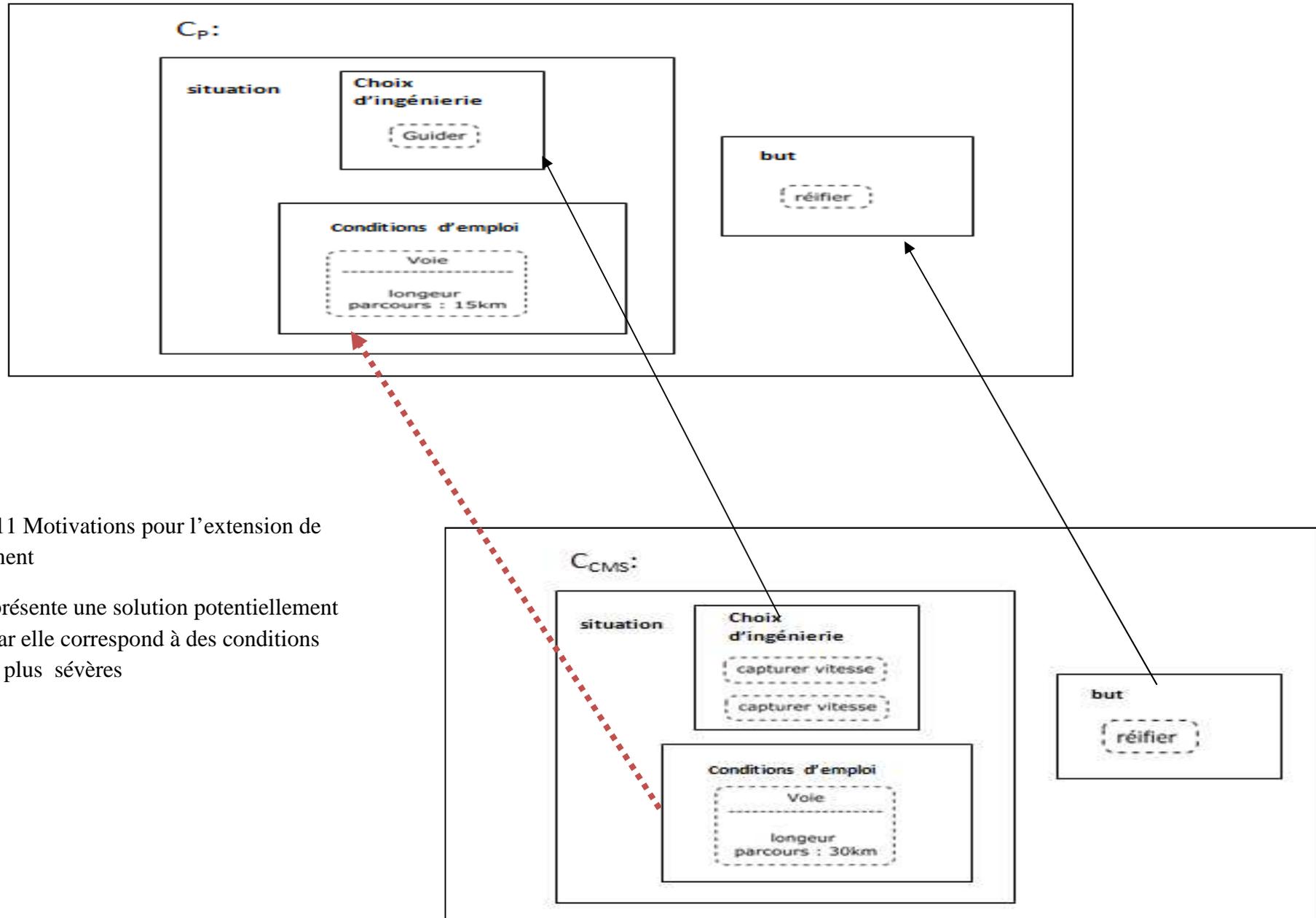


Figure V.11 Motivations pour l'extension de l'appariement

La CMS présente une solution potentiellement possible car elle correspond à des conditions d'emplois plus sévères

## V.4 Conclusion

Dans ce chapitre nous avons proposé une formalisation générique des expériences émergentes en cours des processus techniques d'IS : les Connaissances Métiers Situées.

Pour gérer ces connaissances nous avons introduit : Un modèle de capitalisation multidimensionnel où chaque dimension est formalisée sous la forme d'un ensemble d'annotations sémantiques basées sur des modules ontologiques du cadre OntoIS ; et un modèle de partage reposant sur l'appariement sémantique de ces annotations sémantiques de contexte de projet d'ingénierie avec celles des contextes des Connaissances Métiers Situées préalablement capitalisées. Pour évaluer ces deux modèles nous avons besoins de les transcrire dans un formalisme de représentation et de raisonnement sur ces connaissances offrant une sémantique formelle et computationnelle [Furst, 2004]. Nous étudions dans la partie III de cette thèse le potentiel des Graphes Conceptuels [Sowa, 1984] comme formalisme d'opérationnalisation de notre proposition.

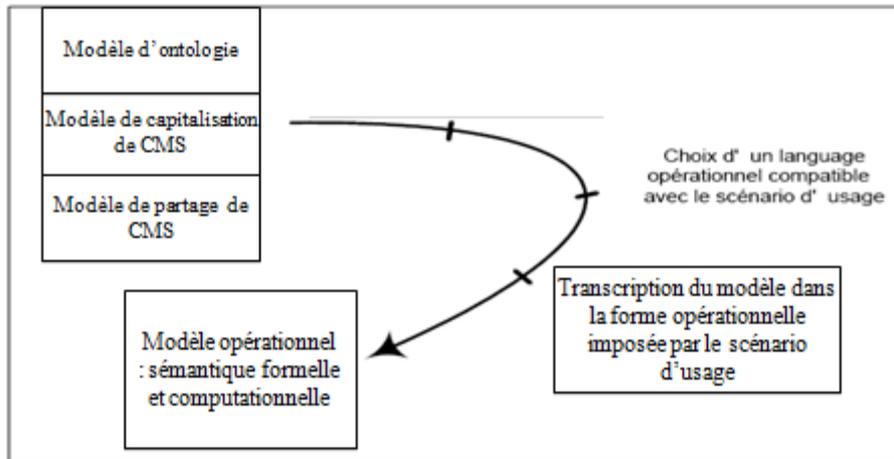


Figure V.12 : Opérationnalisation de la proposition de modélisation, capitalisation et partage de CMS

## **Partie III**

# **Opérationnalisation et Validation de la proposition**

# **Chapitre VI:**

## **Opérationnalisation de la proposition de Gestion des Connaissances Métiers Situées (CMS)**

---

### Plan du chapitre

---

VI.1 Introduction

VI.2 Le formalisme des Graphes Conceptuels

VI.3 Opérationnalisation du cadre ontologique OntoIS à l'aide des Graphes Conceptuels

    V.I.3.1 Opérationnalisation du modèle d'ontologie

    V.I.3.2 Opérationnalisation du modèle de capitalisation de CMS

    V.I.3.3 Opérationnalisation du modèle de partage de CMS

VI.4 Architecture d'un Système de Gestion de CMS ;

VI.5 Conclusion

## VI.1 Introduction

Ce chapitre a pour objectif de proposer une opérationnalisation du cadre ontologique OntoIS et de son application au Système de Gestion de Connaissances Métiers Situées. Par opérationnalisation nous entendons la transcription du modèle d'ontologie, des modèles de capitalisation et de partage de CMS dans un formalisme de représentation de connaissances offrant une sémantique formelle et des mécanismes appropriés de raisonnement [Leclère et al., 2005].

Parmi les principaux formalismes de représentations de connaissances utilisés pour opérationnaliser les ontologies, nous citons les Logiques de Descriptions [Baader, 1991], les Graphes Conceptuels [Sowa, 1984] et les Frames [Minsky, 1975]

Afin d'opérationnaliser notre proposition nous avons en particulier analysé les Logiques de Descriptions et les Graphes Conceptuels selon leurs pouvoir d'expressivité, de lisibilité, leurs mécanismes de raisonnement et leurs adéquation à notre contexte d'usage. (cf. tableau VI.1)

Nous avons retenu le formalisme des Graphes Conceptuels (GC) et plus particulièrement la famille Simple Graph (Simples Graphes SG) issues des travaux du laboratoire LIRMM<sup>54</sup> proposant une formalisation et des extensions à la proposition originelle de John SOWA [Sowa, 1984]. Les raisons de ce choix trouvent leur origine aussi bien dans des considérations théoriques que pragmatiques parmi lesquelles nous citons [Leclère et al., 2007][Carlorni et al., 2006] :

- La représentation des connaissances à l'aide de la famille SG est entièrement graphique. Cet aspect graphique nous semble important car il permet une interprétation intuitive des connaissances d'ingénierie tout en étant proche des langages de modélisation habituellement utilisés en IS.
- Les raisonnements de la famille de SG sont basés sur des opérations de graphe. Ils s'appuient directement sur les connaissances représentées, sans passer par un langage logique. Ceci fournit potentiellement des possibilités d'explication des raisonnements à l'utilisateur puisque les raisonnements peuvent être visualisés dans le formalisme qu'il connaît et directement sur les connaissances qu'il a définies. Les raisonnements sont consistants et complets par rapport à la sémantique formelle en logique de prédicats logique des connaissances représentées
- Il permet l'expression de connaissances variées : des descriptions, des patrons, des règles, des contraintes [Baget & Mugnier, 2002]
- Il permet une structuration et une contextualisation des connaissances grâce aux emboîtements de graphes.

---

<sup>54</sup> <http://www.lirmm.fr/~cogito/papers.html>

<b>Opérationnalisation de la proposition de CMS</b>	<b>Formalisme des GC et son extension avec la famille des SG du laboratoire LIRMM</b>	<b>Formalisme LD</b>
Fondements mathématiques	Les Graphes Existentiels de Charles PIERCE	L'Algèbre des relations de PEIRCE
Lisibilité	+	-
Expressivité	+ ( grâce aux extensions de la famille des SG )	++
Niveau Terminologique	Support de graphes conceptuels	Tbox
	Mécanisme de définitions de type [Badget, 2001]	Concepts et rôles dérivés
	Constructeurs de type de concepts par négation et conjonction	Plus de constructeurs de concepts selon la LD choisie
Niveau axiomatique	Règles et contraintes des SG	Pas de règles en LD
Niveau assertionnel	Graphes conceptuels simples	ABOX
	Graphes conceptuels emboîtés	

Tableau VI. 1 Etude comparative des formalismes GC et DL pour l'opérationnalisation du système de gestion de connaissances métiers situées

## VI.2. Le formalisme des Graphes Conceptuels (GC)

Dans cette section nous présentons les concepts de base du formalisme des GC, ainsi que les principales extensions de la famille SG du laboratoire LIRMM utilisées pour l'opérationnalisation de notre proposition.

Pour illustrer ces concepts, nous nous basons sur un exemple de système modélisé en SysML<sup>55</sup>, présenté dans les travaux<sup>56</sup> de [Leblanc, 2008]. L'exemple considéré est un système embarqué mobile simplifié, constitué d'une batterie et d'une carte de traitement qui transforme des entrées en sorties selon un débit fixe. La carte est alimentée par une batterie.

### *Définition 1 (le support)*

Un support de graphes conceptuels est un quadruplet  $\mathbf{S} = (\mathbf{T}_C, \mathbf{T}_R, \mathbf{I}, \sigma)$ . Avec

- $\mathbf{T}_C$  : est un ensemble de types de concepts
- $\mathbf{T}_R$  : est un ensemble de types de relations d'arité quelconque (l'arité étant nombre d'arguments d'une relation). La notation  $T_{R2}$  désigne un ensemble de type de relations binaires
- $\mathbf{T}_C, \mathbf{T}_R$  sont partiellement ordonnées, l'ordre partiel représente la relation de spécialisation ( $\mathbf{t} \leq \mathbf{t}'$  est interprétée comme  $\mathbf{t}'$  est une spécialisation de  $\mathbf{t}$ ).
- $\mathbf{I}$  : un ensemble de marqueurs individuels. Ces marqueurs permettent d'identifier les concepts, c'est-à-dire les instances des types de concepts. Le support défini en outre un marqueur dit générique, noté \* pour désigner des concepts non instancié. L'ordre entre les éléments de  $\mathbf{I} \cup \{*\}$  est tel que deux éléments de  $\mathbf{I}$  sont incomparables et le marqueur \* est le plus grand des éléments.
- $\sigma$  : associe à toute relation une signature qui définit le type maximal de chacun de ses arguments.

---

<sup>55</sup> [www.sysml.org](http://www.sysml.org)

<sup>56</sup> Le processus appliqué est basé sur la recommandation méthodologique de Telelogic (www.Telelogic.com) appelée Harmony/SE (SE pour System Engineering).

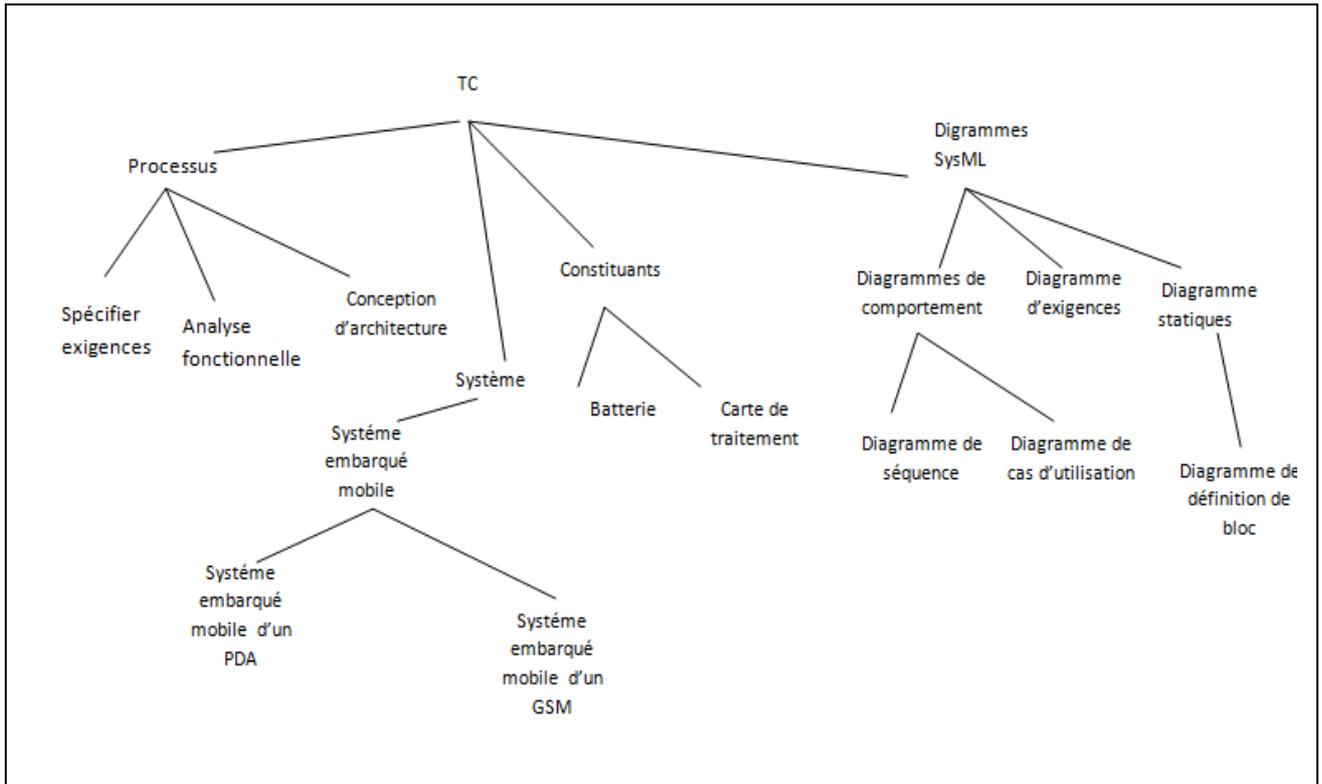


Figure VI.1 Treillis de type de concepts :  $T_C$

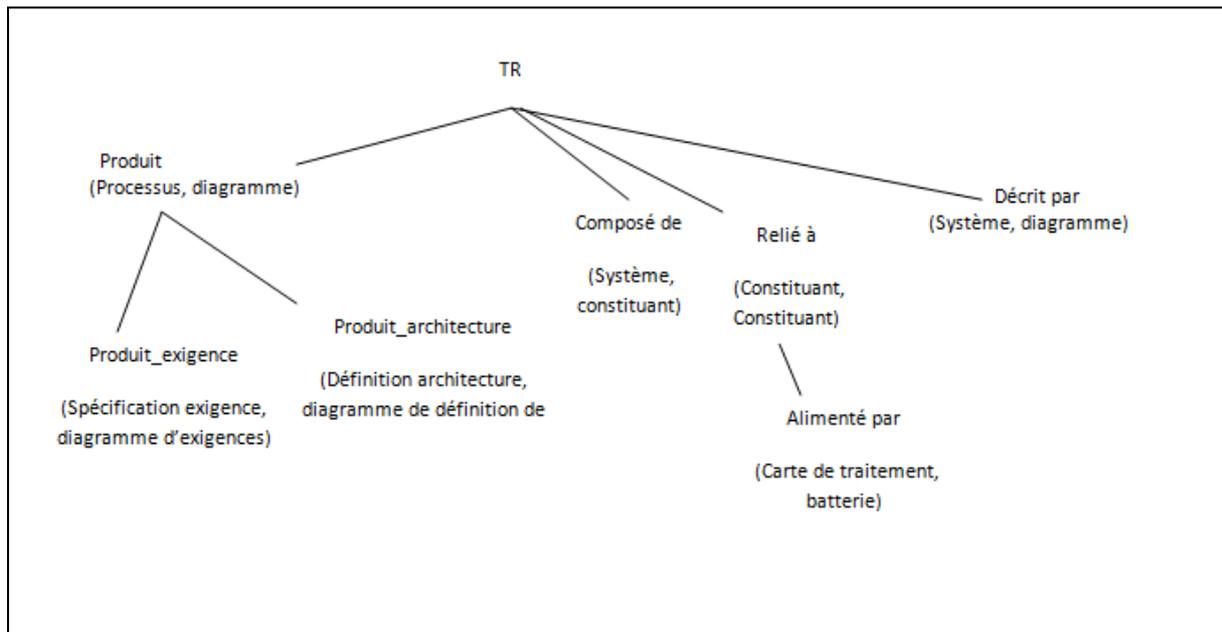


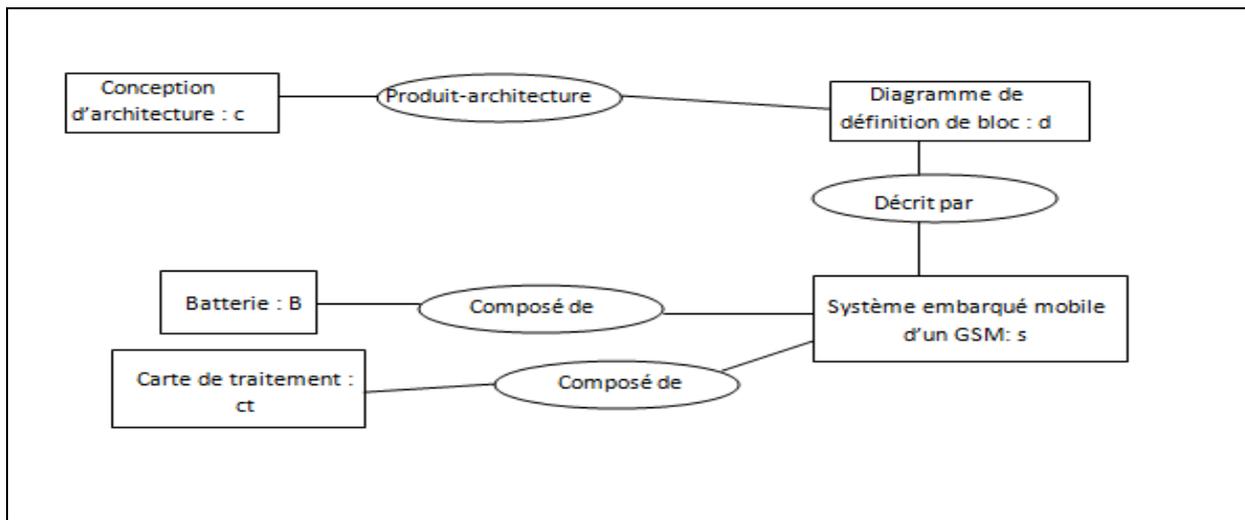
Figure VI.2 Treillis de type de relations :  $T_R$

**Définition 2 (Graphe Conceptuel Simple)**

Un graphe conceptuel simple (SG)  $G$  défini à partir d'un support  $S$  est un multi graphe biparti fini

$G = (CG, RG, EG, IG)$  où  $CG$  est l'ensemble des sommets concepts,  $RG$  l'ensemble des sommets relations.  $EG$  est l'ensemble des arêtes multiples entre  $CG$  et  $RG$ .

$IG$  est la fonction d'étiquetage des sommets et des arêtes. Un sommet concept est étiqueté avec  $IG$  par un couple  $t : m$  où  $t$  est un type de concept de  $TC$ ,  $m$  est un marqueur appartenant à  $I \cup \{*\}$ .



Exemple de graphe conceptuel simple basé sur le support VI.1, VI.2.

**Définition 3 : enrichissement du support par des types défini**

Badget propose dans [Badget, 2001] des mécanismes de définitions de types empruntés à la logique de description, de conjonctions de types et plus récemment de raisonnement sur des types de données. Deux types de concepts peuvent être utilisés dans un graphe conceptuel :

- Un Concept Primitif qui est juste une étiquette et un positionnement dans le support
- Un Concept Défini dont la définition est donnée par un graphe conceptuel

Exemple :

Système embarqué mobile = [carte de traitement :\* ] → (alimentée par) → [batterie :\*]

**Définition 4 (Graphe Emboîtés Typés)**

Ce modèle a été introduit pour contextualiser des connaissances par l'équipe du LIRMM dirigée par Michel CHEIN [Chein et al., 1998] [Mugnier et al., 2007].

Cette classe est récursivement définie par :

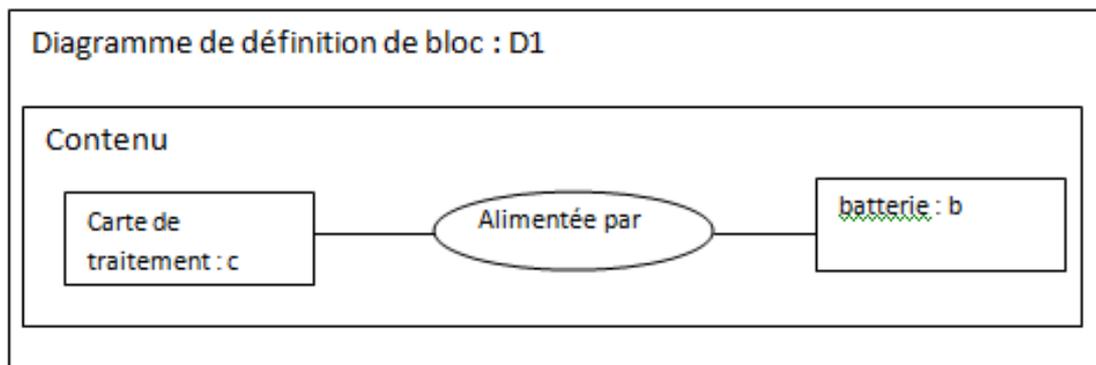
1. Un graphe emboîté typé est obtenu à partir d'un SG en ajoutant à chaque sommet concept **c** un troisième champ, égal à \*\*

2. Soient G un graphe emboîté typé,  $c_1, c_2, \dots, c_k$  des sommets concepts de G,

$n_1, n_2, \dots, n_k$  des types d'emboîtement, et  $G_1, G_2, \dots, G_k$  des graphes emboîtés

typés. Le graphe obtenu en substituant l'ensemble  $\{(n_1, G_1), (n_2, G_2), \dots, (n_k, G_k)\}$  au troisième argument \*\* de  $c_i$  est un graphe emboîté typé.

Les types d'emboîtements TE sont ajoutés au support des graphes conceptuels. Les types d'emboîtement permettent de préciser la nature du lien qui lie un concept à son graphe emboîté (description, association, composition, historique...), Graphiquement, les couples  $\{(n_i, G_i)\}$  associés au sommet **c** sont représentés par des boîtes à l'intérieur du sommet **c**, étiquetées par  $n_i$  et contenant le graphe  $G_i$ .



Exemple de graphe emboîté typé basé sur le support VI.1, VI.2.

**Définition 5 (Projection) [ Sowa, 1984 ]**

La notion fondamentale pour comparer des SG est une application d'un SG dans un autre appelée projection (un homomorphisme de graphes en termes de théorie des graphes).

Intuitivement, l'existence d'une projection de G (CG, RG, EG, IG) dans H (CH, RH, EH, IH) montre que la connaissance représentée dans G est contenue dans H, autrement dit que G se déduit de H, ou encore que H est une spécialisation de G (noté  $H \leq G$ )

Une projection  $\pi$  de G dans H est une application de CG dans CH et de RG dans RH qui conserve les arêtes et peut “spécialiser” les étiquettes des sommets.

G et H sont respectivement les graphes source et cible de la projection.

Une base de graphes conceptuels B peut être interrogée par le mécanisme de projection. Sous sa forme la plus simple, une requête, est elle-même un SG. Par exemple, traiter le SG requête 1 de la figure V.1. 3. comme une requête, consiste à rechercher les occurrences du motif suivant : “diagrammes statique décrivant un système embarqué mobile ”. La base de graphes conceptuels B répond à la requête s’il existe une projection de requête 1 dans B. par exemple un résultat de projection peut être les diagrammes de définition de bloc décrivant un système embarqué mobile d’un GSM.

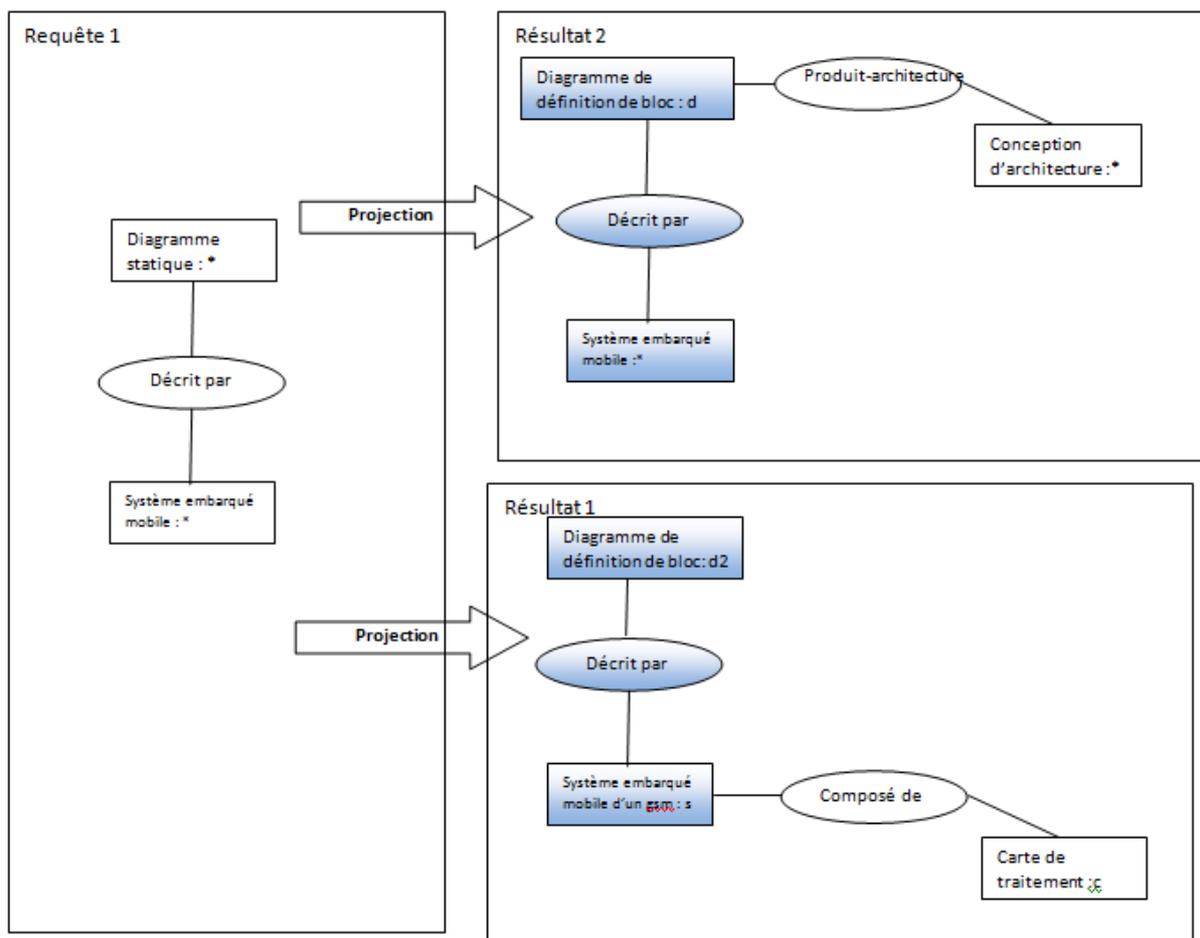


Figure V.1. 3 : Projection de graphes conceptuels : les sommets spécialisés sont représentés en bleu

**Définition 6 : règles**

Une règle d'inférence exprime une connaissance de la forme "si hypothèse alors conclusion", où hypothèse et conclusion sont deux SG. Les pointillés lient certains sommets de l'hypothèse et de la conclusion ; ces sommets sont appelés sommets frontières. Une règle R s'applique à un SG F s'il existe une projection de l'hypothèse de R dans F. L'application de R à F selon une telle projection consiste à "attacher" à F la conclusion de R, en fusionnant chaque sommet frontière de la conclusion avec l'image par projection du sommet frontière lui correspondant dans l'hypothèse.

Une base de graphes conceptuels B est composée du support et d'un ensemble de règles R. Le mécanisme d'interrogation doit donc prendre en compte la connaissance implicite encodée dans les règles.

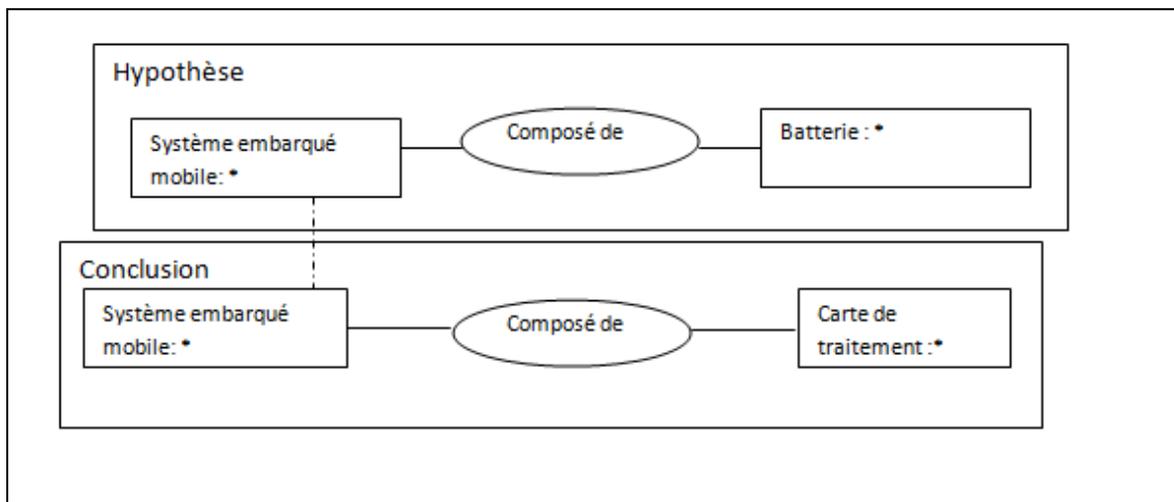


Figure VI. 4 Exemple de règle SG

La règle de la figure VI. 4 exprime qu'un système embarqué mobile constitué d'une batterie, doit nécessairement être constitué d'une carte de traitement.

## VI.3 Opérationnalisation du Système de Gestion de CMS à l'aide des GC

Dans cette section nous présentons les principes d'opérationnalisation de la proposition de système de gestion de connaissances métiers (chapitre V section V.3) à l'aide du formalisme des graphes conceptuel. Nous analysons plus particulièrement les adaptations nécessaires à apporter au modèle d'ontologie pour proposer un appariement sémantique étendu des connaissances métiers situées à un projet d'ingénierie en cours.

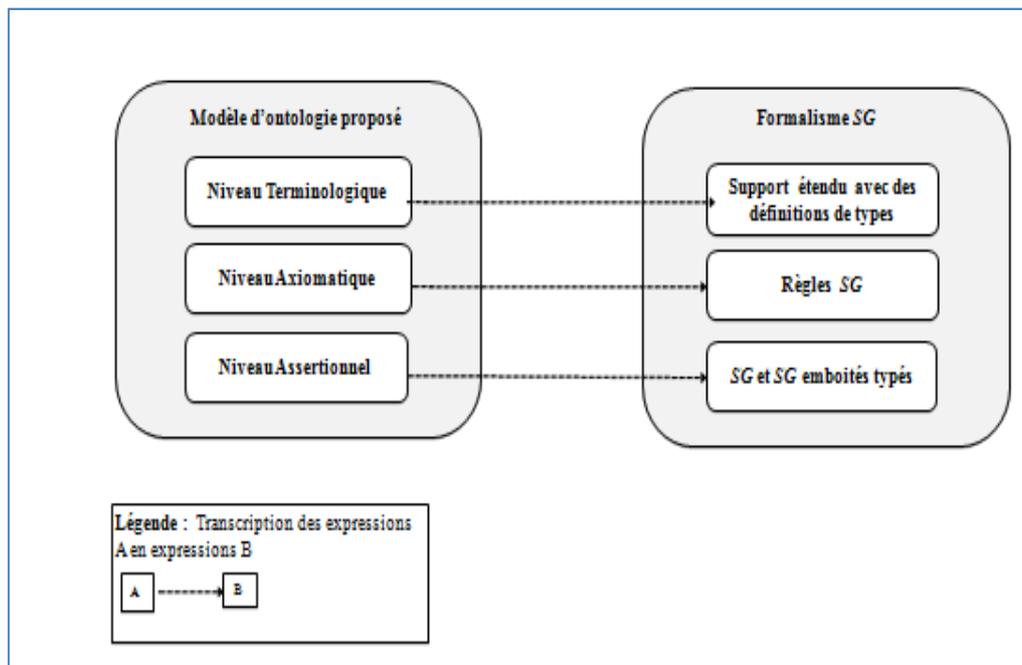


Figure V.1. 5 : Principes de l'opérationnalisation de la gestion de connaissances métiers situées à l'aide du formalisme des graphes conceptuels

### VI.3.1 Opérationnalisation du modèle d'ontologie

Nous avons présenté dans le chapitre V, un modèle d'ontologie est défini par un couple  $(T, Ax)$  où  $T$  représente le niveau terminologique,  $Ax$  représente le niveau axiomatique.

Nous expliquons dans ce qui suit les principes d'opérationnalisation de ce modèle avec les graphes conceptuels.(GC).

– **Transformation du niveau terminologique en support de graphe conceptuel**

Le formalisme des GC offre une représentation des connaissances terminologiques au niveau d'un support constitué de treillis de types et de relations.

Soit :

**Soient :**

Le niveau terminologique,  $\mathbf{T} = (\mathbf{C}, \leq_{\mathbf{C}}, \mathbf{R}, \leq_{\mathbf{R}}, \mathbf{Sign}, \mathbf{P}, \leq_{\mathbf{P}}, \mathbf{TD}, \mathbf{UM}, \mathbf{P}_{\text{applic}}, \mathbf{TD}_{\text{applic}}, \mathbf{UM}_{\text{applic}})$

Et un support de GC,  $\mathbf{S} = (\mathbf{TC}, \mathbf{TR}, \mathbf{I}, \sigma)$ .

**Le Treillis de type de concept  $T_C$**

Quatre types de concepts : C, P, UM, TD sont créés comme sous type du type universel TC. Ces sous-types représentent respectivement les concepts, les paramètres, les unités de mesures et les types de données. Ces concepts sont organisés par la relation de spécialisation de types de concepts.

Le support des GC ne définit pas de mécanisme de représentation des paramètres (attributs) et de types de données dans le support. Ces derniers peuvent être représentés sous la forme de types de concepts ou de relations. Nous avons opté de les représenter au niveau des types de concepts pour les besoins de définitions de paramètres de types capacité de systèmes.

**Le Treillis de type de relations  $T_R$**

Quatre types de relations : R, TP, TTD, TUM sont créés comme sous-types (?) du type universel TR. Ces sous-types représentent les relations et les associations entre paramètres, types de données et unités de mesures. Ces sous-types sont reliés par des signatures  $\sigma$  exprimant les types maximaux de leurs arguments.

Ces deux treillis permettent d'exprimer le niveau assertionnel formé de graphes conceptuels simples ou emboîtés

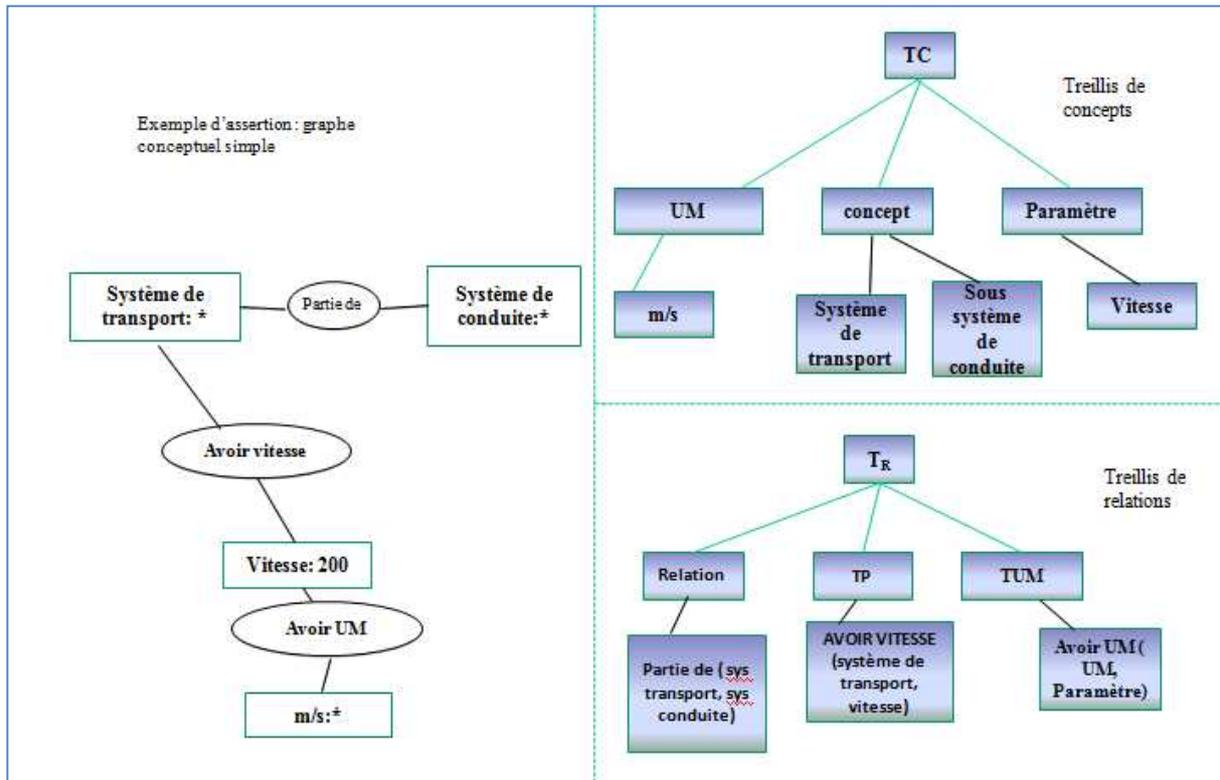


Figure VI.6 : Opérationnalisation du modèle d'ontologie en graphe conceptuel

### Le Traitement des paramètres de capacité de système (chapitre V.3.3)

Dans la définition du modèle de partage de CMS, nous avons présenté nos motivations d'extension de l'appariement sémantique d'un contexte de projet d'ingénierie en cours et d'un contexte de CMS. Nous avons introduit une nouvelle classe de paramètre dénommée « paramètre de capacité système » permettant d'exprimer les CMS potentiellement utiles dans une situation d'ingénierie. Afin d'opérationnaliser cet appariement étendu, nous adaptons le support des graphes conceptuels pour expliciter les relations entre paramètres capacité systèmes.

Nous nous basons sur le mécanisme de définition de type pour étendre le support des graphes conceptuels par des paramètres définis. Cette extension correspond au besoin de représenter des types de paramètres de capacité de système, et permettra d'offrir un mécanisme de raisonnement étendu. Ce mécanisme de raisonnement en GC correspond à l'opération de projection de graphes conceptuels. Nous présentons un exemple illustrant le besoin de cette extension puis le mécanisme de définition de types de paramètres capacité.

L'exemple correspond à l'appariement sémantique d'un contexte de projet et d'un contexte de CMS avec :

**-Contexte de projet :** Besoin de concevoir un missile apte à fonctionner dans de bonnes conditions météorologiques

-**Contexte de CMS** : un choix d'ingénierie capitalisé pour un missile fonctionnant dans toutes conditions météorologique. Alors ce choix d'ingénierie peut constituer une solution possible pour le projet en cours. Or avec la définition de base de l'opération de projection de graphes conceptuels ne permet pas de raisonner sur les instances d'un attribut de concepts. La projection donne un résultat uniquement dans le cas de spécialisation de concepts ou de relations dans un support. Dans la figure V.1. 7 , la CMS n'est pas appariée à la situation de projet.

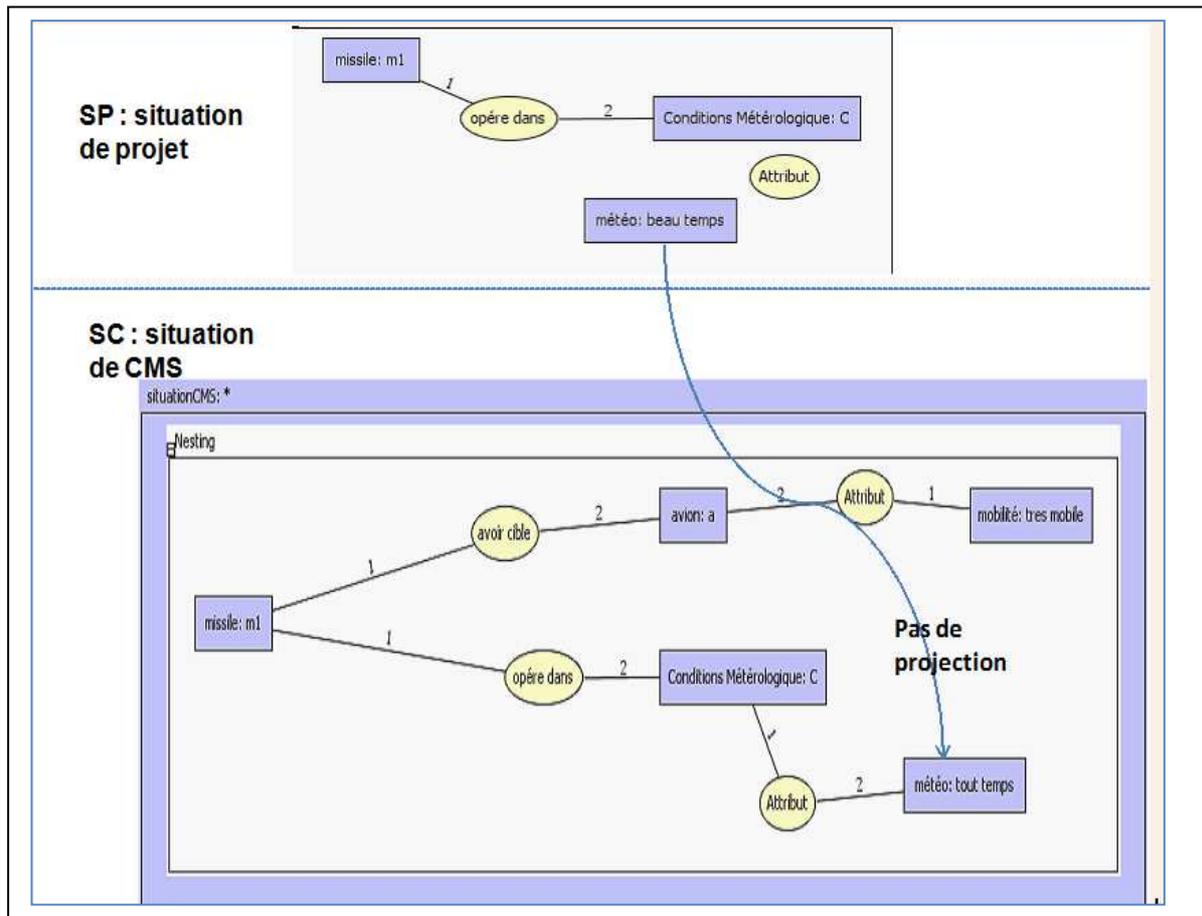


Figure VI 7 : besoins d'extension des paramètres de types capacité

Pour rendre cette projection possible, nous associons pour chaque paramètre de type capacité un type de concept défini et nous définissons une relation d'ordre partiel pour ces concepts définis :

**Définition 4** : un paramètre de type capacité est un sommet concept dont l'étiquette est un triplet [[paramètre, valeur], \*] où [paramètre, valeur] est un type de concept construit dans le support des graphes conceptuel

Exemple : [[Météo,tout\_temps], \*] = [météo: tout temps ]

**Définition 5 :** soient  $P1 = [paramètre1, valeur1]$  et  $P2 = [paramètre2, valeur2]$  deux étiquettes de concepts définis. Alors  $P2$  est une spécialisation de  $P1$  ( $P2 \leq P1$ ) si et seulement si  $Paramètre 1 = Paramètre 2$  et  $valeur2 \geq valeur1$

**Propriété :** l'ensemble des paramètres de type capacité de systèmes  $P_{capabilité}$  muni de la relation de spécialisation est un ensemble partiellement ordonné<sup>57</sup>. La relation de spécialisation vérifie les propriétés de réflexivité, d'antisymétrie et transitivité. Cette relation exprime la sémantique « inclut la capacité de »

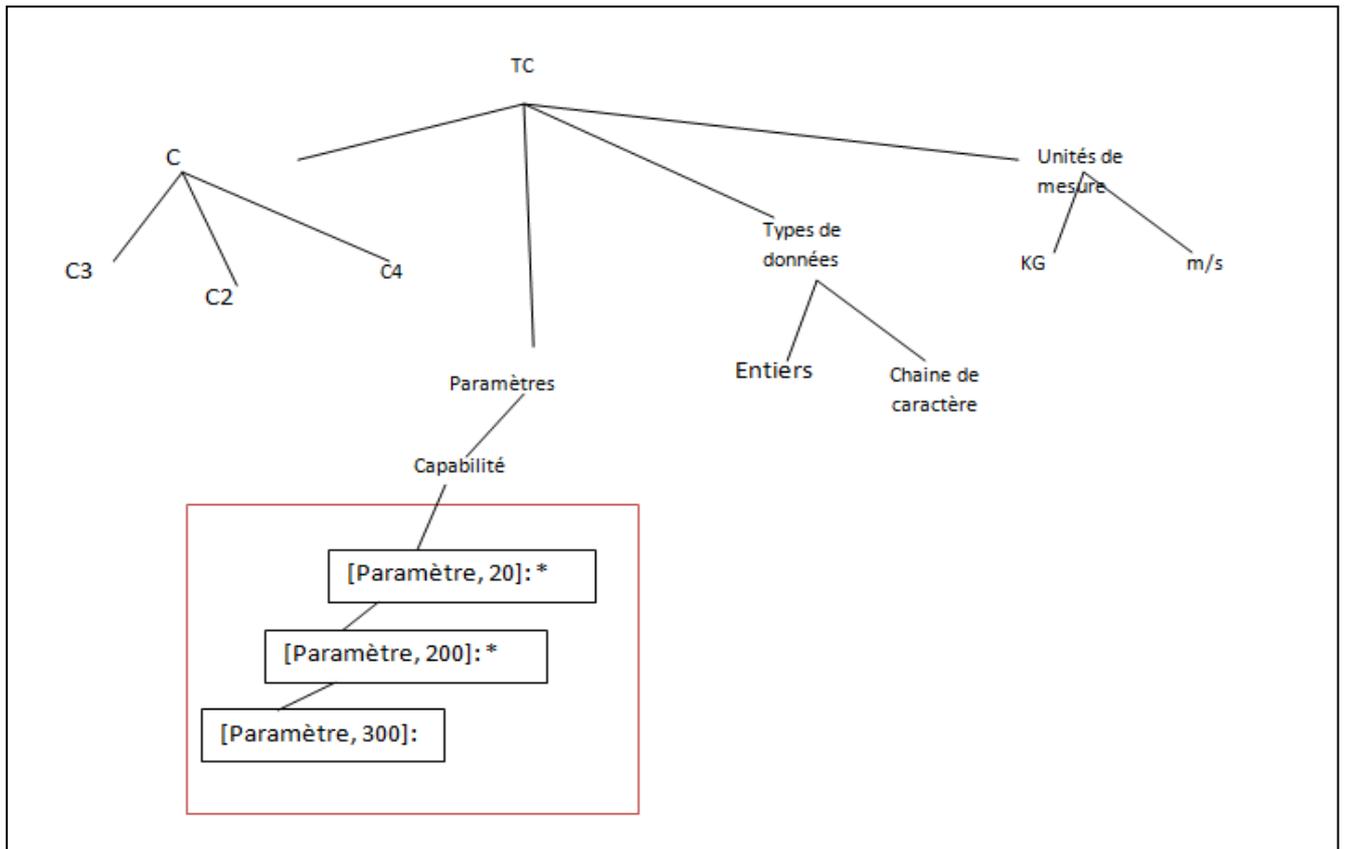


Figure VI. 8 treillis de concepts étendu

<sup>57</sup> Ensemble partiellement ordonné : soit  $P$  un ensemble d'éléments quelconques, liés par une relation binaire quelconque (exemple  $x \leq y$ ) qui satisfasse aux conditions suivantes :

$P1 : \forall x, x \geq x$

$P2 : si x \geq y et y \geq x alors x = y$

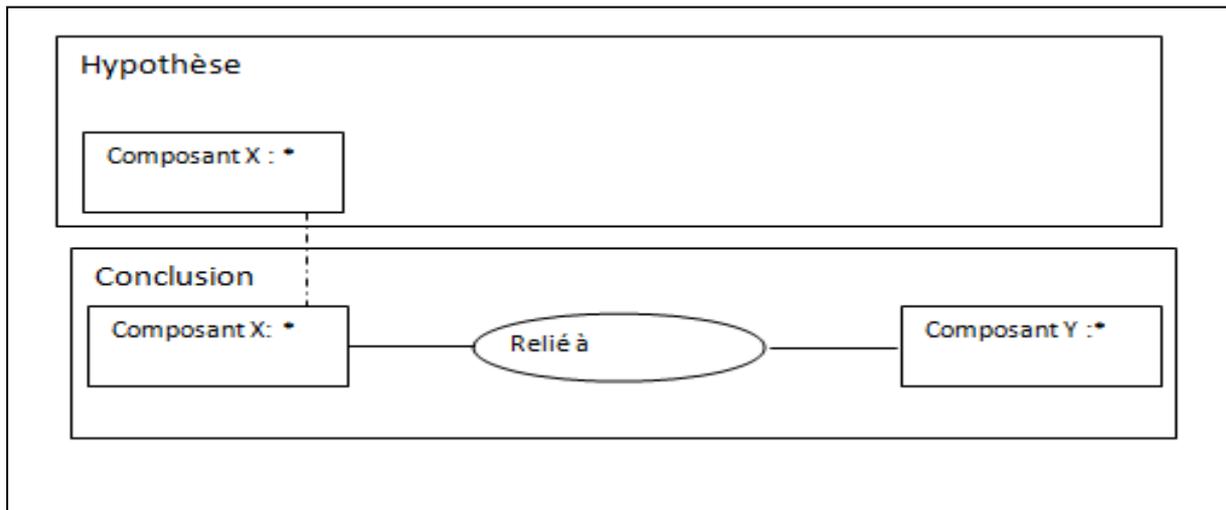
$P3 : si x \geq y et y \geq z alors x \geq z$

$P$  s'appelle ensemble partiellement ordonné

– **Transformation du niveau axiomatique en règles de graphe conceptuel**

Le niveau axiomatique défini dans le chapitre V (section V.3.1.2), est exprimé par des règles de graphes conceptuels. Nous utilisons ces règles dans un contexte de validation de contenu d'une connaissance métier située.

Un axiome de domaine exprimant une contrainte de type nécessité est représenté par :



Si la décision d'une CMS comporte un composant X alors l'application de cette règle en phase de capitalisation dans le référentiel de connaissances permet d'enrichir le contenu de la décision en y attachant la conclusion de la règle.

### VI.3.2 Opérationnalisation du modèle de capitalisation de CMS

Le modèle de capitalisation de CMS que nous avons définis (chapitre V) est une structure multidimensionnelle :

- une dimension situationnelle : représentant le contexte d'ingénierie dans lequel un ingénieur système est amené à prendre une décision (phase du processus, exigences, choix d'ingénieries déjà effectuées)
- une dimension téléologique : représentant un but d'ingénierie. où nous nous considérons plus particulièrement les sous buts : décomposer, détailler et réifier (pour exprimer l'allocation des fonctions aux composants)
- une dimension argumentative : représentant les alternatives de choix d'ingénierie
- une dimension décisionnelle : représentant un choix d'ingénierie

Chaque dimension est représentée par un ensemble d'annotations sémantiques. Pour rendre compte de cet aspect multidimensionnel, nous avons opérationnalisé le modèle de capitalisation de CMS sous la forme de graphes emboîtés typés.

Les graphes emboîtés typés offrent des avantages en termes de contextualisation des éléments du modèle de capitalisation. A chaque dimension du modèle de CMS correspond un type d'emboitement. Et à chaque annotation sémantique correspond un graphe conceptuel simple exprimant une assertion.

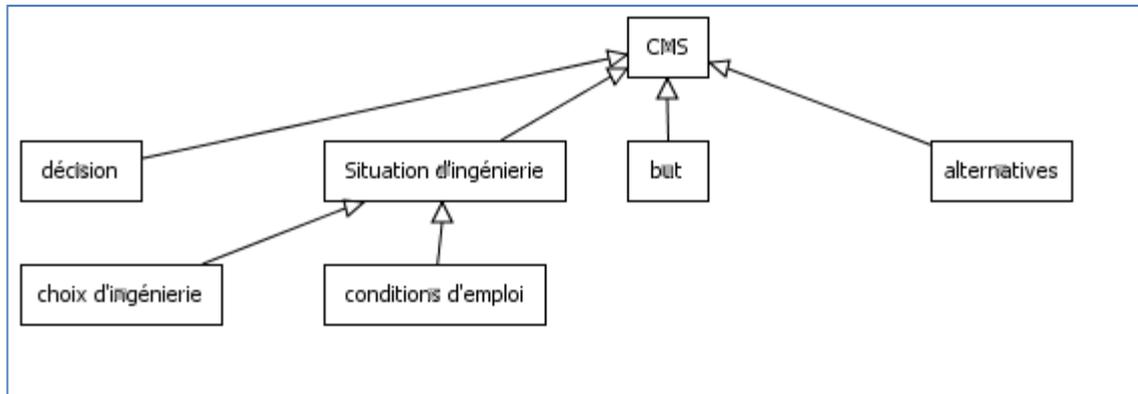


Figure VI. 9 Types d'emboitement du modèle de capitalisation de CMS

**Définition 6 : modèle de capitalisation de CMS en graphes conceptuels emboîtés**

Soit  $GC = (CG, RG, EG, LG)$  un graphe conceptuel définis par rapport à un support de graphes conceptuels système pour faire

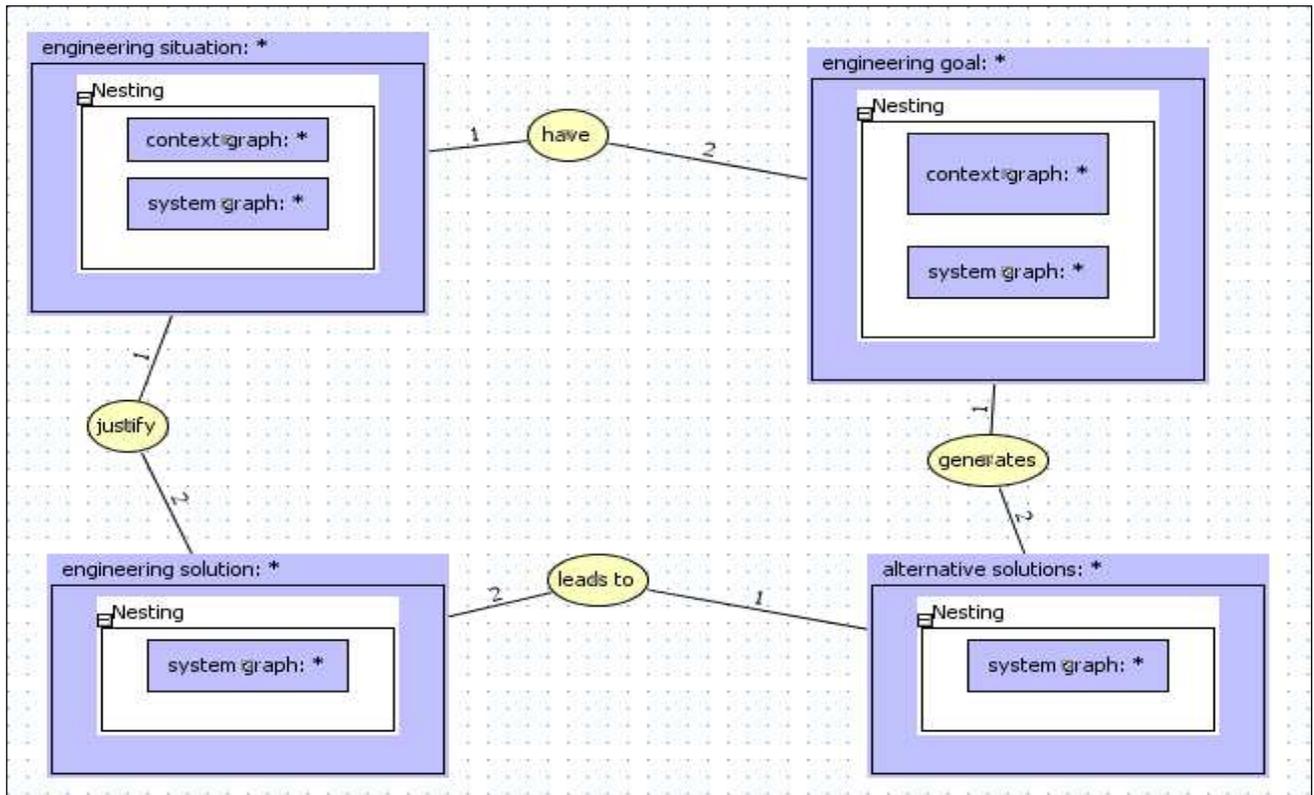
Soit  $GS = (C, R, E, L)$  un graphe conceptuel définis par rapport à un support de graphes conceptuels système à faire

Le modèle de capitalisation de CMS est un graphe emboîté typé composé de quatre types d'emboitement :

- (Situation d'ingénierie : { GC }, { GS } ),
- (but , GC),
- (alternatives, { GS } ),
- (décision, { GS } )

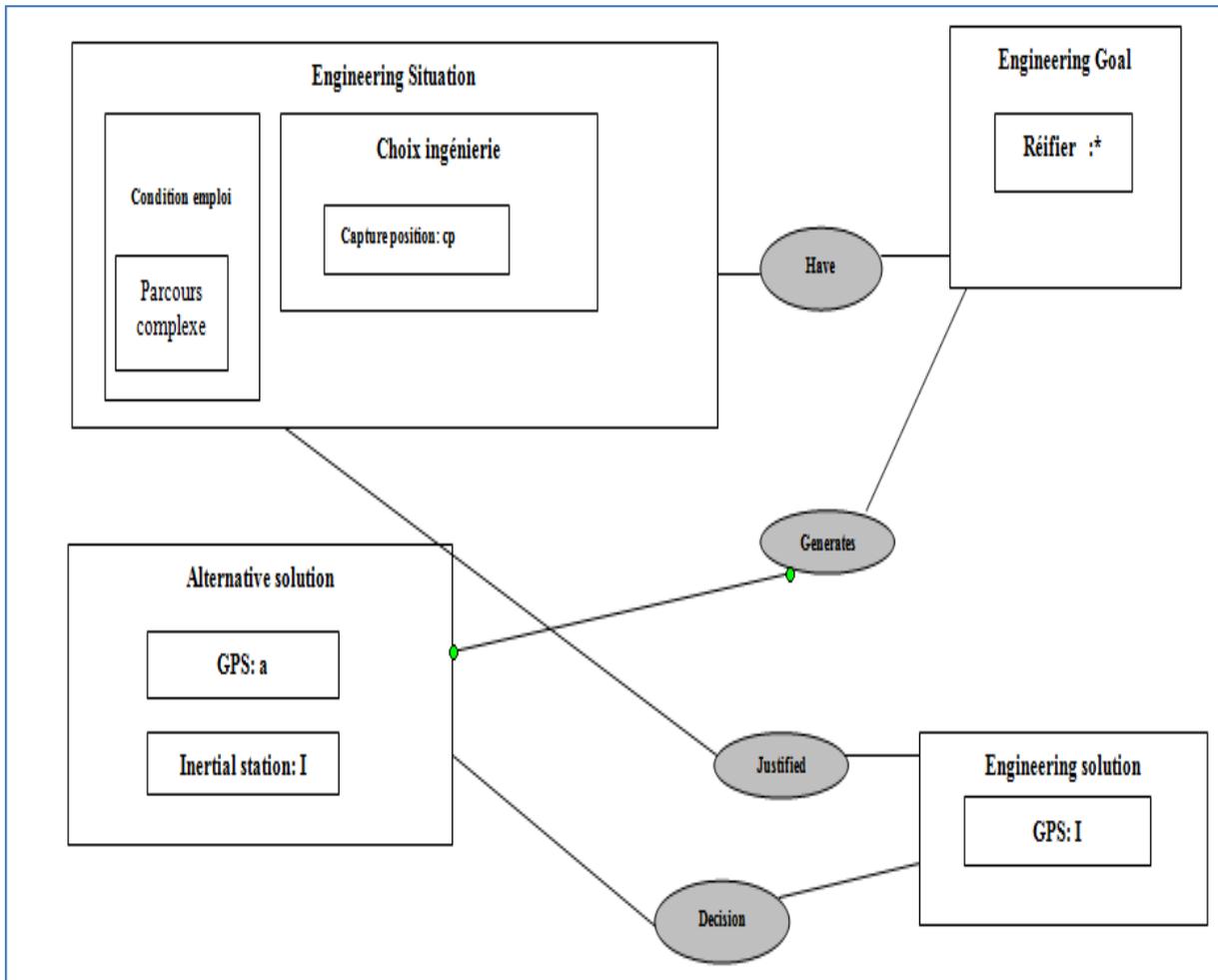
Ces types d'emboitement sont reliés par les relations : avoir-but, avoir-alternatives, justifié-par, génère.

La figure VI.10 présente la transcription du modèle générique de capitalisation en graphes emboîtés typés.



VI.10 : CMS sous forme de graphe emboîté typé

La figure VI.11 présente l'exemple de capitalisation de CMS dans un cas d'ingénierie d'un système de transport présenté dans la chapitre V sous la forme de graphe conceptuel emboîté : la situation d'ingénierie est représentée par deux graphes conceptuels simples exprimant les choix d'ingénierie fonctionnels (capturer position) et les conditions d'emplois (parcours complexe), le but est représenté par un graphe conceptuel simple (réifier), les alternatives par un graphe conceptuel simple (GPS, centrale inertielle), la décision par un graphe conceptuel simple(GPS).



VI.11 Exemple de graphe conceptuel emboité : système de transport automatique

### VI.3.3 Opérationnalisation du modèle de partage de CMS

La notion fondamentale pour comparer les graphes conceptuel est une application d'un SG dans un autre appelée projection (un homomorphisme de graphes en termes de théorie des graphes). L'opération de projection permet d'identifier dans un graphe  $g_1$  les sous-graphes avec la même structure qu'un graphe  $g_2$ , avec des nœuds éventuellement restreints, i.e. leurs types sont des spécialisations des types de nœuds de  $g_2$ .

L'existence d'une projection de  $G$  dans  $H$  montre que la connaissance représentée dans  $G$  est contenue dans  $H$ . [Carloni, 2006].

Nous basons sur cette opération étendue aux graphes emboîtés pour opérationnaliser le modèle de partage de CMS.

La définition de Projection entre graphes emboîtés est empruntée à SOWA : [SOWA, 1984] Une projection d'un graphe emboîté  $H = (RH, CH, UH, \text{étiq}H)$  dans un graphe emboîté  $G = (RG, CG, UG, \text{étiq}G)$  est un couple d'applications  $\Pi = (f, g)$ ,  $f : RH \rightarrow RG$ ,  $g : CH \rightarrow CG$ , qui :

1. conserve les arêtes et la numérotation des arêtes :

Pour toute arête  $rc$  de  $UH$ ,  $f(r)g(c)$  est une arête de  $UG$ .

2. peut "restreindre" les étiquettes des sommets

2.1. Pour tout sommet  $r$  de  $RH$ ,  $\text{étiq}G(f(r)) \leq \text{étiq}H(r)$

2.2. Pour tout sommet  $c$  de  $CH$ , soit  $\text{étiq}H(c) = (t, m, d)$ .  $\text{étiq}G(g(c)) = (t', m', d')$

vérifie : -  $(t', m') \leq (t, m)$ , et

- si  $d$  est un graphe, alors  $d'$  est un graphe tel qu'il existe une projection de  $d$  dans  $d'$  ( $d$  et  $d'$  sont des graphes emboîtés de profondeurs strictement plus petites que celles de  $G$  et  $G'$ ).

- Si  $d = **$ , il n'y a pas de contrainte sur  $d'$ , qui peut être  $**$  ou un graphe quelconque.

Ainsi, pour opérationnaliser l'appariement sémantique d'un contexte de projet en cours et d'un contexte de CMS.

Nous représentons :

- Un contexte de projet en cours sous la forme de graphe conceptuel emboîté : formé d'un graphe situation d'ingénierie, un graphe conceptuel conditions d'emploi et un graphe conceptuel but.
- Un contexte de CMS : formé d'un graphe situation d'ingénierie, un graphe conceptuel conditions d'emploi et un graphe conceptuel but

L'appariement sémantique consiste à chercher une Projection de graphes emboîtés.

- Vérifier l'existence d'une projection des graphes conceptuels but : les buts d'ingénierie sont représentés par un sommet concept de type ayant comme étiquette réifier ou détailler ou décomposer. La projection garantie que la CMS et le contexte de projet ont le même but.
- Vérifier l'existence d'une projection des graphes conceptuels choix d'ingénierie : le choix d'ingénierie d'un contexte de projet est contenu dans le graphe choix d'ingénierie de CMS avec possibilité de spécialisation des choix
- Vérifier l'existence d'une projection des graphes conceptuels conditions d'emploi : avec la définition de types de concepts définis de type capacité de système, la projection est vérifiée si une CMS est décrite par un paramètre de capacité supérieure ou égale à la situation en cours.

Nous notons  $R_{CMS}$  l'ensemble des CMS résultats d'une opération de projection.

Cet ensemble est organisé sous la forme d'un treillis de CMS où la borne supérieure correspond à l'appariement le plus approprié d'un contexte de CMS et d'un contexte de projet et la borne inférieure correspond à l'inexistence d'un appariement.

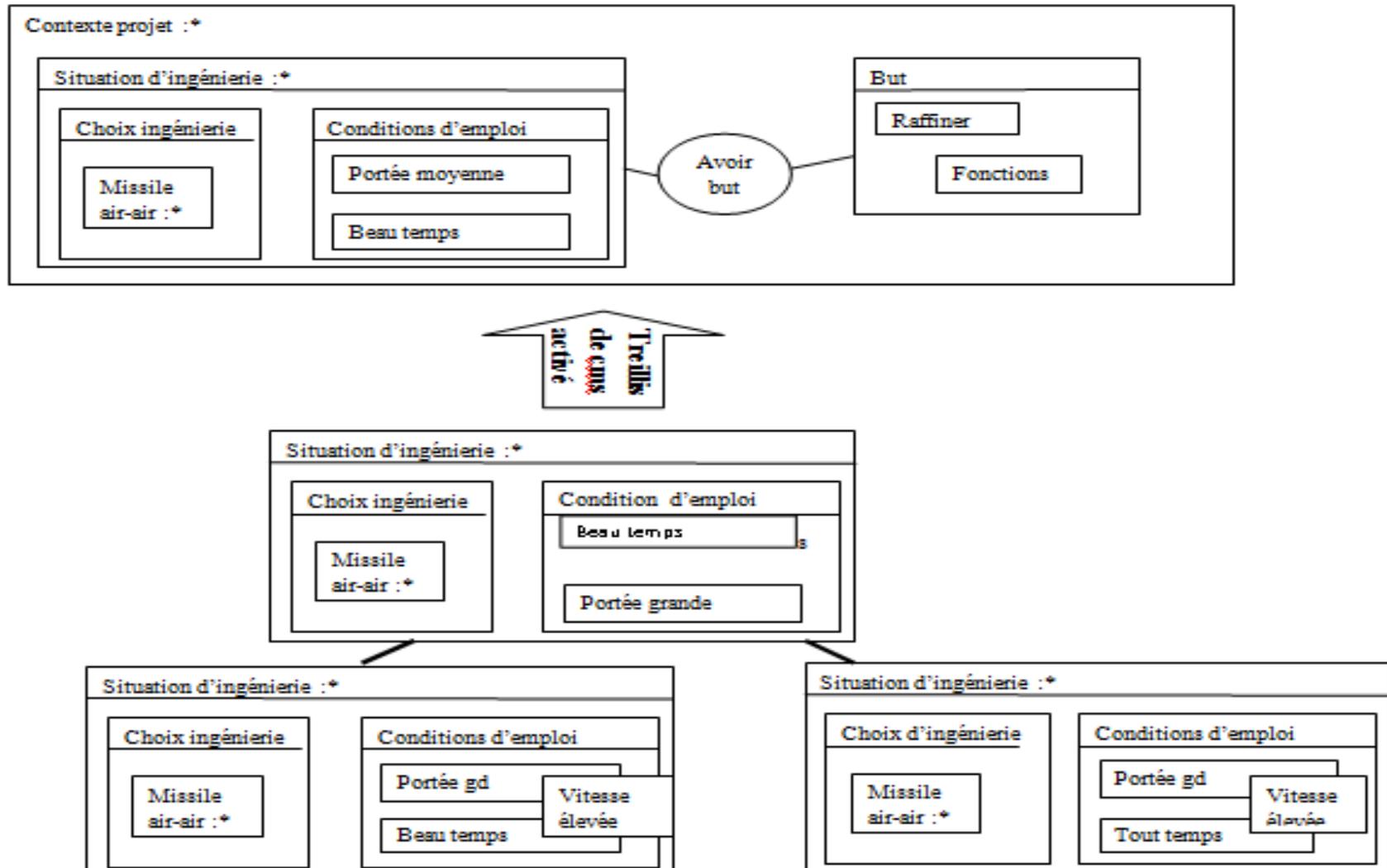
L'ensemble  $R_{CMS}$  muni de la relation d'ordre  $\leq$  est un treillis

- Soit  $CMS1$  et  $CMS2 \in R_{CMS}$ ,
- La borne supérieure est représentée par le plus petit graphe conceptuel se projetant dans  $CMS1$  et  $CMS2$ .
- La borne inférieure correspond au graphe conceptuel cible d'une projection de  $CMS1$  et  $CMS2$ .

La figure V.1.12 présente un exemple d'appariement sémantique, avec un résultat organisé en treillis. Le contexte du projet en cours se projette dans trois contextes de connaissances métiers situées.

L'organisation du résultat est interprétable de la manière suivante :

- La borne supérieure représente le résultat le plus proche de la situation de projet. Dans l'exemple il s'agit du même nombre de concepts dans les deux graphes contexte projet et contexte CMS avec un concept spécialisé de type capacité système : la solution proposée opère dans une grande portée qui est sous type du concept moyenne portée dans le treillis étendu du support des Graphes conceptuel
- Les deux autres CMS appartenant au résultat sont cibles d'une projection de la borne supérieure du treillis en présentant des contextes enrichis par un troisième concept vitesse élevée.



La figure VI.12 : résultat d'appariement sémantique

## VI.4. Architecture d'un Système de Gestion de CMS

Plusieurs plateformes et outils d'implémentation des graphes conceptuels ont été proposés [Gandon, 2008] ; nous présentons dans cette section un outil offrant une implémentation graphique de ce formalisme que nous avons utilisé pour expérimenter la capitalisation et le partage de CMS.

- CoGUI<sup>58</sup> : est un environnement complet pour l'implémentation des graphes conceptuels : il permet l'édition d'ontologies, des différents modèles de graphes, et dispose de fonctionnalités de vérification et validation des objets construits : certaines des propriétés à vérifier sont directement obtenues "par construction" ; pour les autres, une liste de messages d'erreurs et d'avertissements couplée à une visualisation graphique sur le graphe de la zone concernée facilitent la correction des problèmes. De plus, CoGUI permet la mise en œuvre des différents mécanismes de raisonnements associés au modèle par un accès au serveur de raisonnement CoGITaNT.
- CoGITaNT<sup>59</sup> est une API fournissant des composants logiciels pour développer en C++ des moteurs d'inférence compatibles avec la famille SG. Elle fournit aussi un moteur d'inférences prêt à l'emploi sous forme d'un serveur TCP/IP auquel on adresse des requêtes dans un format spécifique XML, le CoGXML. Ainsi un client peut être programmé en un tout autre langage et bénéficier des services inférentielles du serveur CoGITaNT

Ces deux plateformes sont utilisées pour implémenter le modèle d'ontologie, le modèle de capitalisation et le modèle de partage de CMS. Nous souhaiterions intégrer le système de gestion de connaissances métiers située sous forme d'un Plugin à un atelier d'IS.

L'architecture du système de gestion de connaissances métiers que nous proposons est organisée en deux couches :

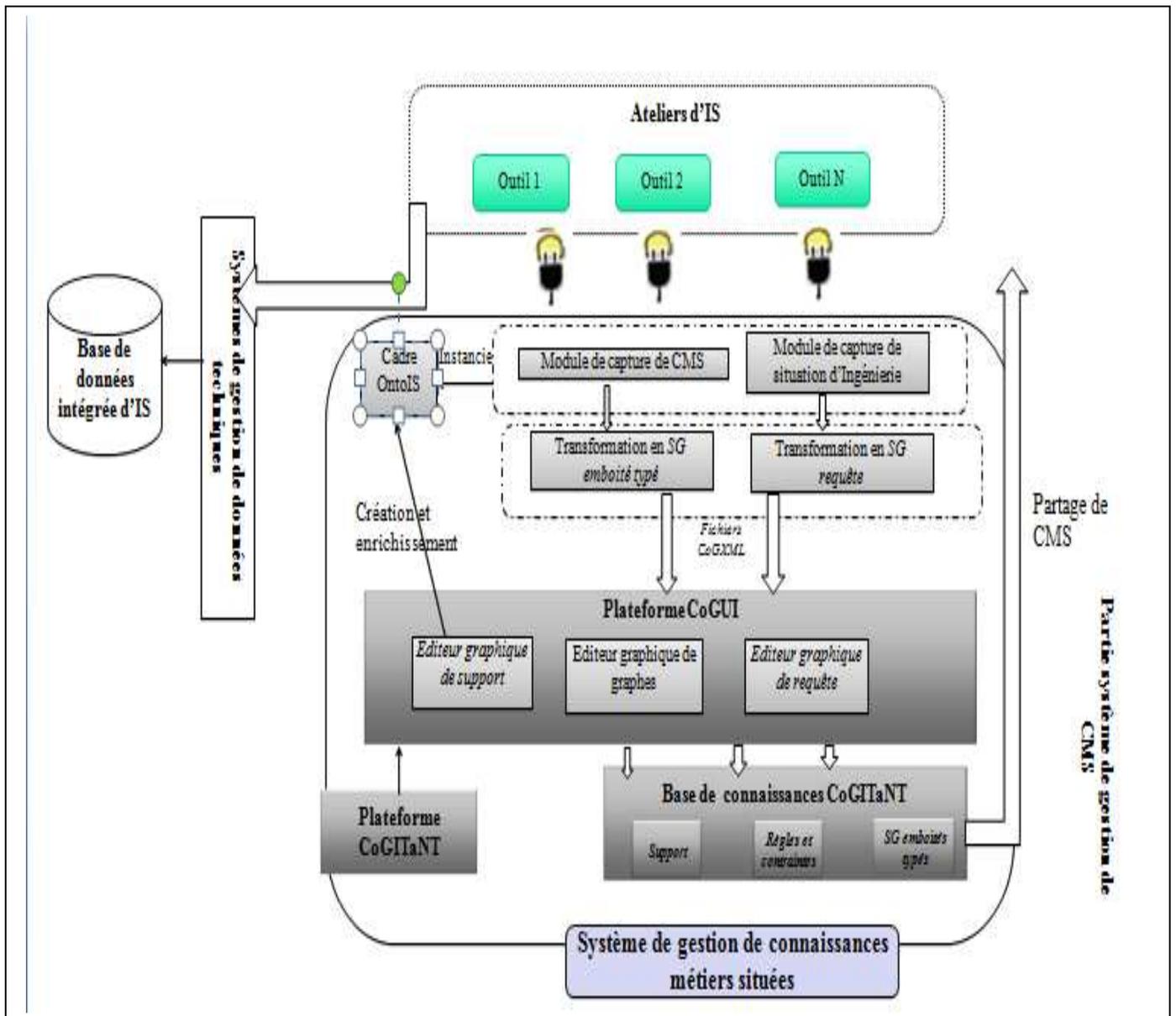
- Une couche dédiée à la capture des informations contextuelles à partir d'un atelier d'IS : la proposition de gestion de connaissances métiers est destinée à être intégrée dans un atelier d'IS. Pour supporter l'ingénieur système en cours de projet, un premier module permet de capturer chaque choix d'ingénierie et la description de contexte de projets sous forme d'instances de primitives ontologiques liées à aux dimensions système à faire et système pour faire. Un deuxième module transforme les informations contextuelles en graphes conceptuels emboîté conformément au modèle de capitalisation de CMS. Le résultat est exprimé sous forme de fichier CoGXML.

---

<sup>58</sup> <http://www.lirmm.fr/cogui/>

<sup>59</sup> <http://www.lirmm.fr/cogitant>

- Une couche dédiée à la gestion des CMS composée de :
  - Editeur d'ontologie : module de création d'un support de graphes conceptuels dans COGUI
  - Un module d'édition de graphes conceptuels : offrant un moyen de visualisation et de vérification des CMS formalisée en CoGXML
  - Un module de raisonnement : basé sur le serveur de raisonnement CoGITaNT pour calculer les projections entre les situations de projets et les situations de CMS



VI.13 Architecture d'un système de gestion de Connaissances Métier situées

## **VI.5 Conclusion**

Les graphes conceptuels et la famille de SG du laboratoire LIRMM offrent un bon compromis entre visualisation de raisonnement (sous la forme graphique par des opérations de graphes) et expressivité dans la représentation des éléments de notre modèle de capitalisation de CMS. De plus la disponibilité d'environnements d'opérationnalisation open source comme COGITANT permet d'étendre et d'adapter les opérations des graphes conceptuel i.e. la projection et les opérations de modélisation de support de graphes en fonction des besoins du scénario d'usage.

# Chapitre VII

## Etude de cas et expérimentation

### *Cas d'un Système de Défense : Ingénierie d'un Missile*

---

#### Plan du chapitre

---

VII.1 Introduction

VII.2 Ingénierie Ontologique d'un Missile

    VII.2.1 Ontologie de la facette contextuelle

    VII.2.2 Ontologie de la facette fonctionnelle

    VII.2.3 Ontologie de la facette organique

VII.3 Scénarios de Capitalisation et Partage de Connaissances Métiers Situées

    VII.3.1 Scénarios de capitalisation de CMS

    VII.3.2 Scénario de partage de CMS

VII.5 Conclusion

## VII.1 Introduction

Ce chapitre pour objectif de valider les propositions conceptuelles de notre thèse : le cadre ontologique OntoIS et la proposition de capitalisation et de partage de Connaissances Métiers Situées. Nous présentons une étude de cas d'un système de défense : Ingénierie d'un missile. Cette étude de cas a été menée en collaboration avec un expert du métier de la société MATRA.

Bien que ce le domaine du missile soit controversé politiquement et scientifiquement, nous l'avons choisi parce que qu'il met en œuvre de multiples connaissances théoriques (i.e. aérodynamique) et métiers. Il présente ainsi un environnement riche en concepts et relations servant d'une bonne illustration à l'ingénierie ontologique.

Ce chapitre est organisé en deux parties :

- Application du cadre ontologique OntoIS (chapitre IV) : par spécialisation des patrons ontologiques noyaux de la facette *système à faire*.
- Application de la proposition de capitalisation et de partage de CMS : nous avons considéré un scénario détaillé de capitalisation et de partage de CMS capturant des choix et des décisions des phases d'analyse fonctionnelle et de synthèse d'un système de missile.

Ces deux applications ont été implémentées sur la plateforme de gestion des Graphes Conceptuels COGUI<sup>60</sup>. L'architecture générale du système de gestion de CMS présentée dans le chapitre VI est restreinte à la partie modélisation, capitalisation et partage de CMS, faisant l'hypothèse que les informations contextuelles de projets ont été préalablement collectées à partir d'un atelier d'ingénierie systèmes.

Ainsi dans une première phase, nous avons créé les trois composants ontologiques des facettes contextuelle, fonctionnelle et organique moyennant l'interface graphique de CoGUI permettant la construction de support de Graphes Conceptuels. Cette interface dispose de fonctionnalités de vérification et de validation des primitives ontologiques.

Dans une deuxième phase, nous avons défini un patron de capitalisation de CMS sous la forme de graphes emboîtées typées capturant les facettes situationnelle, téléologique, argumentative et décisionnelle. Ce patron a été instancié pour peupler la base de connaissance assertionnelle de la plateforme COGUI.

Dans une troisième phase nous avons requêté cette base de connaissances en considérant des variantes de scénarios modélisant des contextes de projets d'IS. La plateforme COGUI propose une interface graphique de soumission de requêtes sous forme de Graphes Conceptuels et permet de visualiser les résultats de projections sur la base de connaissances assertionnelle.

---

<sup>60</sup> <http://www.lirmm.fr/cogui/>

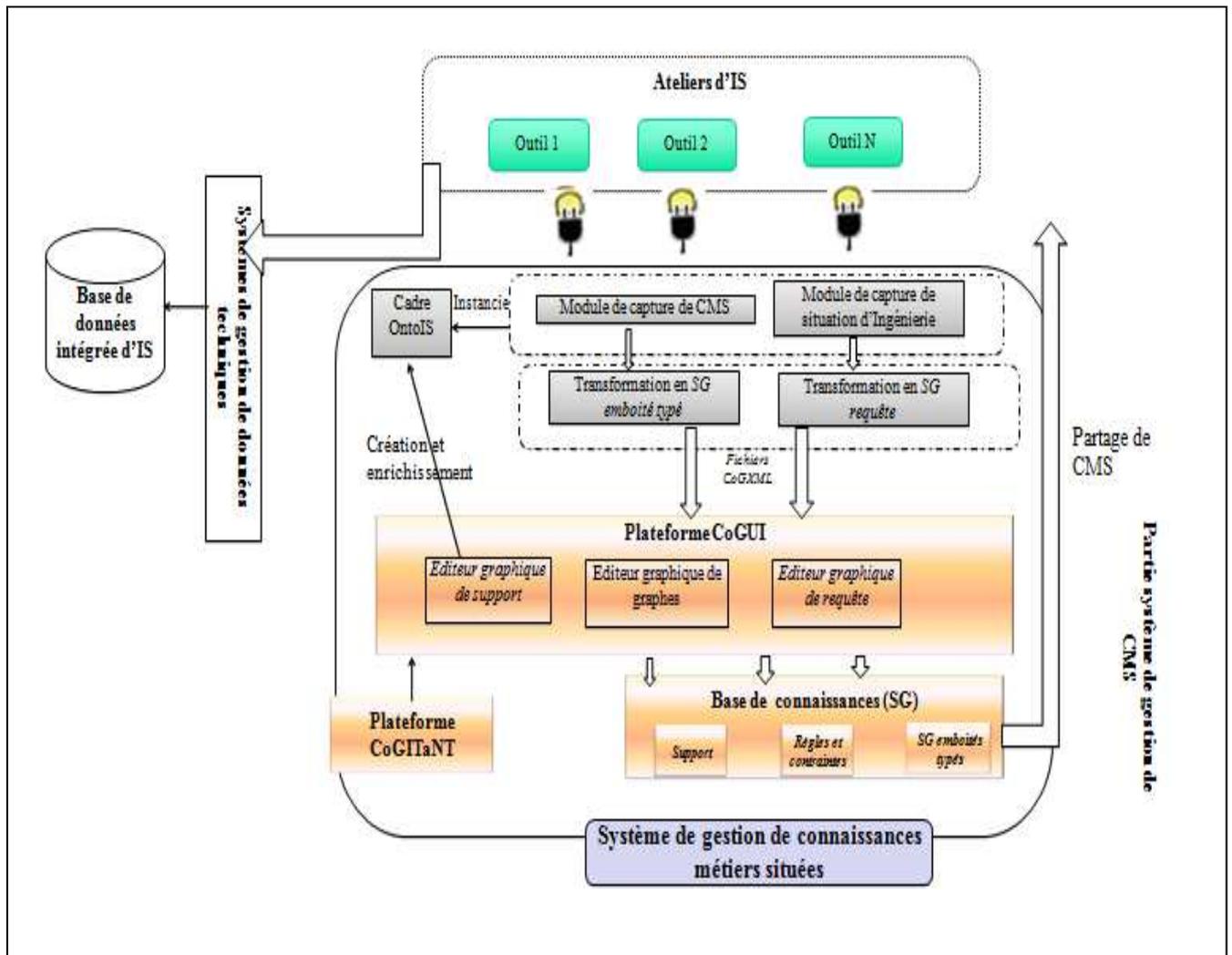


Figure VII.1 Plateforme utilisée pour l'expérimentation de la proposition ( en orange )

## VII.2 Ingénierie Ontologique d'un Missile

Dans cette section nous évaluons notre proposition de composants ontologiques noyaux de la facette *système à faire* un missile. Les composants ontologiques applicatifs ont été élaborés avec l'assistance d'un expert du métier des missiles. Nous nous sommes aussi basé sur de multiples ressources<sup>6162</sup>, dont [Durak, 2006][ Davis, 2004] décrivant les systèmes et les sous systèmes de missiles.

<sup>61</sup> [www.netmarine.net](http://www.netmarine.net)

<sup>62</sup> [www.wikipedia.fr](http://www.wikipedia.fr)

Dans ce qui suit nous considérons détaillons l'application sur trois facettes du *système à faire* : la facette contextuelle, la facette fonctionnelle et la facette organique.

Pour chaque facette nous présentons les résultats des phases de conceptualisation, de l'ontologisation et de l'opérationnalisation.

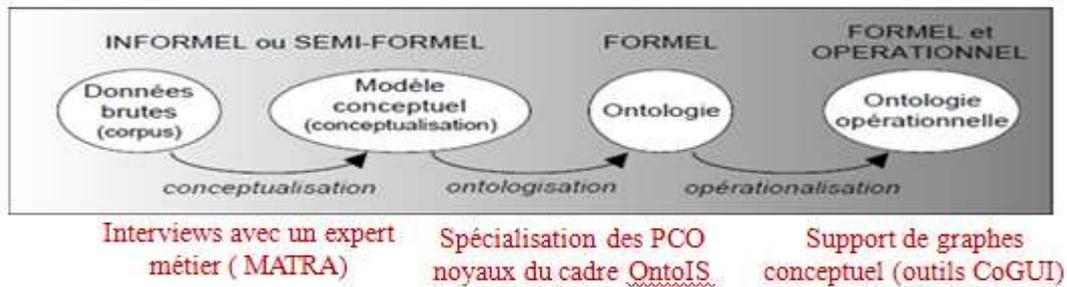


Figure VII.2 : processus d'Ingénierie Ontologique adopté

## VII.2.1 Ontologie de la facette contextuelle

Cette ontologie a pour objectif de systématiser les connaissances métiers considérées dans le processus d'analyse des exigences d'un missile.

Pour conceptualiser les connaissances contextuelles, nous avons commencé par analyser un système de missile selon une vue externe [AFIS, 2009] pour identifier les concepts décrivant ce système, son environnement ainsi que leurs interactions.

Selon l'encyclopédie l'internaute<sup>63</sup> un missile est défini comme « *un engin explosif autopropulsé et guidé* ». Où engin désigne un projectile i.e. un corps lancé ou projeté pour atteindre une cible. Ainsi les deux sous systèmes principaux d'un missile sont le système de guidage et le système de propulsion.

Les missiles peuvent être catégorisés en fonction de nombreux critères parmi lesquels nous citons<sup>64</sup> :

- Selon le profil de mission : désignant d'une part le type de plate-forme de tir et le type de l'objectif du missile. Par exemple Nous trouvons dans cette catégorie les missiles air-air (attaque d'un avion par un autre avion (combat aérien)), les missiles mer-mer (attaque d'un navire par un autre navire)

<sup>63</sup> <http://www.linternaute.com/dictionnaire/fr/definition/missile/>

<sup>64</sup> [www.wikipedia.fr](http://www.wikipedia.fr)

- Selon le type de cible : il s'agit d'une classification dépendante uniquement de la cible, sont distingués ainsi par exemple les missiles anti-CHAR, les missiles anti-aériens destinés à détruire un aéronef (avion ou hélicoptère)
- Selon la portée : où sont distingués les missiles de courte portée (quelques kilomètres), de moyenne portée (dizaines de kilomètres) de longue portée (centaines de kilomètres).
- Selon le type de vol : dans cette catégorie nous trouvons essentiellement les missiles balistiques et les missiles de croisière.

Ces différentes catégorisations se regroupent partiellement et rendent une classification des missiles relativement complexe. Pour rendre compte de ces multiples points de vue nous avons opté pour une modélisation des différentes composantes de l'environnement d'un missile et de ses relations d'interactions avec son environnement. Nous obtenons ainsi une couverture des différentes typologies de ces systèmes.

Nous représentons dans l'ontologie de la facette contextuelle, des exemples types de paramètres caractérisant un missile. Nous mettons l'accent sur les paramètres de type capacité utilisés pour étendre l'appariement des contextes de projets avec les CMS.

Ces paramètres incluent par exemple la manœuvrabilité d'une cible, la vitesse, la puissance de destruction d'un missile et la couverture nuageuse rattachée au concept de conditions météorologiques caractérisant l'environnement opératoire du missile. La figure VII.3 présente un extrait de l'ontologie contextuelle en mettant en évidence quelques relations de spécialisation du patron ontologique noyau du cadre OntoIS.

Fragment du  
PCO contextuel  
OntoIS

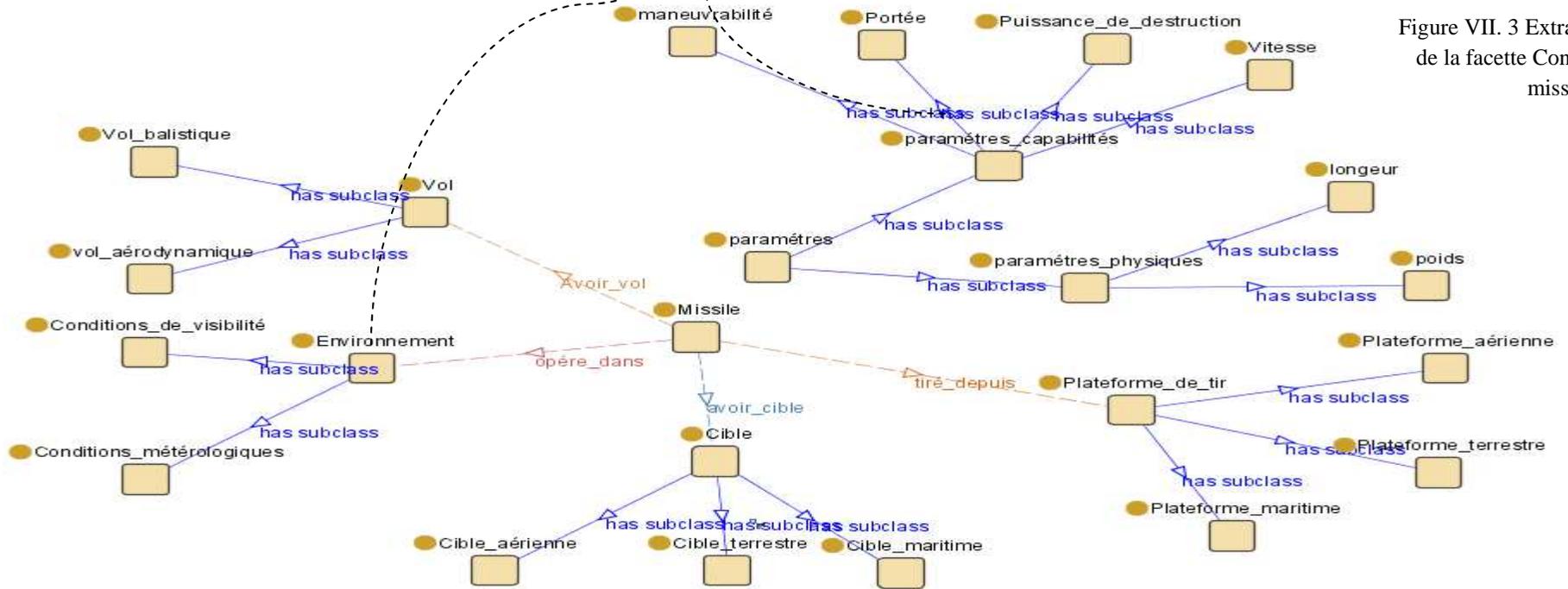
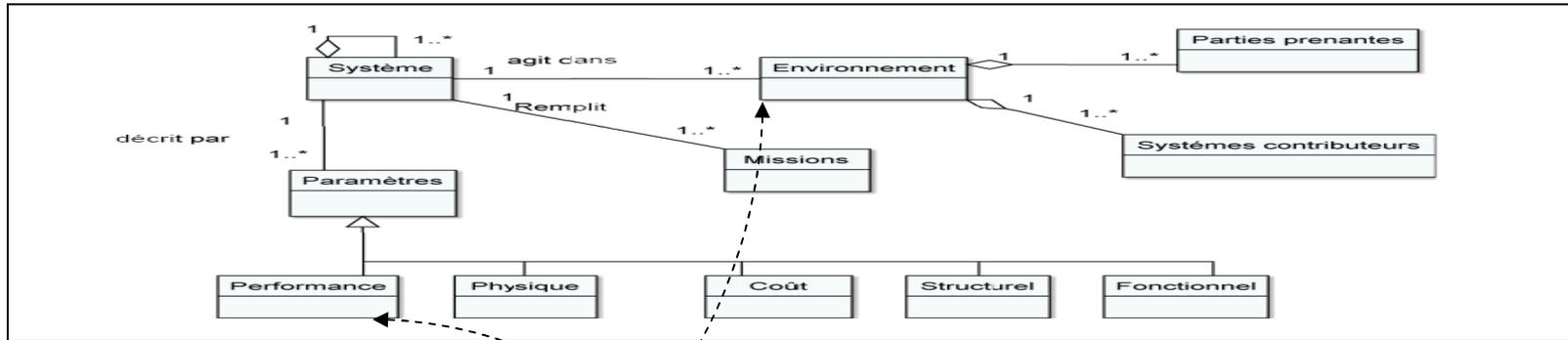
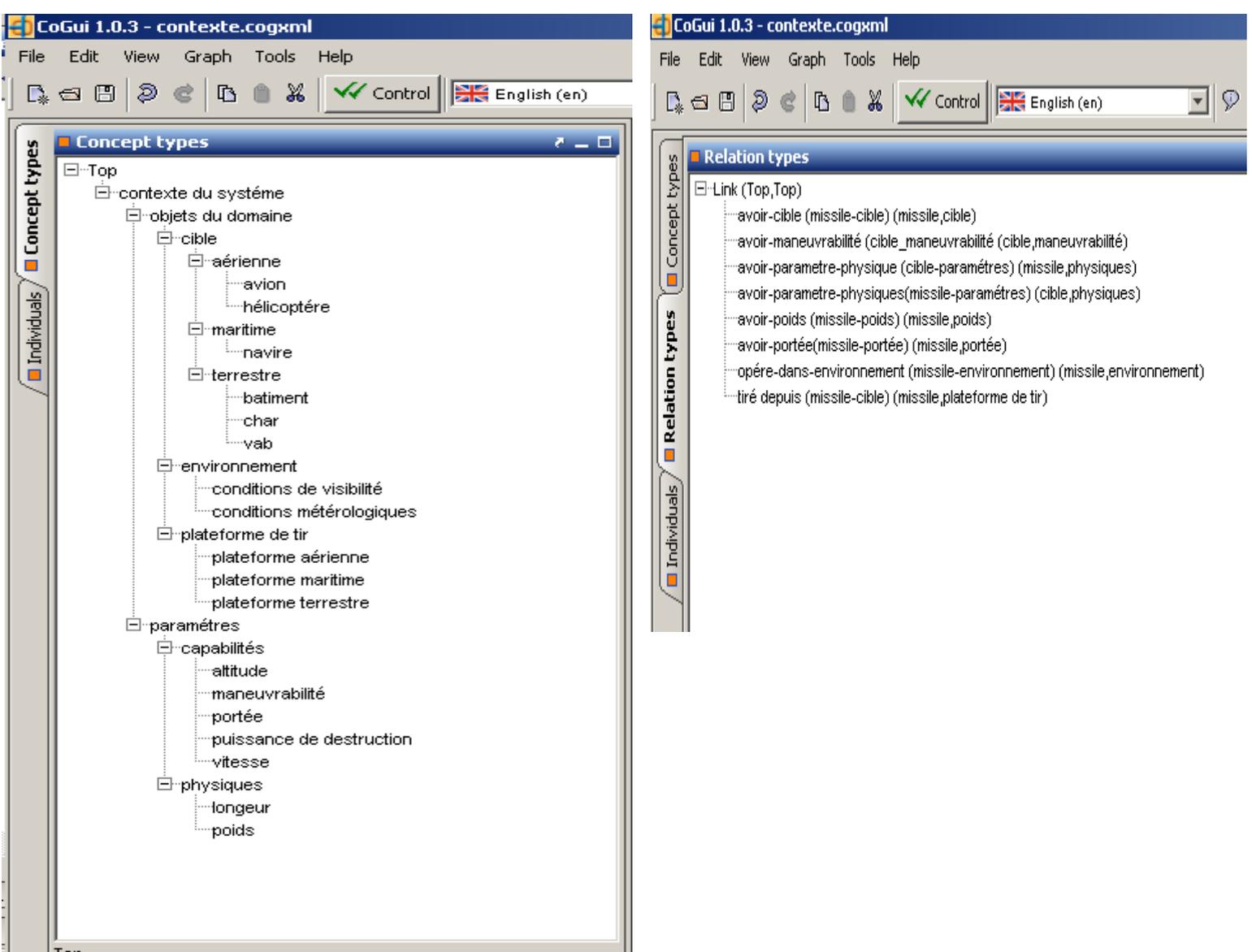


Figure VII. 3 Extrait de l'ontologie de la facette Contextuelle d'un missile

- **Opérationnalisation** : Cette ontologie contextuelle a été opérationnalisée sous la forme de support de graphes conceptuels dans l'atelier COGUI. la figure VII.4 présente un extrait du treillis de concepts et du treillis de relation.



VII.4 support de graphes conceptuels : facette contextuelle

## VII.2.2 Ontologie de la facette fonctionnelle

La conceptualisation de l'ontologie fonctionnelle a été réalisée en collaboration avec un expert du métier de l'ingénierie des missiles de l'entreprise Matra.

Dans une première étape quatre fonctions de base d'un missile ont été dégagées à savoir : *se diriger, voler, propulser et détruire une cible*. Puis nous avons défini une spécialisation de ces fonctions selon le mode de guidage des missiles. Le mode de guidage d'un missile permet de définir une chaîne fonctionnelle spécifique. Les deux principaux modes de guidage d'un missile sont *l'autoguidage et le téléguidage*. Pour un missile *autoguidé* les ordres de guidage et de pilotage sont élaborés à bord du missile qui, une fois parti de son poste de lancement, est complètement autonome pour atteindre l'objectif qui lui a été désigné, préalablement à son lancement. Dans ce cas le missile doit aussi offrir une fonction de capture (allouée à un composant : autodirecteur). Pour un missile *téléguidé* les ordres de guidage sont élaborés par la plateforme de tir qui lui indique la cible tout au long de sa trajectoire. Le tableau

Ainsi, une chaîne fonctionnelle minimale pour un missile *autoguidé* :

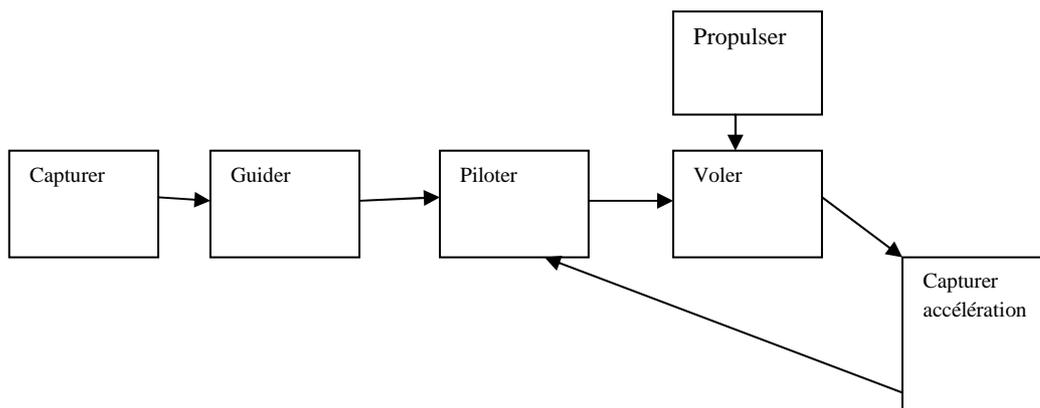


Figure VII. 5 Chaîne fonctionnelle de l'un missile autoguidé

Fonctions	Description
Voler	la fonction voler est spécialisée généralement en fonction du type de voilure du missile. Nous distinguons ainsi la fonction voler en voilure fixe et voler en voilure mobile (canard par exemple)
Diriger	Cette fonction est spécialisée selon le mode de guidage : autoguidé et téléguidé. Chaque mode de guidage se décline en guidage direct ou indirect
Capturer	Cette fonction apparaît pour les missiles autoguidés et se spécialise en capturer position cible et capturer position missile. La capture de la position de la cible est soit une capture de la vitesse angulaire (état point), capture d'écartomètre (epsilon), ou capture de distance du missile-cible.
Piloter	Le pilotage se spécialise en un pilotage selon la position ou selon la vitesse du missile
Propulser	Deux principaux types de propulsions sont distingués : les réacteurs utilisant l'air ambiant comme carburant et les moteurs fusés ou statoréacteur à carburant po <sup>1</sup> <b>PCO</b>
<b>Flux fonctionnels</b>	Nous avons un flux fonctionnel de type informationnel : la quantité fonctionnel les fonctions capture-position missile et diriger. La quantité angulaire se spécialise en état point ou epsilon

Tableau VII. 1 Conceptualisation des fonctions d'un missile

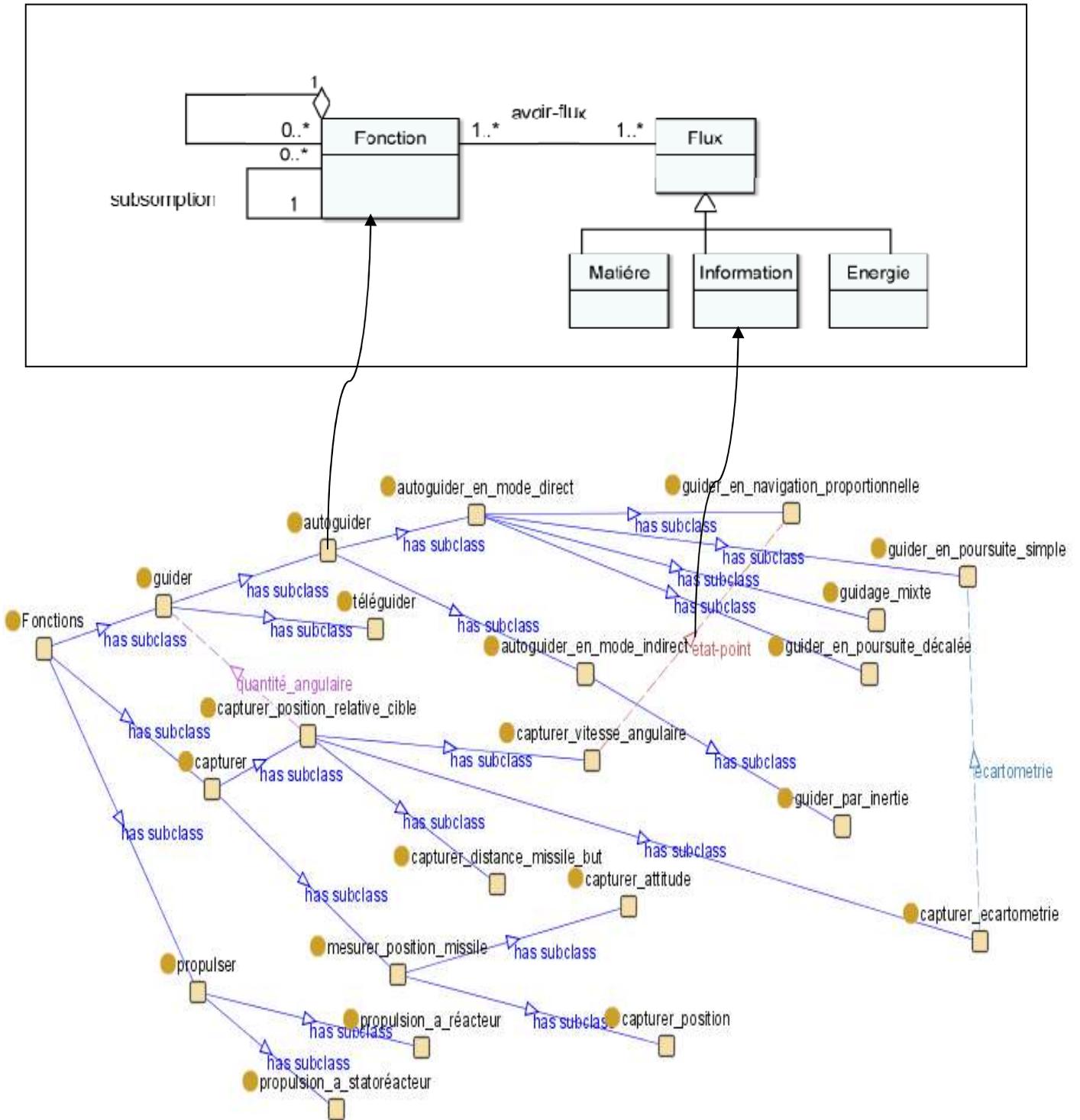
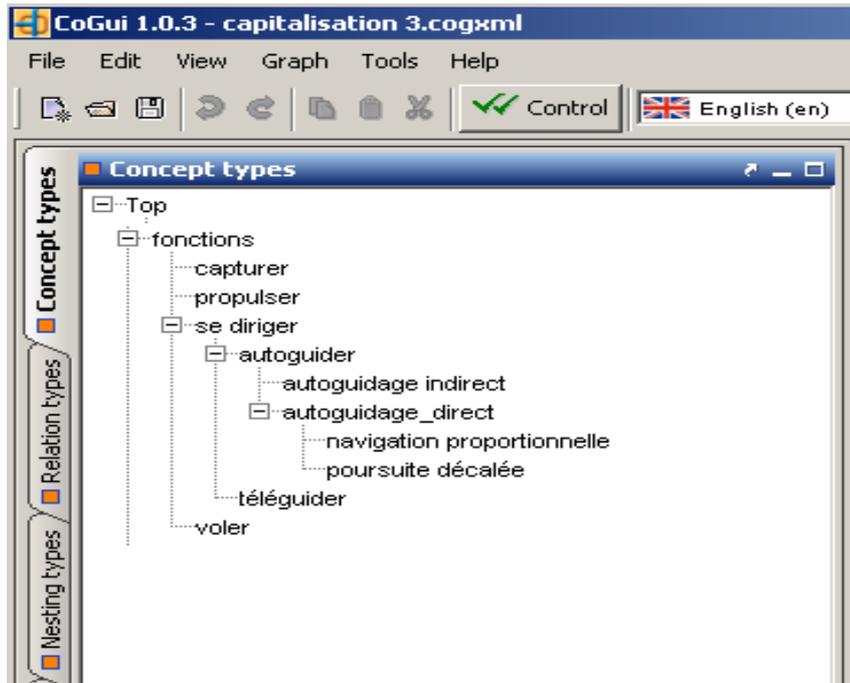


Figure VII. 6 Extrait de l'Ontologie de la facette fonctionnelle

Cette conceptualisation de la facette fonctionnelle à fait l'objet d'une étape d'ontologisation (figure VIII.6) par spécialisation du PCO noyau de la facette fonctionnelle puis l'objet d'une étape d'opérationnalisation sous forme de treillis de concepts et de treillis de relations implémentés dans l'atelier COGUI.(figure VII.7)



VII.7 treillis de concepts fonctionnel

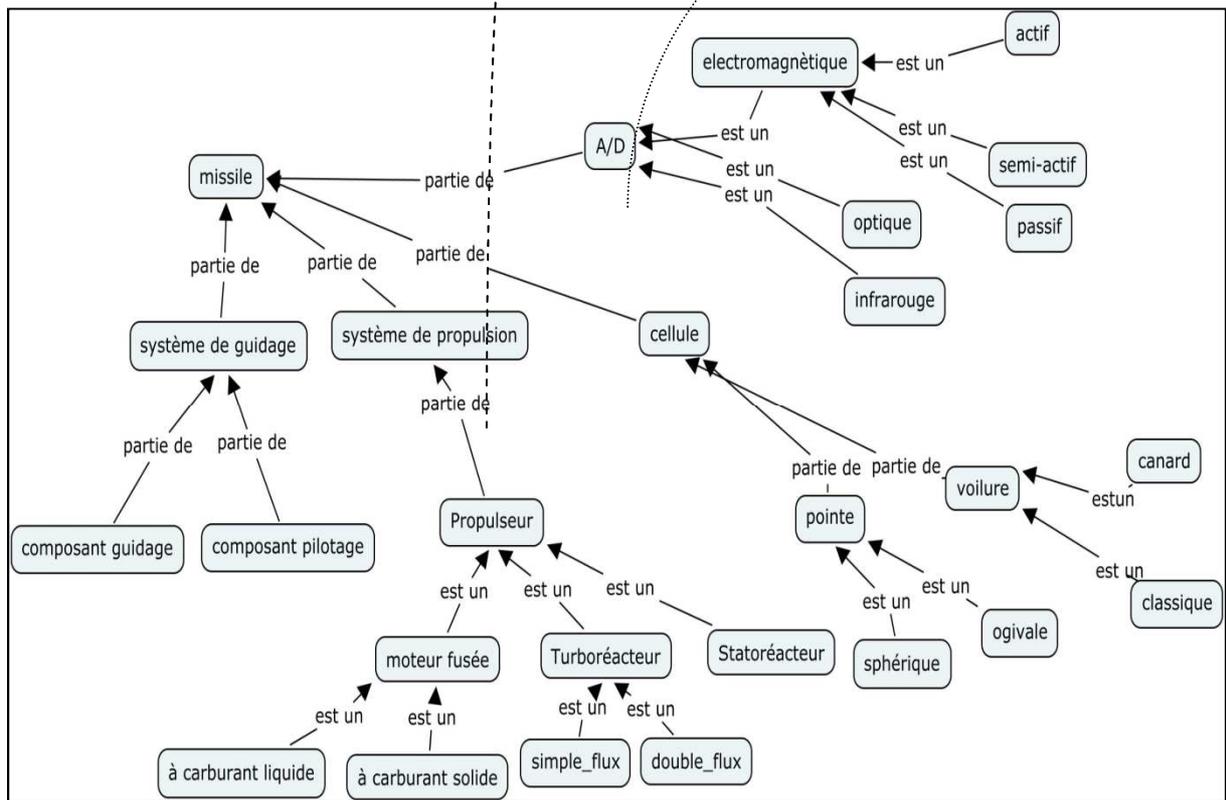
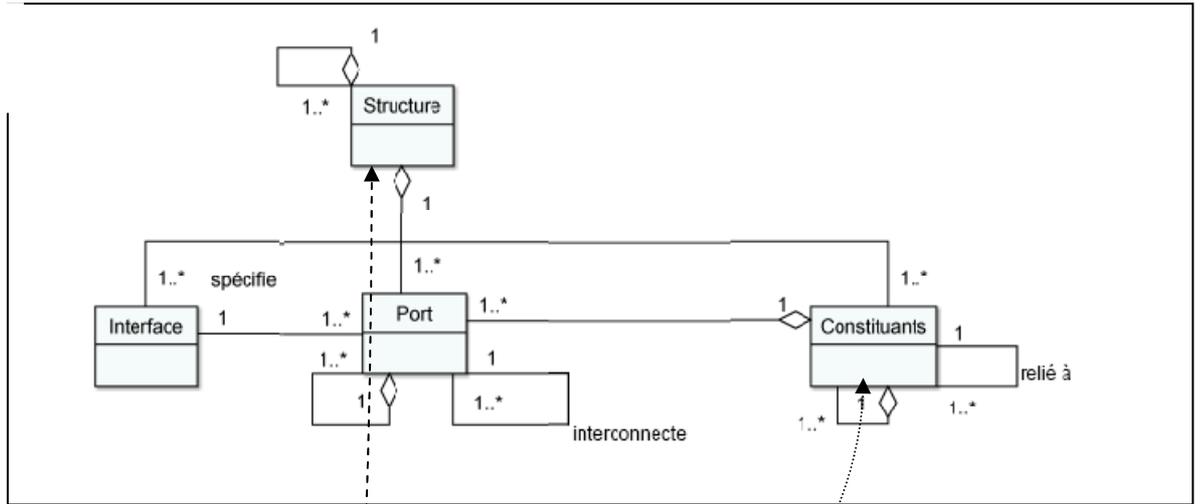
### VII.2.3 Ontologie de la facette organique

L'ontologie de la facette organique a pour objectif de systématiser les concepts de constituants physiques et logiciel ainsi que leurs relations structurelles et d'interaction.

Le tableau VIII. 2 synthétise les principaux concepts de cette facette. Les relations que nous avons considérées sont les relations de subsomption et de composition.

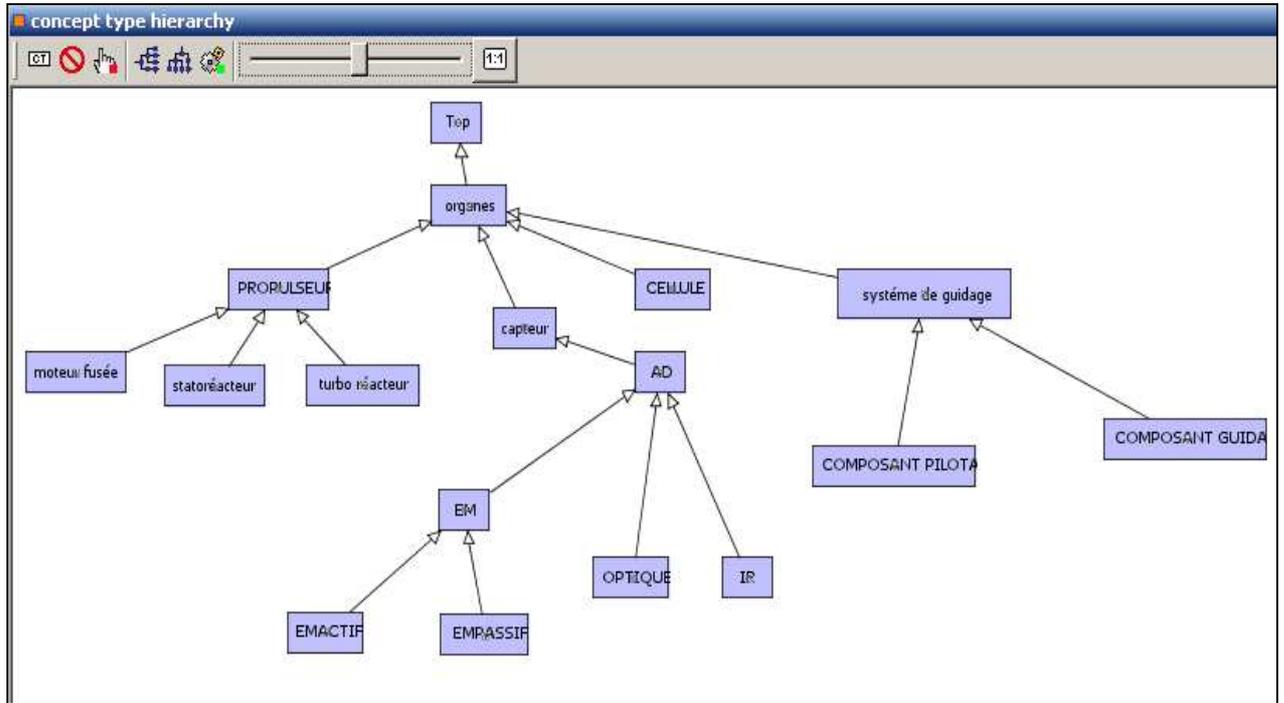
<b>Organes</b>	<b>Description</b>
Système de guidage	Le système de guidage conduit le missile balistique vers sa cible. Il est constitué de deux sous systèmes logiciels : composant de guidage et composant de pilotage
Système de propulsion	Le système de propulsion est l'élément qui envoie le missile balistique vers sa cible. Selon la portée du missile, le système de propulsion peut comprendre plusieurs étages.
Cellule	Allouée à la fonction voler du missile
autodirecteur	Composant d'un missile destiné à capter un signal provenant de la cible à atteindre, ou par un autre dispositif (i.e.avion, radar) pour fournir les informations permettant d'orienter le missile vers sa cible

PCO organique



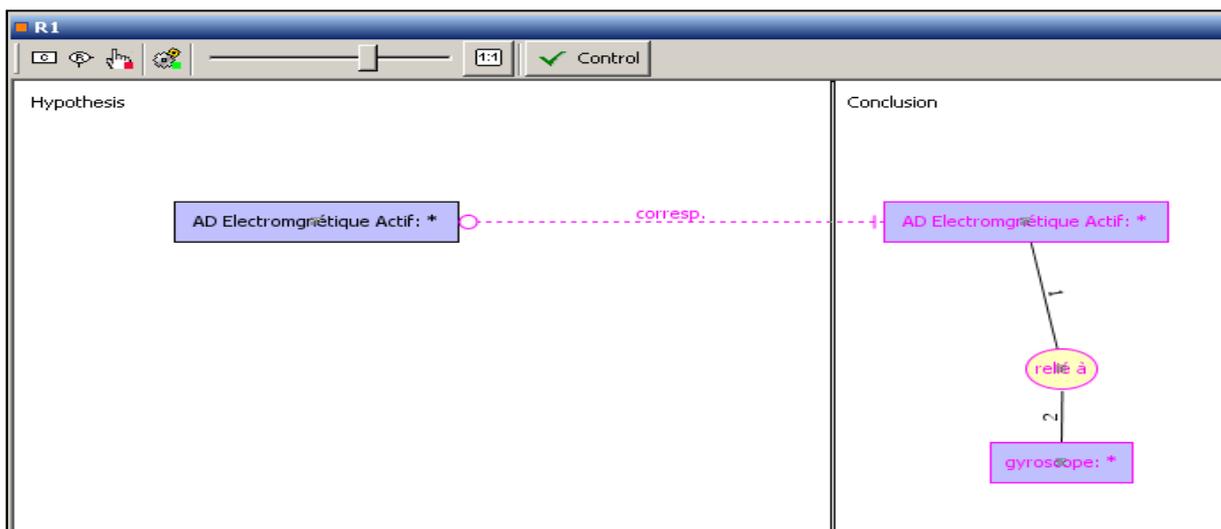
VII.8 Ontologie de la facette organique d'un missile

## Opérationnalisation



Treillis de types de concepts

**Exemple d'axiome de domaine :** exprimant une contrainte de type nécessité : un autodirecteur électromagnétique actif nécessite un constituant gyroscope.



règle SG

## VII.3 Scénarios de Capitalisation et de partage de CMS

### VII.3.1 Scénarios de capitalisation de CMS

Ces scénarios de capitalisation ont été capturés durant les processus d'analyse fonctionnelle et de synthèse d'un système de missile, pour répondre aux exigences décrites dans le tableau VII.9 .

exigence du scnero.xlsx								
	A	B	C	D	E	F	G	H
1								
2		<b>caratéristique système</b>	<b>valeur</b>	<b>type de données</b>	<b>unité</b>			
3								
4		<b>type du système</b>						
5		<b>missile air-air</b>						
6		<b>attributs</b>						
7		<b>autonomie</b>	<b>tire et oublie</b>					
8		<b>vitesse</b>	<b>[1.2, 2.2]</b>	<b>réel</b>			<b>mach</b>	
9		<b>plateforme</b>	<b>avion</b>					
10		<b>cible</b>	<b>avion</b>					
11		<b>attributs</b>						
12		<b>vitesse</b>	<b>[1.2, 2.2]</b>	<b>REEL</b>			<b>m/s</b>	
13		<b>manœuvrabilité</b>	<b>5 REEL</b>				<b>G</b>	
14								
15		<b>Environnement</b>						
16		<b>condition météorologique</b>	<b>tout temps</b>					
17								

VII.9 : Exigences du système considéré

Pour chaque situation de projet donnant lieu à un choix d'ingénierie nous décrivons la situation informellement puis nous présentons le modèle de capitalisation de CMS correspondant.

<p>Présentation informelle</p>	<p>Capitalisation de CMS sous forme de graphe conceptuel emboité</p>
<p>Cette situation concerne l'analyse fonctionnelle d'un sous système de guidage de missile.</p> <p>L'objectif est de raffiner la fonction se diriger en fonction des exigences du système. Il s'agit si le sous système de guidage sera auto guidé ou téléguidé. Le choix d'autoguidage est retenu car le mode de tire ( ou autonomie) est de type tire et oublie (fire and forget )</p>	

<p>Présentation informelle</p>	<p>Capitalisation de CMS sous forme de graphe conceptuel emboîté</p>
<p>L'ingénieur système a choisi la fonction d'autoguidage. L'objectif de cette situation est de détailler les choix fonctionnel. Le champ décision est décrit par un graphe conceptuel formalisant le choix d'une chaîne fonctionnelle d'autoguidage</p>	

<p>Présentation informelle</p>	<p>Capitalisation de CMS sous forme de graphe conceptuel emboîté</p>
<p>Dans cette situation d'ingénierie, une fonction de type autoguidage directe est choisie. L'objectif est de raffiner ce choix d'ingénierie. deux alternatives de solutions sont envisageables : la fonction poursuite décalée et la navigation proportionnelle. La navigation proportionnelle est retenue car la portée du missile est grande et le vol est court (vitesse élevée)</p>	

<p>Présentation informelle</p>	<p>Capitalisation de CMS sous forme de graphe conceptuel emboîté</p>
<p>Cette situation présente un exemple de but d'allocation de la fonction propulsion du missile. L'organe moteur fusée a été retenu car la plateforme de tir est aérienne et la cible est manœuvrante.</p>	



### VII.3.2 Scénario de partage de CMS

Pour expérimenter le partage de CMS. Nous avons défini le contexte d'un projet en cours dans l'éditeur des requêtes de COGUI : la situation d'ingénierie : un missile fonctionnant dans de bonnes conditions météorologique ; but : allocation de la fonction capturer

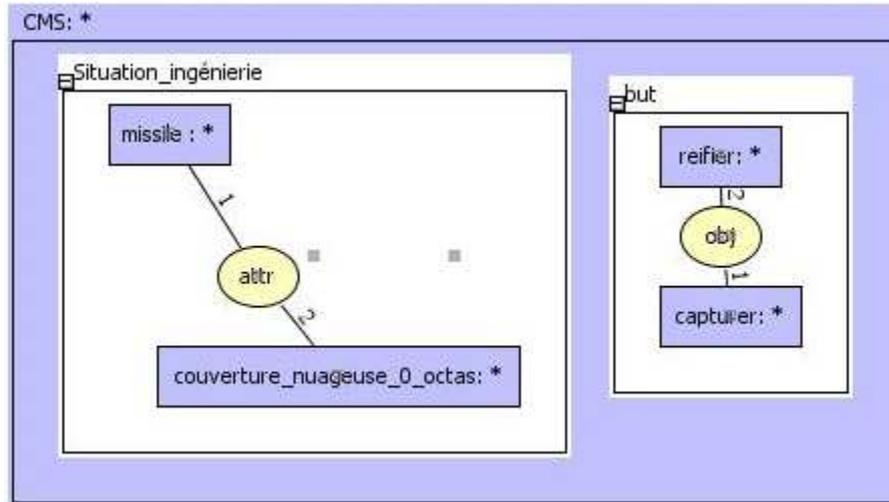
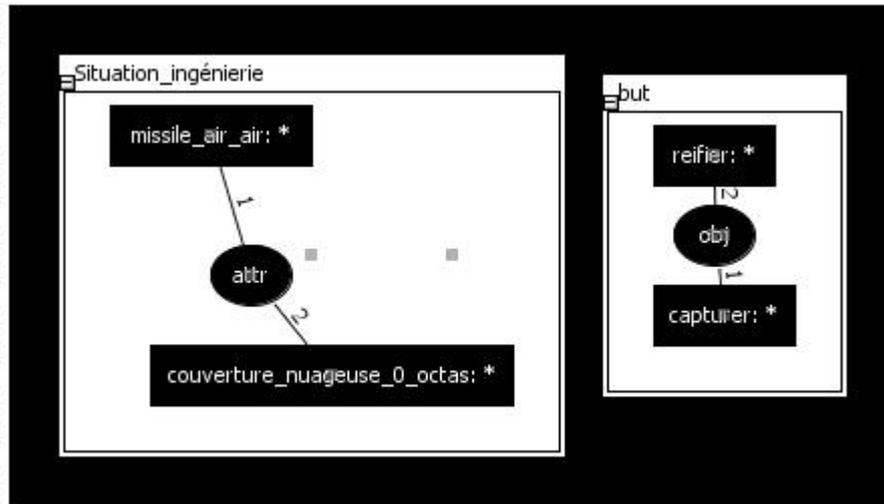


Figure VII.9 un contexte de projet

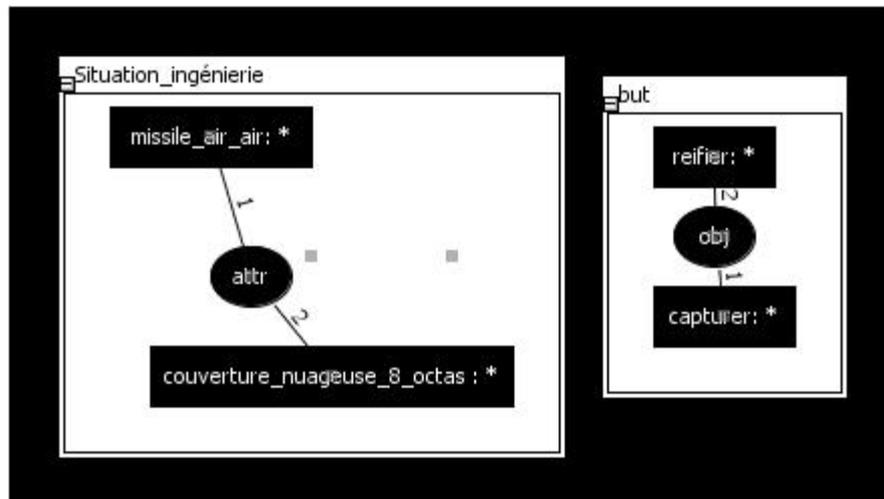
La projection de ce contexte de projet sur la base de connaissances métiers constituée des CMS définies dans la phase de capitalisation génère trois résultats :

In graph	Description
1	projection 1/1 into 1.
2	projection 1/1 into 2.
3	projection 1/1 into 3.

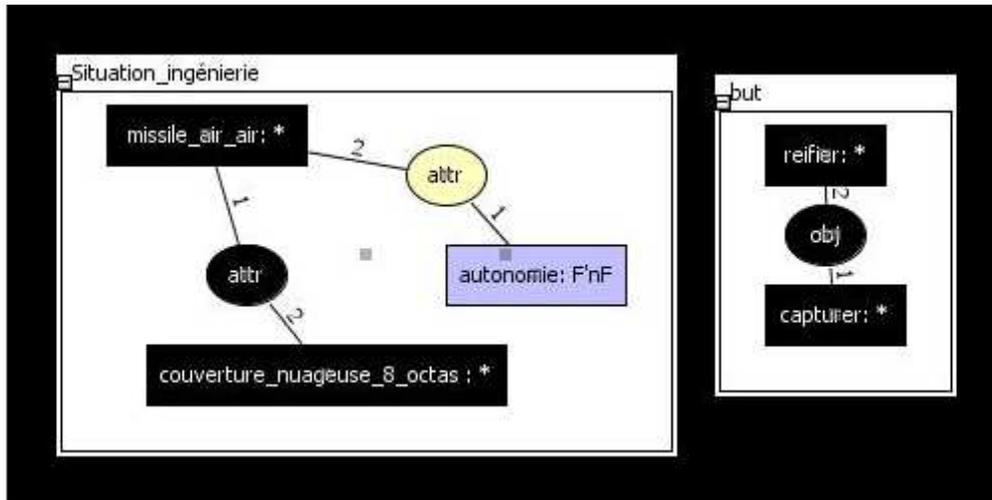
La première solution présente un appariement parfait :



La deuxième solution présente un appariement spécialisant des conditions d'emploi : un choix d'ingénierie valable pour une couverture nuageuse\_8 octas est potentiellement utile pour la situation en cours.



La troisième solution présente un appariement spécialisant les conditions d'emploi et ajoutant une contrainte sur l'autonomie du missile.



## VII.4 Conclusion :

Au travers de cette étude de cas nous illustré la spécialisation de patrons conceptuels ontologiques de la facette système à faire dans le cadre d'une application d'ingénierie de missile. Nous avons utilisés ces patrons ontologiques spécialisés pour annoter sémantiquement des décisions de projet sous forme de graphe conceptuels emboîtés.

L'opérationnalisation du modèle de partage étendu, nécessite une étape d'enrichissement du support de graphes conceptuel par des concepts définis comme le concept couverture\_nuageuse\_8\_octas. Cette extension est possible via une personnalisation des opérations de construction du support de graphes conceptuels de la bibliothèque COGITANT<sup>65</sup>.

<sup>65</sup> [cogitant.sourceforge.net](http://cogitant.sourceforge.net)

# Conclusion générale

L'Ingénierie Système est un domaine complexe et pluridisciplinaire, mettant en œuvre plusieurs catégories de connaissances de résolution de problème. Tout au long de ce travail nous avons tenté de donner des éléments de réponse à la problématique de gestion de connaissances en Ingénierie Système.

En dépit de la prolifération des travaux de recherche sur ce sujet, nous avons remarqué qu'il n'existait pas à ce jour ni une définition consensuelle et précise des connaissances d'IS ni de système de gestion de connaissances en IS. Les propositions analysées sont soit génériques et abstraites donc non appliquées, soit spécifiques, dépendantes d'un domaine d'ingénierie et d'une catégorie précise de connaissances i.e. expérience ou connaissance métier, ou réutilisation d'artefact d'ingénierie.

Nous nous sommes basés sur une combinaison de concepts fondamentaux du domaine de la gestion de connaissances et de l'Ingénierie Ontologique.

Cette combinaison vise à générer une sémantique globale et consensuelle couvrant plusieurs facettes des processus d'IS et explicitant les connaissances métiers tout au long du cycle de vie de ces processus, en vue de rendre cette sémantique partageable et réutilisable par l'ensemble des parties prenantes dans des situations similaires et quel que soit le domaine d'application d'ingénierie.

Afin d'aboutir à un certain consensus en terme de gestion de connaissances en IS, nos travaux se sont situés à l'intersection de deux applications de l'Ingénierie Ontologique afin de :

- Modéliser d'une part les connaissances métiers d'IS : Nous avons proposé un cadre ontologique générique stratifié et multi-facette pour la modélisation de connaissances d'ingénierie de systèmes. Ce cadre ontologique a pour objectif de couvrir les diverses dimensions décrivant les processus d'IS à savoir : la dimension « système à faire » capturant explicitement une conceptualisation du domaine du problème et des exigences, du domaine fonctionnel, et du domaine physique (ou organique (assemblage, composants matériels ou logiciels.) système étudié. La dimension « système pour faire » capturant explicitement le contexte d'un projet d'ingénierie sous la forme de processus, ressources humaines et matérielles, et artefacts d'ingénierie.
- Formaliser des expériences d'ingénierie en nous basons sur des connaissances métiers partagées : dans ce contexte nous avons analysé les apports de l'annotation sémantique comme base de capitalisation et de partage d'expériences d'IS. Ces expériences sont dénommées Connaissances Métiers Situées parce qu'elles émergent dans des situations de résolutions effectives de problèmes d'Ingénierie et tiennent compte du contexte métier, organisationnel et décisionnel des processus d'ingénierie. Pour gérer ces connaissances nous avons proposé deux modèles :
- Un modèle conceptuel de capitalisation de Connaissances Métiers Situées (CMS) : représentant les expériences de projets d'Ingénierie sous forme d'annotations

sémantiques multidimensionnelle basées sur le cadre ontologique que nous proposons. Quatre dimensions d'annotation sont proposées : la dimension situationnelle, la dimension téléologique (but d'ingénierie), la dimension argumentative et la dimension décisionnelle.

- Un modèle de partage de ces connaissances basé sur un mécanisme d'appariement sémantique étendu aux spécificités d'une activité d'IS et permettant de suggérer les CMS potentiellement utiles à un ingénieur système, par opposition à un système d'appariement exacte.

L'opérationnalisation de la proposition générique de gestion de connaissances métiers situées s'est basée sur le formalisme de Graphes Conceptuels, un formalisme combinant les avantages de lisibilité et de puissance de raisonnement.

La proposition a été expérimentée dans le domaine de la défense: Ingénierie de Systèmes de Missiles. Le cadre ontologique générique a été instancié pour systématiser les connaissances métiers du domaine des missiles selon trois facettes : analyse des besoins, analyse fonctionnelle et analyse organique. Un scénario de capitalisation et de partage de Connaissances Métiers Situées a été testé pour illustrer le processus de gestion de connaissances décisionnelles de projets d'IS. Notre approche a été implantée sur une plateforme de gestion de graphes conceptuels, qui exploite l'API CoGITaNT de raisonnement sur ce formalisme et une API CoGUI de manipulation graphique des éléments de ce formalisme.

Le cadre générique proposé s'inscrit dans le contexte « d'une nouvelle génération d'environnements de gestion de connaissances en Ingénierie système »

Nous pensons que ce travail ne constitue qu'un point départ et que les perspectives de recherche de la gestion de connaissances en IS sont diverses et multiples. Nous proposons, à ce stade de notre réflexion, trois perspectives de recherches :

- Etudier des mécanismes de définition de référentiels de bonnes pratiques métiers plus génériques à partir de référentiels de connaissances métiers situées : le référentiel de connaissances métier située constitue une base de partage de choix et de décisions rencontrées en cours de projet. Ce référentiel sera enrichi incrémentalement au fil des projets. Il serait intéressant d'identifier des relations de généralisation de CMS, par exemple en formalisant des contraintes liées à une plage de valeurs de paramètres système sous forme de règles de domaine.
- Etendre les patrons conceptuels ontologiques en étudiant d'autres normes d'IS et en validant l'applicabilité de ces patrons dans divers secteurs d'activité d'IS.
- Etudier la possibilité d'intégration du cadre proposé dans un atelier d'ingénierie système en réduisant l'effort de capture des choix et des décisions en cours de projets.

# Liste des publications

## -Sur le sujet de thèse :

- Olfa CHOURABI, Yann POLLET, Mohamed BENAHMED (2009). An Ontological Framework for Knowledge Management in Systems Engineering Processes. In Knowledge Management, in course of edition by Aleksandar Lazinica, publisher INTECH.( <http://intechweb.org/>)
- Olfa CHOURABI, Mohamed BENAHMED, Yann POLLET (2009). Ontological Framework for Engineering Knowledge Capitalization and Sharing. Accepted for publication in Third International Conference on Research Challenge in Information Science (RCIS), fez, Morocco, april, 22-24, IEEE.
- Olfa CHOURABI, Mohamed BENAHMED, Yann POLLET (2009). Capitalization and sharing of Situated Explicit Engineering Knowledge. Accepted for publication in 11th International Conference on Enterprise Information Systems (ICEIS), Milan, Italy, May 6-10, Springer.
- CHOURABI Olfa, Yann POLLET, Mohamed BENAHMED: Knowledge Modelling Framework for System Engineering projects. RCIS'2008. June 2008.( Best doctoral paper award)
- CHOURABI Olfa, , Mohamed BENAHMED, Yann POLLET: Ontology Based Knowledge Management in System Engineering. Insight Journal of INCOSE, Volume 11, ISSUE 3, July 2008.
- CHOURABI Olfa., Yann POLLET, Mohamed BEN AHMED. Ontology Based framework for system engineering knowledge modeling. IFIP AI 2008, Milan (Italy), September 2008.
- O.CHOURABI, Y.POLLET, M.BENAHMED. Cadre générique de modélisation de connaissances d'Ingénierie de Systèmes. le magazine de l'ingénierie du logiciel et de systèmes. Décembre 2007.
- O.CHOURABI, M.BENAHMED, Y.POLLET. Modélisation générique de connaissances de projets d'ingénierie de système. Actes de la quatrième conférence annuelle sur les avancées de l'ingénierie de systèmes. Toulouse, Palais de congrès, Mai 2006.
- O.CHOURABI, M.BENAHMED, Y.POLLET,. Integrating knowledge management and system engineering domain. (ICESSA 06).
- O.CHOURABI, M.BENAHMED, Y.POLLET. Knowledge Management support for system engineering community. Proceedings of the 7th international conference on enterprise information systems (ICEIS 05). Miami/USA, 24-28 MAI 2005.

## -Autres publications

- - Y.POLLET, O.CHOURABI. A Formal approach for optimized concurrent System Engineering. ENGOPT 2008. juin 2008.

- O.CHOURABI, Y.POLLET, **Le Langage de Modélisation de Systèmes SysML : Un compendium**, la revue de Génie logiciel ISSN 1265-1397, 2008, no84, pp. 32-36 [5 page(s) (article)] (1/4 p.).

# Références

- [**Aamodt et al., 1994**] : Aamodt, A., Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. *AI Communications, the European journal on AI*, Vol 7, N°1, p. 39-59.
- [**Abecker et al., 1998**] : Abecker, A., Bernardi, A. , Hinkelmann , K., Kühn , O., Sintek, M. (1998). Toward a Technology for Organizational Memories, *IEEE Intelligent Systems*, Vol 13, N°.3, p.40-48.
- [**Abecker et al., 2000**] : Abecker A., Bernardi A., Hinkelmann K., Kühn O. and Sintek M. (2000). Context- Aware, Proactive Delivery of Task-Specific Knowledge: The KnowMore Project. *Int. Journal on Information Systems Frontiers*, Vol 2, Issue 3-4 (October-November 2000), P. 253 – 276.
- [**Abecker et al., 2001**] : Abecker, A., Bernardi, A., Van Elst, L., Lauer, A., Maus, H., Schwarz, S., Sintek, M. (2001). FRODO: A Framework for Distributed Organizational Memories. *Milestone M1: Requirements Analysis and System Architecture. DFKI Document D-01-01, DFKI GmbH*, March 2001.
- [**AFIS, 2009 a**] : Association Française d'Ingénierie Système. [www.afis.fr](http://www.afis.fr) . Accédé le 9 déc. 2009.
- [**AFIS,2009 b**] : Découvrir et comprendre l'Ingénierie Système. Version expérimentale. Ouvrage collectif AFIS préparé par le groupe de travail Ingénierie Système. Version 3- 12-02-2009.
- [**Ahmed et al., 2005**] : Ahmed, S., Kim, S., Wallace, K.M. (2005) A methodology for creating ontologies for engineering design. *ASME International Design Engineering Technical Conferences (IDETC/CIE2005)*, 24-28 September, Long Beach, California, USA.
- [**Alavi et al., 2001**] : Alavi, M. et Leidner, D. ( 2001). Knowledge Management and Knowledge Management Systems : Conceptual Foundations and Research Issues", *MIS Quarterly*, Vol. 25, N° 1, p. 107-176.
- [**Alberts, 1993**] : Alberts, L. K. (1993). YMIR: an ontology for engineering design. Doctoral dissertation, University of Twente.
- [**ANSI/EIA 632**] : Electronic Industries Alliance. EIA-632 : Processes for systems engineering. ANSI/EIA-632-1998 Approved : January 7, 1999.
- [**Antunes et al, 2007**] : Antunes, B. , Seco, N. , Gomes, P. (2007). Using Ontologies for Software Development Knowledge Reuse. *Progress in Artificial Intelligence*. Vol. 4874/2007 P.357-368, Springer Berlin / Heidelberg, ISSN : 0302-9743 (Print) 1611-3349 (Online).

**[Assadi et al, 2000]** : Bourigault D., Assadi H. (2000). Analyse syntaxique et analyse statistique pour la construction d'ontologie à partir de textes, in Charlet J, Zacklad M., Kassel G., Bourigault D. éd. Ingénierie des connaissances. Tendances actuelles et nouveaux défis. Editions Eyrolles/France Telecom, p. 243-255.

**[Assali et al., 2009]** : Abou Assali, A., Lenne, D., Debray, B., Bouchet, S. (2009). COBRA : Une plate-forme de RàPC basée sur des ontologies. IC 2009 : Actes des 20es Journées Francophones d'Ingénierie des Connaissances, Hammamet, Tunisia, May 25-29, ISBN 978-2-7061-1538-7. [http://ic2009.inria.fr/docs/Actes\\_IC2009.pdf](http://ic2009.inria.fr/docs/Actes_IC2009.pdf).

**[AUBRY, 2007]** : Aubry. S. (2007). Annotations et gestion des connaissances en environnement virtuel collaboratif. Thèse, Université de Technologie de Compiègne. Spécialité informatique.

**[BAADER , 1991]** : Baader, F., McGuinness, D., Nardi, D., Schneider, P. (1991). The Description Logic Handbook: Theory, Implementation and Applications. *Cambridge University Press*.

**[Bachimont, 2000]** : Bachimont, B. (2000). Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en Ingénierie des connaissances. In J. Charlet, M. Zacklad, G. Kassel & D. Bourigault (Eds.), *Ingénierie des connaissances, évolutions récentes et nouveaux défis*. Paris: Eyrolles

**[Baget et al., 2002]** : Baget, J.F. and Mugnier, M.L. (2002). Extensions of Simple Conceptual Graphs: the Complexity of Rules and Constraints. *Journal of Artificial Intelligence Research (JAIR)*, vol. 16, pages 425-465. <http://www.jair.org/papers/paper918.html>.

**[Barthès, 1996]** : Barthès, J.-P. (1996), ISMICK and Knowledge Management. In J. F. Schreinemakers ed, *Knowledge Management: Organization, Competence and Methodology, Proc. of ISMICK'96*, Rotterdam, the Neth., Wurzburg: Ergon Verlag, October 21-22, p. 9-13.

**[Basili, 1995]** : Basili, V. (1995). The Experience Factory and its Relationship to Other Quality Approaches, in *Advances in Computers*, Vol.41, Academic Press.

**[Beaugard, 2009]** : Beaugard, G. (2009). Ligne de produits satellite de télécommunication SPACEUBUS. *Quatrième journée thématique de l'AFIS*. Architecture de systèmes-lignes de produits.

**[Bekhti, 2003]** : Bekhti, S., Matta, N., (2003). A formal approach to model and reuse the project memory, *Proceedings of the third international conferences on knowledge management I-KNOW'03*, Industry meets science, Graz, 2003

**[Ben AHMED, 2007]**: Mohamed BEN AHMED. (2007). Cognition entre Philosophie, Science et Technologie. *Centre de publication universitaire*, 647 p., ISBN 978-9973-37-417-2.

- [Benmhamed, 2006]** : Benmahamed, D., Ermine, J-L. (2006). Techniques de gestion et d'ingénierie des connaissances pour la conception des dispositifs de transfert de savoir-faire dans les métiers pétroliers. Actes des 17<sup>e</sup> journées francophones d'Ingénierie des connaissances. (IC 2006). Nantes, France, June 26-30, Université de Nantes.
- [Bergmann , 2002]** : R. Bergmann, Experience Management: Foundations, Development Methodology, and Internet-Based Applications, *Lecture Notes in Artificial Intelligence Series*, Vol. 2432 , Springer.
- [Bergmann et al., 2003]** : Bergmann, R., Schaaf. M. (2003). On the Relations between Structural Case-based Reasoning and Ontology-based Knowledge Management. *In Proc. of German Workshop On Experience Management*, P. 279-286, April.
- [Blomqvist et al., 2005]** : Blomqvist E., Sandkuhl K. (2005). Patterns in Ontology Engineering : Classification of Ontology Patterns. In: *Proc. of 7th International Conference on Enterprise Information Systems (ICEIS 2005)*, Miami, USA, May.
- [Boehm, 1988]** : Boehm, B. (1998). A Spiral Model of Software Development and Enhancement, *IEEE Computer*, Vol. 21, N° 5, p. 61-77
- [Borst et al., 1997]** : Borst, P. & Akkermans, H. (1997) Engineering ontologies. *International Journal of Human and Computer Studies*, Vol. 46, p. 365-406.
- [Broekstra,2005]** : Broekstra, J., Kampman, A., van Harmelen, F., Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. *in 1st International Semantic Web Conference, ISWC* , Sardinia, Italy, p. 54-68.
- [Bush et Tiwana, 2005]** : Bush, A. et Tiwana, A. (2005). Designing sticky knowledge networks. *Communication of the ACM*, vol. 48, n°5, p. 66-71.
- [Brandt et al., 2007]** : Brandt, C., Morbach, J., Miatidis, M., Theißen , M., Jarke , M., Marquardt, W. (2007). An ontology-based approach to knowledge management in design processes. *Computers and Chemical Engineering*, Vol.32, Issue 1-2, January, P.320-342
- [Carloni, 2006]** : Carloni, O., Leclère, M., Mugnier, M.-L., (2006). Introduction de raisonnement dans un outil de gestion de connaissances basé sur les Topic Maps, *Actes d'IC'06 (17èmes journées francophones d'Ingénierie des Connaissances)*, Semaine de la connaissance, Nantes. <http://hal-lirmm.ccsd.cnrs.fr/lirmm-00156561/en/>
- [Champin, 2002]** : Pierre-Antoine Champin (2002). Modéliser l'expérience pour en assister la réutilisation : de la Conception Assistée par Ordinateur au Web Sémantique , Thèse de doctorat en informatique, Université Claude Bernard - Lyon 1, Villeurbanne (FR) , décembre.
- [Chandrasekaran, 1999]**: Chandrasekaran, B. (January 1999). What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems*, Vol. 14, Issue 1 , P. 20 – 26, ISSN:1541-1672.

**[Charlet et al, 2000]** : Charlet, J., Zacklad, M., Kassel G., Bourigault, D. (2000). Ingénierie des connaissances : recherches et perspectives. In CHARLET, J., ZACKLAD, M., KASSEL G., BOURIGAULT D. (Eds). *Ingénierie des connaissances, évolutions récentes et nouveaux défis*. Eyrolles.

**[Chung, 2003]**: Chung Liang, V., and Paredis, C. (2003). A Port Ontology for Automated Model Composition. *Winter Simulation Conference*, New Orleans, LA, 2003.

**[Corby, 2008]** : Corby,O. (2008). Web, Graphs & Semantics, Proc. of the 16th International Conference on Conceptual Structures (ICCS'2008), July 2008, p. 43-61, Toulouse.

**[Cordier et al., 2006]** : Cordier, A. , Fuchs, B. (2006). Apprendre à mieux adapter en raisonnement à partir de cas. Dans 14ème Atelier de Raisonnement à Partir de Cas, Brigitte Morello ed. Besançon, France. P. 27-37.

**[Dameron, 2003]**: Olivier Dameron. (2003). Modélisation, représentation et partage de connaissances anatomiques sur le cortex cérébral. Thèse , Université de Rennes 1.

**[Darses, 2002]** : Darses, F. (2002). Trois conditions sociotechniques Pour l'optimisation De la conception continue du Système de production. *Revue Française de Gestion Industrielle*, Vol. 21, N°1, P. 5-27.

**[Davenport, 1998]** : Davenport, T.H. (1998), Some principles of knowledge management, available at: [www.bus.utexas.edu/kman/kmprin.htm](http://www.bus.utexas.edu/kman/kmprin.htm).

**[Damjanović et al., 2007]** : Damjanović, V., Behrendt, W., Plößnig, M., Holzapfel, M. (2007). Developing Ontologies for Collaborative Engineering in Mechatronics. Proceedings of the 4th European conference on The Semantic Web: Research and Applications. Lecture Notes In Computer Science; Vol. 4519, P. 190 – 204, ISBN:978-3-540-72666-1.

**[Després et al, 2006]**: Després, S. and Szulman, S. (2006). Terminae method and integration process for legal ontology building. *In Proceedings of the 19th International Conference IAE/AIE*. Special Session on Ontology and text- Springer Berlin/Heidelberg Advances in Applied Artificial Intelligence: 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2006, Annecy, France, P. 1014-1023, 2006

**[Diagne et al, 1998]** :Diagne, A., Estrallier, P., Kordon, F., Vernier, I., Cazin, J., Doche, M., Le Berre, D., Michel, P., Seguin, C. and Wiels, V. (1998). Opération VaMoS. Spécification et Validation Modulaires de Systèmes Avioniques. In Action FORMA. Bilan de la première année, p. 69-94, janvier.

**[Dieng et al, 1998]** : Dieng, R., Corby, O., Giboin, A. et Ribière, M. (1998). Methods and tools for corporate Knowledge Management. In Proceedings of the 11th workshop on Knowledge Acquisition, Modeling and Management (KAW'98), p.17–23, Banff, Canada,

- [Dieng et al.01]** : Dieng-Kuntz R., Corby O., Gandon F., Giboin A. Golebiowska J., Matta N. Ribière M., Methodes Et Outils Pour La Gestion Des Connaissances: Une approche pluridisciplinaire du Knowledge Management (2nd Edition).
- [Dieng et al.2004]** : R. Dieng-Kuntz, O. Corby, F. Gandon, J. Golebiowska, (2004). Ontologies pour la construction d'un web sémantique d'entreprise, Chapitre 1 dans le livre "Gestion dynamique des connaissances industrielles" ed. B. Eynard, M. Lombard, N. Matta, J. Renaud, Hermès, 2004. p. 27-43. Traité IC2. ISBN 2-7462-0952-7.
- [Doche et al, 1998]** : M. Doche, J. Cazin, D. Le Berre, P. Michel, C. Seguin, V. Wiels. Module Templates for the Specification of Fault-Tolerant Systems.DASIA 98 (DATA System In Aerospace) May 25-28, 1998, Athens, Greece.
- [Domingue , 1998]** : Domingue J. : Tadzebao and WebOnto : Discussing, Browsing, and Editing Ontologies on the Web. In 11th Banff Knowledge Acquisition Workshop (KAW'98), Banff, Canada, 1998.
- [Ermine, 2002]** : J.-L. Ermine.(2002) La gestion de connaissances, Hermès sciences publications, 2002.
- [Euzenat, 2005]** : EUZENAT J. (2005). L'annotation formelle de documents en (8) questions. In Ingénierie des connaissances. L'Harmattan.
- [Eynard, 1999]** : Eynard B. (1999). Modélisation du produit et des activités de conception. Contribution à la conduite et à la traçabilité du processus d'ingénierie. PhD thesis, Université de Bordeaux I, Bordeaux.
- [Fannmy,2009]** : G. Funmy (2009). Lignes de produits dans un contexte automobile. Quatrièmes de journée thématique de l'AFIS. Architecture de systèmes-lignes de produits.
- [Farrar , 2004]** : Farrar, S. and Bateman, J. General ontology baseline. In Deliverable D1I1-[OntoSpace] ; Workpackage 11, April 2004.
- [FIORESE,2005]** : FIORESE Serge. Processus de l'IS. Présentation de l'IS, Fiche 3, extrait de IS, pourquoi? Comment? . [www.afis.fr](http://www.afis.fr).
- [Forsberg, 1995]** : K. Forsberg, M. Harold, (1995). Application of the "Vee" to Incremental and Evolutionary Development, Proceedings of the Fifth Annual International Symposium of the National Council on Systems Engineering, St. Louis, MO, Juillet.
- [Furst, 2004]** : Frédéric FÜRST,(2004). Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationnalisation. Thèse d'Informatique, Université de Nantes.
- [Gamma, 1995]** : Gamma E., Helm R., Johnson R. et Vlissides J. (1995). Design Patterns : Elements of Reusable Object-Oriented Software. Addison-Wesley.
- [Gandon , 2008]**: Gandon, F. (2008) .Graphes RDF et leur Manipulation pour la Gestion de Connaissances. Mémoire d'Habilitation à Diriger les Recherches.

**[Gandon, 2000]**: Fabien Gandon, Rose Dieng, Alain Giboin, (2000). CoMMA : Une approche distribuée de la mémoire organisationnelle, Séminaire Systèmes distribués et Connaissances, INRIA Sophia-Antipolis, France, November 27-28.

**[Gangemi et al, 1999]**: Gangemi A., Pisanelli D. M. et Steve G. : An Overview of the ONIONS project : Applying Ontologies to the Integration of Medical Terminologies. *Data and Knowledge Engineering*, 31(2), 1999.

**[Gangemi et al, 2003]** : Gangemi A. et Mika P. : Understanding the Semantic Web through Descriptions and Situations. In *International Conference on Ontologies, Databases and Applications of Semantics (ODBASE'03)*, Catania, Italy, 2003.

**[Gangemi et al., 2004]** : Gangemi, A., Catenacci, C., Battaglia, M.: The Inflammation Ontology Design Pattern: an Exercise in Building a Core Biomedical Ontology with Descriptions and Situations. In Pisanelli D and Smith B, *Biomedical Ontologies*, IOS Press.

**[Gangemi, 2006]**: Aldo Gangemi. (2006). *Ontology Design Patterns: A primer, with applications and perspectives*. Tutorial on ODP, Laboratory for Applied Ontology Institute of Cognitive Sciences and Technology CNR, Rome, Italy.

**[Gao et al., 2003]** : Gao, J. X., Aziz, F. L., Maropoulos, P. G., & Cheung, W. M. (2003). Application of product data management technologies for enterprise integration. *International Journal of Computer Integrated Manufacturing*, 16(7–8), 491–500.

**[Goel et al., 1989]**: Goel, A. & Chandrasekaran, B. (1989) Functional representation of designs and redesign problem solving. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pp. 1388–1394. Los Altos, CA: Morgan Kaufmann.

**[GÓMEZ-PÉREZ et al., 2003]** : GÓMEZ-PÉREZ A., FERNÁNDEZ-LÓPEZ, M. & CORCHO, O. (2003). *Ontological Engineering : with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, First edition. New York: Springer.

**[Gomez-Perez, 1999]**: Asuncion Gomez-Perez, (1999). *Ontological Engineering: a State of the Art*, Expert Update. British Computer Society. Vol. 2. n° 3., pp. 33 – 43.

**[Graves, 2007]** : H. Graves. (2007). *Ontology engineering for product development*. In *Proc. of the Third OWL Experiences and Directions Workshop*.

**[Graves, 2008]** : Henson Graves<sup>1</sup> and Ian Horrocks (2008). *Application of OWL 1.1 to Systems Engineering*. [http://www.webont.org/owled/2008dc/papers/owled2008dc\\_paper\\_9.pdf](http://www.webont.org/owled/2008dc/papers/owled2008dc_paper_9.pdf)

**[GRUBER, 1995]** : GRUBER, T. R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies*, 43(5/6): 907-928, 1995.

**[GRUBER, 93]** : GRUBER, T. R.: A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5 (2): 199-220, 1993.

**[Grundstein, 1995]** : Michel Grundstein: La Capitalisation des Connaissances de l'Entreprise, Système de Production des Connaissances. Actes du Colloque L'Entreprise Apprenante et les Sciences de la Complexité, Aix en Provence, France, 22-24 Mai 1995.

**[Grundstein, 2000]** : Grundstein M., Management des connaissances de l'entreprise: problématique, axe de progrès, orientations, 2000.

**[GRUNINGER et al, 1995]** : GRUNINGER M. & FOX M. S., Methodology for the design and evaluation of ontologies, in Proceedings of the Workshop on Basic Ontological Issues on Knowledge Sharing, IJCAI'95, 1995.

**[GUARINO, 1995]** : N. GUARINO et P. GIARETTA. Ontologies and knowledge bases, towards a terminological clarification, pages 25–32. IOS Press, 1995

**[Guarino, 1998]** : Nicola Guarino (ed.), Formal ontology in information systems, IOS press, Amsterdam (NL), 1998.

[Guarino, 1998] : GUARINO, N.: Formal Ontology and Information Systems. In Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998, 3-17. IOS Press: Amsterdam, 1998.

**[Guizzardi, 2005]** : Guizzardi, G. Ontological Foundations for Structural Conceptual Models, PhD Thesis (CUM LAUDE), University of Twente, The Netherlands. Published as the book "Ontological Foundations for Structural Conceptual Models", Telematica Instituut Fundamental Research Series No. 15, ISBN 90-75176-81-3 ISSN 1388-1795; No. 015; CTIT PhD-thesis, ISSN 1381-3617; No. 05-74. (read about the PhD promotion and the Cum Laude distinction in the UTNieuws, Jaargang 40, Number 32, Nov.2005 – in dutch).

**[Guizzardi, 2006]** : Guarino, N.; Guizzardi, G., "In the Defense of Ontological Foundations for Conceptual Modeling", Scandinavian Journal of Information Systems, Vol.18, No. 1, ISSN 0905-0167, 2006.

**[Hansen et al., 1999]** : Hansen M. T., Nohria N., et Tierney T. "What's your strategy for managing knowledge?" Harvard Business Review, vol. 77, n°2, pp. 106-116. 1999

**[Heijst, 1996]** : G. van Heijst, R. van der Spek, and E. Kruizinga. Organizing corporate memories. In Proceedings of Tenth Knowledge Acquisition for Knowledge-Based Systems (KAW96), Ban Alberta, Canada, November 1996.

**[Hew et al , 2001]** : Ping Hew K., Fisher N., Awbi H., « Towards an integrated set of design and tools based on a common data format for building and services design », Automation in Construction, Vol. 10, n°4, 2001, p 459-476.

**[Holz, 2003]** : H. Holz: An Incremental Approach to Task-Specific Information Delivery in SE Processes. Proceedings of the 18th IEEE International Conference on Automated Software Engineering (ASE '03), Montreal, Canada, 200.

- [INCOSE,2004]** : Technical Board International Council on Systems Engineering (INCOSE), Systems engineering Handbook, INCOSE-TP-2003-016-02, Version 2a, 1 Juin 2004.
- [Isaac, 2005]** : Antoine Isaac. Conception et utilisation d'ontologies pour l'indexation de documents. Thèse de Doctorat, Université Paris IV – Sorbonne. 2005.
- [Jatinder et al, 2008]** : Jatinder N. D. Gupta, Sushil K. Sharma, Jeffrey Hsu. An Overview of Knowledge Management. Information Science Reference (an imprint of IGI Global). <http://www.igi-pub.com/reference>. 2008.
- [Jézéquel et al, 2008]** : Jean-Marc Jézéquel, Gilles Perrouin. Vers des Lignes de Produits Flexibles: Apports de l'Ingénierie Dirigée par les Modèles à la Dérivation de Produits. RSTI L'objet – 14/2008. Lignes de produits logiciels, pages 33 à 45.
- [Keller et al, 2007]** : Keller, U., Reitbauer, A., and Mungenast, R. (2007). Principles of SENSEful Engineering Support Systems. TechReport at <http://www.uwekeller.net/publications.html> DERI TR 2007-01-07.
- [KIFER, 1995]** : M. KIFER, G. LAUSEN et J. WU. Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the ACM, 42(4):741–843, 1995.
- [Kimatura, 2006]** : YOSHINOBU KITAMURA. Roles of ontologies of engineering artifacts for design knowledge modeling. In Proc. of the 5th International Seminar and Workshop Engineering Design in Integrated Product Development (EDIProD 2006), 21-23 September 2006, Gronów, Poland, pp. 59-69, 2006.
- [Klein et al, 2000]** : Klein, R in M. Spitalni 'Towards an integration of engineering knowledge management and knowledge based engineering', Proceedings of the CIRP Conference on Advanced Engineering Information Systems, Haifa, Israel, May 2000.
- [Kopena et al., 2003]** : Joseph B. Kopena , William C. Regli .Design Repositories On The Semantic Web With Description-Logic Enabled Services (2003). Proceedings of VLDB Semantic Web and Databases Workshop. <http://edge.cs.drexel.edu/assemblies/publications/>
- [Kraines et al, 2006]** : Kraines, S. B., W. Guo, B. E. Kemper, and Y. Nakamura. 2006a. EKOSS: A knowledge-user centered approach to knowledge sharing, discovery, and integration on the Semantic Web. Proceedings of the 2006 International Semantic Web Conference.
- [Lassila et al, 2001]** : Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The semantic web. Scientific American, may 2001, 35-43.
- [Leclère et al, 2005]** : Comte F., Leclère M., Mugnier M.L. : "OWL-SG : un sous-langage pour la famille OWL". Journée thématique raisonner le web sémantique avec des graphes (RWSG'05), Nice, France, 2005.

- [Leclère et al, 2005]** : Comte F., Leclère M., Mugnier M.L. : "OWL-SG : un sous-langage pour la famille OWL". Journée thématique raisonner le web sémantique avec des graphes (RWSG'05), Nice, France, 2005.
- [Leclère et al, 2007]** : Moreau N., Leclère M., Chein M., Gutierrez A.: Annotation Formelle Graphique de Documents Multimedia Actes d' IC'07 (18èmes journées francophones d'Ingénierie des Connaissances), Cépaduès Edition, pp 313-324, Grenoble, 2007.
- [Lee et al, 2009]** : Jeongsoo Lee Design of product ontology architecture for collaborative enterprises. Expert Systems with Applications Volume 36, Issue 2, Part 1, March 2009, Pages 2300-2309.
- [Leblanc, 2008]** : Leblanc, P., (2008). Modélisation SysML par l'Exemple. Revue de Génie logiciel, vol. 84, p. 37-45, ISSN 1265-1397.
- [Lin et al, 1996]** : Lin, J., Fox, M. S.; Bilgic, T.: A Requirement Ontology for Engineering Design. Enterprise Integration Laboratory,, University of Toronto, Manuscript, September (1996).
- [Longueville et al, 2003]** : Longueville B., Stal Le Cardinal J. and Bocquet J.-C. Toward a project memory for innovative production design, a decision-making process model.. ICED 03, 14th International Conference on Engineering Design. Stockholm, Sweden, 8/2003.
- [M. Chein et M.-L. Mugnier,04]** : M. Chein et M.-L. Mugnier. Concept Types and Coreference in Simple Conceptual Graphs. In Proc. ICCS'04, LNAI. Springer, 2004.
- [Maier, 2002]** : Ronald Maier State-of-Practice of Knowledge Management Systems: Results of an Empirical Study. The European Online Magazine for the IT Professional. <http://www.upgrade-cepis.org> Vol. III, No. 1, February 2002.
- [Masolo et al., 2003]** : Claudio MASOLO, Stefano BORGIO, Aldo GANGEMI, Nicola GUARINO, Alessandro OLTRAMARI, Roberto OLTRAMARI, Luc SCHNEIDER and Ian HORROCKS: **WonderWeb Deliverable D17. The WonderWeb Library of Foundational Ontologies and the DOLCE ontology**, 2002. WonderWeb Deliverable D18, Final Report (version 1.0, 31-12-2003).
- [Matta et al., 2001]** : Bekhti S., Matta N., Andéol B., Aubertin G., Représentation des connaissances dans une mémoire de projet, Document Numérique, Numéro Spécial Création et Gestion Coopératives de Documents Numériques d'Information et de Communication, 2001.
- [Matta, 1998]** : Ribière M., Matta N., Cointe C., A proposition for managing project memory in concurrent engineering, In proceedings International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'98), Churchill, Australia, February 1998.

- [Matta, 2004]** : Matta, N. Rapport Habilitation à diriger des Recherches. Ingénierie des connaissances en conception pour la mémoire de projet, de l'Université de Technologie de Compiègne, 2004
- [Mayank et al., 2002]** : Mayank, V., Kositsyna, N., Austin, M.: Requirements Engineering and the Semantic Web, Part II. Representation, Management, and Validation of Requirements and System-Level Architectures. Technical Report. TR 2004-14, University of Maryland (2004)
- [McMahon, et al, 2004]** : McMahon, C., Lowe, A., & Culley, S. J. Knowledge management in engineering design, personalisation and codification. Journal of Engineering Design, 15(4), 307–325. 2004.
- [Meinadier, 2002]** : J.Meinadier. Le métier d'intégration de systèmes. Hermes Science Publications, décembre 2002.
- [Mendes, 2003]** : État de l'art sur les méthodologies d'ingénierie ontologique. Montréal, Québec, Canada: Centre de recherche LICEF. 2003
- [Mirbel, 2008]** : Isabelle Mirbel. Vers une ontologie pour les communautés de développement de logiciel libre. 19es Journées Francophones d'Ingénierie des Connaissances (IC 2008), Nancy : France (2008)"
- [Mizoguchi et al., 2004]** : Kitamura, Y., & Mizoguchi, R. (2004). Ontology-based systemization of functional knowledge. Engineering Design, 15(4), 327-351.
- [Mizoguchi, 2004 a]**: Tutorial on Ontological Engineering. Part 2: Ontology Development, Tools and Languages. New Generation Computing, 22, 61-96, Ohmsha and Springer Verlag.
- [Mizoguchi, 2000]** : Riichiro Mizoguchi, A Personal History of AI Research Activity - Expert System, Knowledge Engineering and Ontological Engineering.
- [Mizoguchi, 2003]** : Tutorial on Ontological Engineering. Part 1: Introduction to Ontological Engineering. New Generation Computing, 21, 365-384, Ohmsha and Springer Verlag.
- [Mizoguchi, 2004 b]** : Tutorial on Ontological Engineering. Part 3: Advanced Course on Ontological Engineering. New Generation Computing, 22, 196-220, Ohmsha and Springer Verlag.
- [MIZOGUCHI, 95]** : MIZOGUCHI, R., et al.: Ontology for Modeling the World from Problem Solving Perspectives, in Proceedings of IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, pp. 1-12, 1995.

- [**Mizoguchy, 1998**] : Mizoguchi, R. A Step towards Ontological Engineering, National Conference on AI of JSAI, AI-L13, 1998, <http://www.ei.sanken.osaka-u.ac.jp/english/step-onteng.html>.
- [**MONTICOLO, 2008**] : Davy MONTICOLO. Une approche organisationnelle pour la conception d'un système de gestion des connaissances fondé sur le paradigme agent. Thèse. Laboratoire Systèmes et Transport. Université de Technologie de Belfort-Montbéliard 2008
- [**NASA,1995**] : NASA Systems Engineering Handbook: Fundamentals of Systems Engineering. SP-610S juin 1995.
- [**Niles et al., 2001**] : Ian Niles and Adam Pease. Origins of the standard upper merged ontology : A proposal for the iee standard upper ontology. In IEEE Standard Upper Ontology, pages 37–42. Working Notes of the IJCAI-2001 Workshop on the, 2001.
- [**Niles et al., 2001**] : Ian Niles, Adam Pease . Towards a Standard Upper Ontology FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA. Copyright 2001 ACM.
- [**Nonaka & Takeuchi, 1995**] : Nonaka I. et Takeuchi H., The Knowledge-Creation Company : How Japanese Companies Create the Dynamics of Innovation. New York/Oxford, Oxford University Press, 1995.
- [**Noy , 2003**] : Noy N. F. et Musen M. A. : The PROMPT suite : interactive tools for ontology merging and mapping. International Journal of Human-Computer-Studies, 59(6), 2003.
- [**Oberle et al, 2006**] : Oberle D., Lamparter S., Grimm S., Vrandecic D., Staab S., Gangemi A., "Towards Ontologies for Formalizing Modularization and Communication in Large Software Systems", Applied Ontology, vol. 1, n° 2, 2006, p. 163-202.
- [**Oldham et al, 1998**] : Oldham, K, Kneebone, S, Callot, M, Murton, A and Brimble, R, in N. Mårtensson, R. Mackay and S. Björgvinsson. 'MOKA - A Methodology and tools Oriented to Knowledge-based engineering Applications. Changing the Ways We Work, Advances in Design and Manufacturing, Volume 8, Proceedings of the Conference on Integration in Manufacturing, Göteborg, Sweden, IOS Press, Amsterdam, October 1998, 198-207.
- [**Polanyi, 1996**] : Polanyi, M., The Tacit Dimension, Routledge & Kegan Paul, Gloucester, 1966
- [**Pomian, 1996**] : J.Pomian. Mémoire d'entreprise, techniques et outils de gestion du savoir, Sapienza. 1996.
- [**Prié, 2008**] : Yannick Prié, Ingénierie ontologique. UFR Informatique Université Claude Bernard Lyon 1. 2008.
- [**Prusak, 2001**] : Larry Prusak. Where did Knowledge Management come from?, IBM System Journal, 2001.

**[Psyché et al , 2003]** : Valéry Psyché , Olavo Mendes ,Jacqueline Bourdeau Apport de l'ingénierie ontologique aux environnements de formation à distance. Revue des Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF) 10 (2003) 89-126.

**[Psyché, 2003]** : Valéry Psyché, Olavo Mendes, Jacqueline Bourdeau, "Apport de l'ingénierie ontologique aux environnements de formations à distance", revue sticef.org, Volume 10, 2003.

**[Rakoto, 2004]** : H. Rakoto. Intégration du Retour d'Expérience dans les processus industriels : Application a Alstom Transport (in French), PhD thesis, National Polytechnic Institute of Toulouse (France), October 2004.

**[Regli et al, 2000]** : Xiaochun Hu, Jun Pang, Yan Pang, Michael Atwood, Wei Sun, William C. Regli.A Survey On Design Rationale: Representation, Capture And Retrieval. Proceedings of DETC'00 2000 ASME Design Engineering Technical Conferences September 10-13, 2000, Baltimore, Maryland

**[Regli et al. 2003]** : Kopena, J., & Regli,W. C. (2003). Functional modeling of engineering designs for the semantic web. IEEE Data Engineering Bulletin, 26(4), 55–61.

**[Reimer , 2000]** : U. Reimer , EULE: A Knowledge-Based System to Support Business Processes. Knowledge-Based Systems Volume 13, Issue 5, October 2000, Pages 261-269.

**[RENAUD et al., 2007]** : RENAUD J., MORELLO B., FUCHS B. & LIEBER J. . Raisonement `a Partir de Cas 1 : conception et configuration de produits. Hermes-Lavoisier, February. 2007.

**[Royce, 1970]** : W.W. Royce, Managing the Development of Large Software Systems, Proceedings of IEEE WESCON 26, pp. 1-9, Août 1970.

**[Rus et al, 2001]** : Rus I., Lindvall M.. Knowledge Management in Software Engineering A State-of-the-Art-Report. Produced by Fraunhofer Center for Experimental Software Engineering Maryland and The University of Maryland, 2001.

**[Saeki et al., 2006]** : Saeki, M., Kaiya, H. (2006). On Relationships Among Models, Meta Models, and Ontologies. Proceedings of the 6th OOPSLA Workshop on Domain-Specific Modeling.

**[Salinesi et al., 2009]** : Camille Salinesi, Daniel Diaz , Raul Mazo , Olfa Djebbi. Spécification d'exigences dans le contexte de lignes de produits. IDM-INFORSID. 2009.

**[Sarder et al, 2007]** : Sarder, B. Ferreira, S. Developing Systems Engineering Ontologies System of Systems Engineering, 2007. SoSE '07. IEEE International Conference on

**[Sharmin et al, 2009]** : Moushumi Sharmin, Brian P. Bailey, Cole Coats, and Kevin Hamilton. Understanding Knowledge Management Practices for Early Design Activity and Its Implications for Reuse. CHI 2009, April 4–9, 2009, Boston, MA, USA.

**[SOWA, 1984]** : J. SOWA. Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, 1984.

**[SOWA, 1995]** : SOWA, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Boston, MA: PWS Publishing Company, 1995.

**[Susca et al. 2000]** : Susca L., Mandorli F., Rizzi C., Cugini U., « Racing car design using knowledge aided engineering », Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 14, 2000, p. 235-249

**[Sveiby, 1997]** : Sveiby K., The New Organization Wealth. San Francisco: Berrett-Koehler, 1997.

**[Sycara et al. 1991]** : Sycara, K., Guttal, R., Koning, J., Narasimhan, S. & Navinchandra, D. 1991 CADET: a case-based synthesis tool for engineering design. International Journal of Expert Systems, 4(2)157–188.

**[Szykman et al, 2000]** : Szykman, S., J. W. Racz, C. Bochenek, and R. D. Sriram (2000), “A Web-based System for Design Artifact Modeling,,” Design Studies, Vol. 21, No. 2, pp. 145-165.

**[Szykman, 2001]** : Simon Szykman, Ram D. Sriram, William C. Regli the role of knowledge in next-generation product development systems ASME Journal of Computation and Information Science in Engineering Volume 1, Number 1, 2001

**[Temal, 2008]** : Lynda TEMAL, Ontologie de partage de données et d'outils de traitement dans le domaine de la neuroimagerie. Thèse de l'Université Rennes I, Juin 2008. <http://www.irisa.fr/visages/publi/category/Theses-fra.html>

**[Turki, 2008]** : Skander Turki, *Ingénierie système guidée par les modèles: Application du standard IEEE 15288, de l'architecture MDA et du langage SysML à la conception des systèmes mécatroniques*. Discipline Sciences et techniques industrielles, Université du Sud Toulon, 2008, 197 p.

**[Uschold et al, 1999]** : Uschold, M. & Jasper, R. (1999) A Framework for Understanding and Classifying Ontology Applications. Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods, Stockholm.

**[Uschold et King, 1995]** : Uschold M. et King M. (1995). Towards a Methodology for Building Ontologies. Paper presented at the Workshop on Basic Ontological Issues in Knowledge Sharing.

[**Uschold, 1996**] : Building Ontologies: Towards a Unified Methodology In 16th Annual Conf. of the British Computer Society Specialist Group on Expert Systems(1996).

[**U'Ren , 2003**] U'Ren J., “An Overview of AP233 STEP’s Systems Engineering Standard” in proceedings of the 6th Annual System Engineering Conference, San Diego, CA, Oct 2003

[**Wenger, 1998**] : Wenger E. “Communities of Practice”, Learning, Meaning, and Identity, Cambridge University Press:New York, 336 p. 1998

[**Wiels, 1997**] : V. Wiels. Modularité pour la conception et la validation formelle de systèmes. Rapport de thèse ENSAE, octobre 1997.

[**Wiig, 1999**] : Wiig, K.M. (1999) What future knowledge management users may expect, in: Journal of Knowledge Management, Vol. 3, No. 2, pp. 155-165.

[**Wilson, 2002**] : Wilson T. D..The nonsense of 'knowledge management." Information Research, vol. 8, n°1. 2002.

[ **Wand, 1996**] Wand, Y. (1996). Ontology as a foundation for meta-modelling and method engineering. Information and Software Technology, 38, 281–287.

[**Zacklad et al., 2001**] : Imed Boughzala, Manuel Zacklad, Nada Matta: Gestion des connaissances dans une entreprise étendue - Mémoire d'entreprise et systèmes d'information coopératifs interentreprises. EGC 2001: 259-270.

[**Zargayouna et al., 2004**] : H. Zargayouna, S. Salotti Mesure de similarité sémantique pour l'indexation de documents semi-structurés dans 12ème Atelier de Raisonnement à Partir de Cas, Mars 2004.

[**Zdrahal et al, 2002**] : Martin Dzbor et Zdenek Zdrahal. Interactions between specification and solution in design. European Conference on AI. ECAI-2002.

[**Zdrahal et al. 2003**] : Zdrahal Z., Mulholland P., Valasek M., Sainter P., Koss M. and Trejtnar L.: A toolkit and methodology to support the collaborative development and reuse of engineering models, DEXA 2003.

[**Zhang et al., 2006**] : W. Y. Zhang & J. W. Yin . Exploring Semantic Web technologies for ontology-based modeling in collaborative engineering design Springer-Verlag London Limited 2006.

[**Ziadi, 2003**] : Ziadi T., Hélouët L., Jézéquel J.-M., « Towards a UML Profile for Software Product Lines », PFE2003 : 5th International Workshop on Software Product-Family Engineering, vol. 3014 of Lecture Notes in Computer Science, Springer, Siena, Italy, p. 129-139, November, 2003.