



HAL
open science

Ingénierie de la connaissance et génie logiciel : la modélisation objet et le microscope de la connaissance

Denis Malavieille, Mathias Chaillot, Jean-Louis Ermine, Boris Charreton,
Philippe Bigeon

► To cite this version:

Denis Malavieille, Mathias Chaillot, Jean-Louis Ermine, Boris Charreton, Philippe Bigeon. Ingénierie de la connaissance et génie logiciel : la modélisation objet et le microscope de la connaissance. 1997. hal-00998814

HAL Id: hal-00998814

<https://hal.science/hal-00998814>

Preprint submitted on 2 Jun 2014

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.

**Ingénierie de la connaissance et génie logiciel :
la modélisation objet
et le microscope de la connaissance.**

Denis Malavieille, Mathias Chaillot, Jean-Louis Ermine, Boris Charreton, Philippe Bigeon

1997 Non publié

Ingénierie de la connaissance et génie logiciel : la modélisation objet et le microscope de la connaissance.

Denis Malavieille, Mathias Chaillot, Jean-Louis Ermine, Boris Charreton, Philippe Bigeon

Commissariat à l'Énergie Atomique, Centre d'Études Nucléaire de Saclay

DIST/SMTI, Groupe Gestion des Connaissances

CEA, CEN Saclay

91190 Gif sur Yvette Cedex

RESUME : Pour aborder la problématique de la gestion du patrimoine des logiciels scientifiques du CEA, nous pensons qu'il faut adopter le point de vue des physiciens et non plus uniquement celui de l'informatique. Notre expérience nous amène dans ce but à considérer le besoin d'une approche méthodologique globale de l'ingénierie de la connaissance, intégrant une méthode de génie logiciel orientée objet. Cette démarche nous a permis d'identifier des voies de recherches qui aboutiront à des améliorations de l'aide au pilotage des codes.

MOTS CLES: ingénierie des connaissances, génie logiciel, systémique, microscope, OMT, pilotage de programmes, calcul scientifique.

ABSTRACT: In order to get on to the management of the CEA's scientific softwares patrimony, we think we must take the physicists view point and not only the computer science one. Our experience then leads us to consider the need for a global methodological approach of knowledge engineering, integrating a object oriented software engineering methodology. In following this approach we identified research directions wich will bring some improvements in helping with program monitoring.

KEY WORDS: knowledge engineering, software engineering, systemic, microscope, OMT, program monitoring, scientific computing.

1. Introduction

Cet article aborde la problématique de l'interaction entre la gestion des connaissances et le développement logiciel dans le cas particulier des logiciels scientifiques. Au CEA cette question est centrale dans la mesure où les différentes unités ont été amenées à développer et mettre au point de nombreux outils de calcul. Le portefeuille du CEA comprend 196 logiciels libres (logiciels scientifiques ouverts non couverts par des clauses de confidentialité) parmi lesquels 21 très grands logiciels dépassent 100 000 lignes de code. Ils concernent les domaines suivants : 7 en thermohydraulique, 4 en mécanique et thermique, 3 en neutronique, 1 en protection contre les rayonnements ionisants, 1 en imagerie, 3 outils généraux, 1 en temps réel et 1 en robotique. Cet aperçu quantitatif du patrimoine logiciel du CEA montre l'importance du problème de la gestion des connaissances associées à ces codes dont seulement 45% ont moins de cinq ans¹. L'effort (considérable) en cours pour gérer ce patrimoine consiste à reprendre l'existant en appliquant l'approche orientée objet afin de transformer les logiciels monolithiques en bibliothèques de composants réutilisables.

La reprise de l'existant réalisée aujourd'hui relève donc d'une démarche informatique et n'aborde pas le point de vue de l'utilisateur (le physicien ...) pour lequel deux grandes interrogations demeurent :

- Quels composants je dois utiliser pour résoudre mon problème ?
- Comment valuer les paramètres des fonctions proposées pour ces composants ?

Il semble aujourd'hui admis que cet aspect de l'utilisation des codes relève des techniques d'ingénierie de la connaissance en ce qu'il s'agit de modéliser une expertise humaine. Mais si les systèmes à base de connaissances peuvent apporter une aide significative aux physiciens, ils ne proposent selon notre expérience aucune réponse à la deuxième question. La raison en est simple : les paramètres des logiciels n'ont aucun liens directs avec les phénomènes physiques modélisés. Il faudrait donc effectuer l'équivalent d'une rétroingénierie sur la sémantique des paramètres, par rapport aux points de vue des différents utilisateurs potentiels des bibliothèques de composants. Il faut pour cela disposer d'un cadre méthodologique qui remplace les codes, ainsi que les systèmes à

¹ Sources : Dossier de presse « Les très grands logiciels du CEA » CEA DCOM/SIP, février 1994.

base de connaissances développés, dans une problématique de gestion des connaissances au niveau de l'organisation dans laquelle ces bibliothèques seront utilisées.

Nous proposons dans ce but d'appliquer la méthode MKSM (Method for Knowledge System Management) développée au CEA par le Groupe de Gestion des Connaissances de la DIST (Direction de l'Information Scientifique et Technique). Après avoir présenté les fondements théoriques de la méthode basés sur la notion de macroscopie, nous verrons qu'il existe des liens entre une méthode d'ingénierie de la connaissance et la méthode de génie logiciel OMT. Ces liens permettent d'établir une continuité méthodologique entre MKSM et OMT, et donc d'intégrer le développement des systèmes à base de connaissances dans une problématique de gestion des connaissances. Nous verrons enfin en quoi MKSM apporte une réponse à la valuation des paramètres, et de façon plus générale aux difficultés liées au facteur d'échelle des projets concernés par les logiciels scientifiques.

2. Le macroscopie de la connaissance dans la méthode MKSM

La méthode MKSM part du principe que la connaissance est un objet complexe qui ne peut être maîtrisé qu'en le décrivant suivant plusieurs points de vue qui rendent compte de réalités différentes. Ces points de vue constituent un macroscopie [DeRosnay75] pour les systèmes de connaissances dont les fondements théoriques (voir [Ermine96b]) sont les suivants :

2.1 Le triangle sémiotique

La première hypothèse forte à la base de la méthode MKSM est que le patrimoine de connaissances d'une organisation est un « objet », un « phénomène », perçu par tous comme un ensemble global d'éléments abstraits, concrets, conceptuels, matériels etc. qui, même s'il est malaisé de les distinguer ou de les interpréter, donnent un sens, une signification, une cohésion intrinsèque au système. C'est la perception globale de ces éléments qui justifie qu'il existe bien quelque chose susceptible d'observation, voire de modélisation, que nous ne connaissons pas encore et que nous ne pouvons peut-être même pas nommer pour le moment.

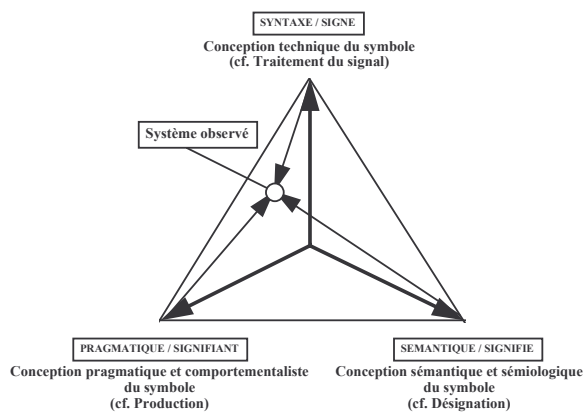


Figure 1 : Les trois axes du triangle sémiotique

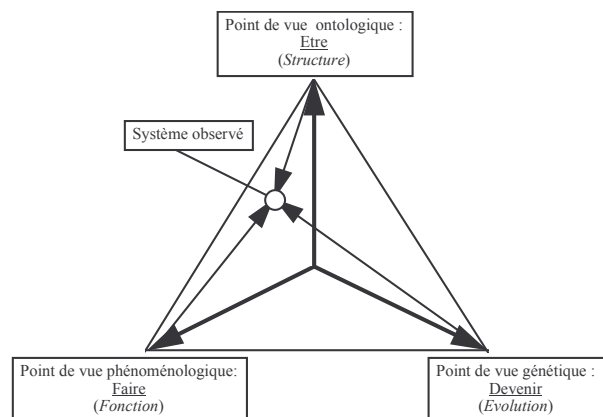


Figure 2 : Les trois axes du triangle systémique

C'est l'hypothèse sémiotique : ces éléments sont désignés sous le terme général de signes, et donc un système de connaissances est perçu comme un *système de signes*. Tout phénomène perceptible (signe) s'observe selon trois niveaux indissociables : le référent ou signe (la manifestation), le signifié (la désignation), le signifiant (le sens), ou encore se perçoit selon trois dimensions : syntaxique, sémantique, pragmatique. Ainsi donc, tout système doit être vu comme la conjonction de trois points de vue liés. Ces trois points de vue ont été nommés de bien des façons. Nous retiendrons, par convention terminologique, les trois termes : syntaxique, sémantique et pragmatique qui se représentent dans un triangle appelé triangle sémiotique symbolisé par la figure 1.

2.2 Le triangle systémique

La seconde hypothèse forte à la base de MKSM est que le patrimoine de connaissances d'une organisation est un *système* au sens de la théorie du système général [LeMoigne77] : "un système se définit comme un objet *actif* et *stable*, *évoluant* dans un *environnement* et par rapport à quelque *finalité*". Cette définition du système général aboutit, comme pour la sémiotique, à une « triangulation » symbolisée par la figure 2. Un système s'observe selon trois points de vue indissociables. Un premier point de vue dit *ontologique* considère le système dans sa *structure*, en tant qu'il est perçu comme un ensemble d'objets agencés : c'est le point de vue de l'*être* du système. Un second point de vue dit *phénoménologique* (ou *fonctionnel*), considère le système dans sa *fonction*, en tant qu'il est perçu comme agissant : c'est le point de vue du *faire* du système. Un troisième point de vue dit *génétique*, considère le système dans son *évolution*, en tant qu'il est perçu comme se modifiant au cours du temps

en accord avec son projet : c'est le point de vue du *devenir* du système. Là encore, on choisit par convention la terminologie : structure, fonction et évolution.

2.3 La structure d'analyse d'un système

La dernière hypothèse qui est le fondement de MKSM découle directement des deux premières, et fournit le microscope qui permet d'appréhender globalement un patrimoine de connaissances dans une organisation. Puisqu'un système est perçu comme un système de signes et comme un système général, avec chacun trois points de vue, la perception globale qu'on peut en avoir est le produit de ces deux types d'approche, qui fournit donc neuf points de vue possibles : un point de vue syntaxique, sémantique et pragmatique, chacun étant décomposé en trois autres points de vue : ontologique, phénoménologique et génétique (structure, fonction, évolution). Cet ensemble de neuf points de vue sera considéré désormais comme notre microscope permettant d'aborder, voire de modéliser, la complexité des systèmes que nous aurons à étudier et à gérer.

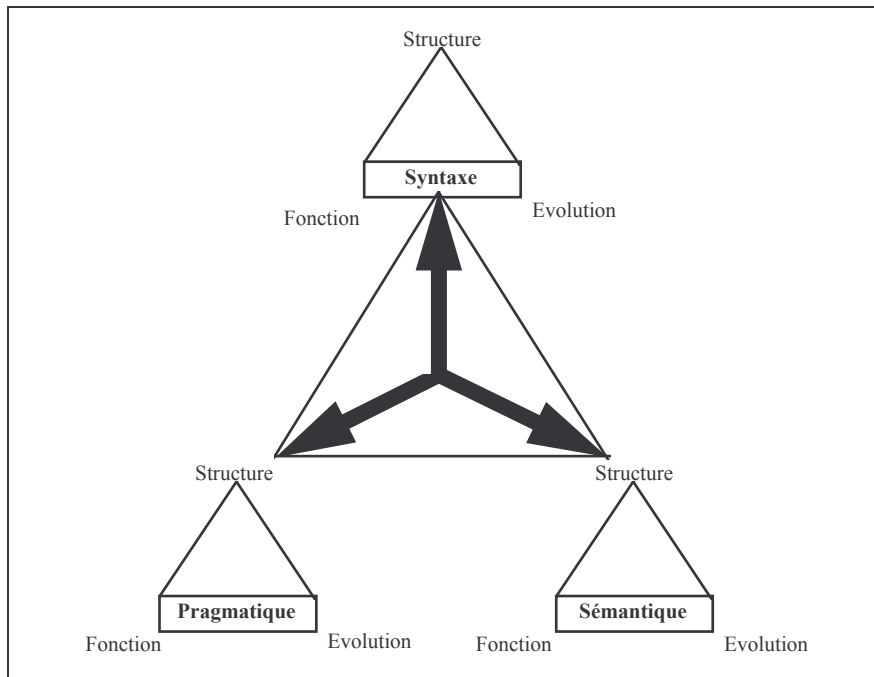


Figure 3 : Les neuf points de vue sur un système

2.4 Le microscope de la connaissance

Cette décomposition d'un système en neuf points de vue, qui défissent un microscope pour les systèmes complexes généraux, est la base de la méthode MKSM. Il reste à faire les hypothèses qui permettent d'interpréter et d'adapter ce microscope à la problématique de la connaissance.

La première hypothèse est que l'aspect syntaxique de la connaissance concerne l'*information*. Nous considérons que l'information est la partie visible, la mise en forme de la connaissance, au même titre que l'orthographe ou la grammaire concernent la partie visible du langage. C'est donc le point de vue qui s'occupe de la forme sous laquelle se traduit la connaissance, du code qu'elle utilise pour *prendre forme*.

La seconde hypothèse est que l'aspect sémantique de la connaissance concerne la *signification* de l'information, qui est bien évidemment distincte de sa forme. Il ne suffit pas d'accumuler des données sur une connaissance. Encore faut-il y adjoindre d'une manière ou d'une autre le sens de ces données pour obtenir quelque chose un tant soit peu pertinent. L'accumulation d'information (au sens brut), ne fait pas plus de la connaissance que l'accumulation de briques ne fait un mur! C'est donc ce point de vue qui s'occupe du fond par rapport à la forme de la connaissance, de la structure qu'elle utilise pour *prendre sens*.

La troisième hypothèse est que l'aspect pragmatique de la connaissance concerne le *contexte* dans lequel le sens dont on vient de parler se met en place et qui, comme on le devine aisément, influe fortement sur cette composante. Une connaissance n'existe pas seulement parce qu'elle a une forme et une signification donnée, mais aussi parce que cette forme et cette signification sont données dans un milieu qui lui donne sa richesse et sa pertinence. C'est donc ce point de vue qui s'occupe du système, de l'environnement que la connaissance utilise pour *se mettre en contexte*.

Un système de connaissances est donc vu, à travers notre microscope, comme de l'information qui prend une certaine signification dans un contexte donné. C'est ce qui est symbolisé dans la figure 4.

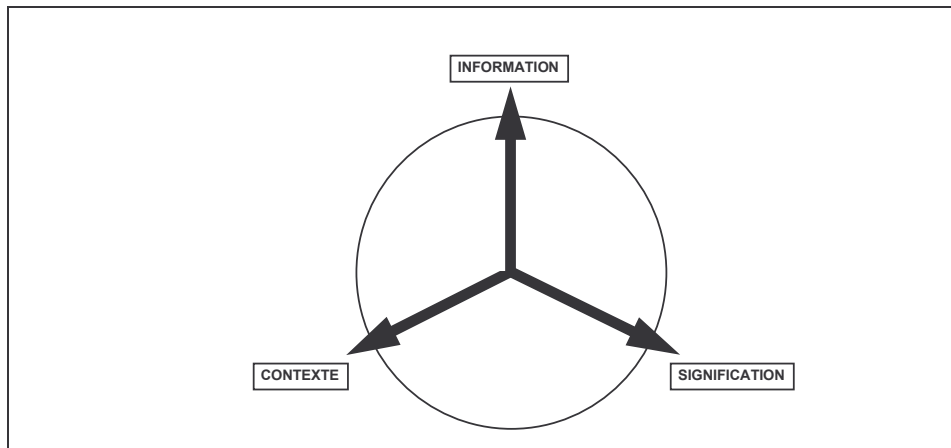


Figure 4 : Les trois axes de la connaissance,
traduction du triangle sémiotique pour les systèmes de connaissances

3. Présentation de la méthode MKSM

Il est possible d'utiliser ce microscope de différentes manières, chaque point de vue étant traité avec des techniques qui dépendent des objectifs recherchés et du domaine d'application.

3.1 Connaissance et information

La définition précise, mathématique de l'information existe. Elle a été donnée par Shannon en 1949 [Shannon 49], et constitue ce qu'on appelle la théorie de l'information. Elle fournit une définition formelle, quantifiée du code qu'utilise la connaissance pour se mettre sous forme de *données informationnelles* (on utilisera le terme simplifié de *données*, malheureusement lui aussi très connoté). L'aspect fonctionnel de l'information concerne le *traitement de l'information*. Il décrit comment on peut manipuler les données. Les aspects structurels et fonctionnels de l'information sont en parfaite dualité. MKSM ne fixe pas de choix pour aborder ce point de vue. Pour fixer un cadre concret, lorsqu'aucune solution n'est imposée, c'est la méthode OMT (voir §4) qui est utilisée.

3.2 Connaissance et signification

La signification aborde le problème du sens. L'aspect structurel s'attache à définir la nature du sens. Ce point de vue est celui des langages de représentation des connaissances. Il se traduit par des *réseaux sémantiques* (des données sémantiques, par opposition aux données informationnelles). L'aspect fonctionnel aborde le traitement des données sémantiques. Les structures sémantiques n'existent pas dans l'abstrait, elles sont construites parce qu'elles doivent être utilisées dans un certain but, dans une certaine action. Cette action est caractérisée par un problème à résoudre, un objectif, et est décrite par une stratégie construite par l'esprit humain afin de résoudre ce problème. C'est pour cela que les données sémantiques sont utilisées par des *tâches cognitives* (on utilisera là encore le terme simplifié de *tâches*), qui sont des méthodes de résolution de problèmes.

3.3 Connaissance et contexte

Le contexte aborde la connaissance en tant qu'elle est intégrée à un système, au sens du système général. Il convient alors de représenter ce système pour mettre la connaissance en contexte. L'aspect structurel du contexte concerne le *domaine* de la connaissance. Pour en avoir une vision globale, le domaine est représenté intégré au système qui produit les connaissances, puis comme un système général « tapissé de processus ». C'est cet ensemble de processus identifiés qui permet de définir une ontologie raisonnée (ou un « glossaire ») du domaine. L'aspect fonctionnel du contexte est une étude fonctionnelle du système de connaissances. Il consiste à identifier ou définir les *activités* qu'il contient et leurs liaisons en terme d'échanges (de données le plus souvent).

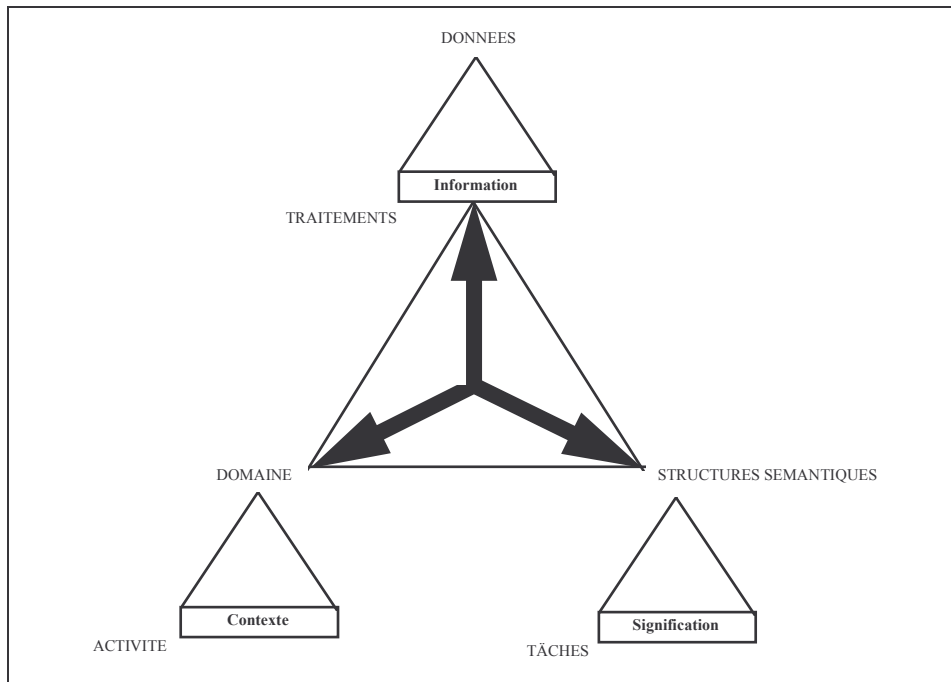


Figure 5 : Le macroscopie de la connaissance dans MKSM

On remarquera une absence de taille dans ce qui vient d'être décrit, à savoir la composante « évolution » de la connaissance à laquelle la méthode MKSM n'apporte pas de réponse complète. Il n'est, à notre connaissance du moins, pas possible de fournir un modèle pertinent de l'évolution des systèmes de connaissances, de leur progression à travers le temps et de la naissance des connaissances jusqu'à leur disparition. Et ceci que ce soit dans la composante informationnelle, sémantique ou pragmatique. Des solutions partielles sont envisageables (comme les systèmes de maintenance, de traçabilité...), mais il n'y a pas de vision théorique globale. Les recherches en cours, notamment en sciences cognitives, permettront à terme de lever cette lacune. Mais il est trop tôt pour parler de modèles « d'évolution » validés et opérationnels.

En résumé, MKSM s'articule autour du macroscopie symbolisé dans la figure 5. La méthode prend en compte six points de vue (sur neuf possibles). Le point de vue de l'information est traité par la méthode de génie logiciel OMT (*Object Modelling Technique*). Le point de vue sémantique, que nous appellerons aussi cognitif, concerne une discipline moins répandue qui est le *génie cognitif* (MKSM intègre la méthode de génie cognitif MOISE [Ermine 93] proche de la méthodologie classique KADS), et le point de vue pragmatique concerne l'*analyse systémique*, et plus particulièrement l'*analyse fonctionnelle*.

4. Intégration de la méthode OMT à la méthode MKSM

La méthode MKSM une fois appliquée sur une organisation donne une vue d'ensemble sur celle-ci qui permet de déterminer en quoi des outils informatiques apporteraient une aide intéressante. On a alors le choix entre la mise en place d'outils déjà commercialisés, une adaptation de l'existant par une approche d'intégration et de développement verticale, ou enfin le développement de logiciels spécifiques. Dans ce dernier cas, notre soucis est de mettre en oeuvre une méthode de génie logiciel afin d'assurer la qualité des développements dans le cadre d'une relation client fournisseur. La méthode OMT comme standard industriel émergent parmi les différentes méthodes orientées objet s'avère pour nous le meilleur choix. Afin d'intégrer OMT à la démarche de MKSM nous avons étudié l'existence de liens entre les modèles des deux méthodes. Cette démarche nous a amené à établir une continuité entre les deux méthodes qui positionne l'ingénierie de la connaissance comme un complément méthodologique intéressant pour le génie logiciel.

4.1 Phase d'analyse de OMT : la mise en contexte du logiciel vu comme un système

Selon l'approche OMT, la phase d'analyse consiste à décrire le comportement externe du système (le logiciel) vu comme une boîte noire [Rumbaugh95].

Dans un premier temps, il est primordial de bien comprendre les besoins des utilisateurs et des experts afin de s'assurer que l'on va résoudre les bons problèmes. Il faut pour cela spécifier les besoins de telle façon que les utilisateurs puissent comprendre et valider les spécifications. Les dernières évolutions de OMT mettent dans ce but l'accent sur une approche orientée utilisateur, basée sur les uses cases de Jacobson [Rumbaugh94]. L'identification des besoins se fait par la description des séquences d'interactions possibles (les use case) entre le

système et les acteurs, qui représentent les entités extérieures au système et en interaction directe avec lui. Ces acteurs sont soit des être humains, soit des systèmes physiques. Un acteur est plus précisément une entité extérieure assurant un rôle particulier : une même personne peut par exemple être perçue à la fois comme un administrateur de base de données, et comme un simple utilisateur. Ces séquences d'interaction vont permettre d'identifier les informations échangées et les fonctionnalités demandées.

Dans un deuxième temps, on construit le modèle du domaine et le modèle de l'application. Le modèle du domaine modélise l'expertise sur le/les problèmes traités par le système en termes compréhensibles par les experts du domaine. Les classes d'objet de ce modèle sont indépendantes d'une application particulière et n'ont donc pas à être déterminées à partir des use cases. Le modèle de l'application est propre au système étudié. Les classes du modèle d'application n'ont aucune existence indépendamment de l'application, mais sont vues par les acteurs (comme par exemple les classes de l'interface graphique) : elles peuvent être identifiées par les use cases. J. Rumbaugh résume les différentes influences qui concourent à la construction du modèle d'analyse dans la figure 6.

Le modèle d'analyse complet présente donc l'application étudiée comme un modèle du domaine donné par les experts du domaine, sur lequel s'appuie le modèle de l'application qui prend en charge les interactions du système avec son environnement (figure 7).

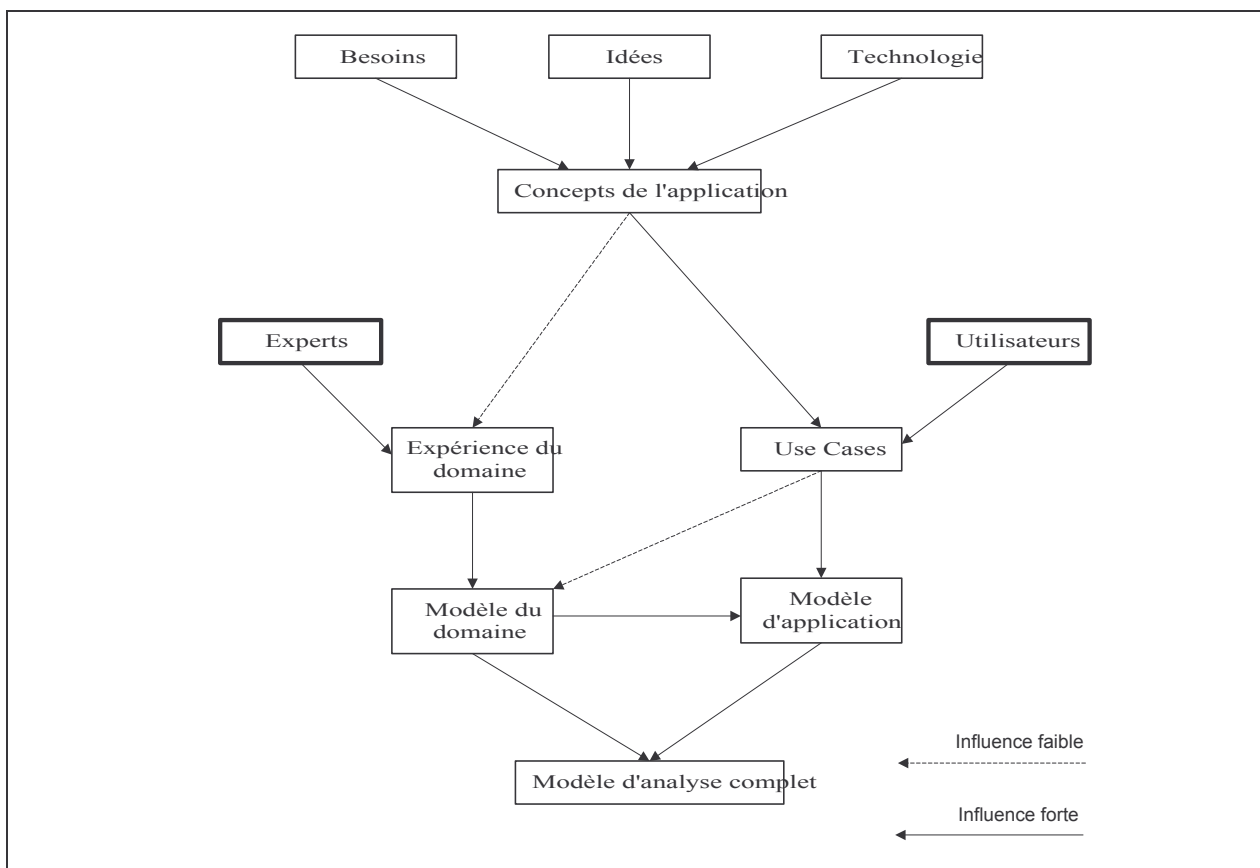


Figure 6 : influences participant à la construction du modèle d'analyse de OMT [Rumbaugh94].

4.2 La méthode MKSM : point de départ pour la phase d'analyse de OMT

Si nous revenons à la figure 6, on s'aperçoit que l'analyse s'appuie sur les connaissances des experts du domaine, et sur un dialogue avec les utilisateurs. Mais OMT ne donne aucune indication méthodologique pour identifier les utilisateurs et modéliser l'expérience du domaine. Ces deux aspects présentent pourtant des difficultés lorsque l'on étudie des logiciels qui s'adressent à des utilisateurs nombreux et aux compétences variées, et qui intègrent le savoir faire de plusieurs experts. MKSM propose alors un cadre méthodologique dont les modèles servent de point de départ à la phase d'analyse de OMT.

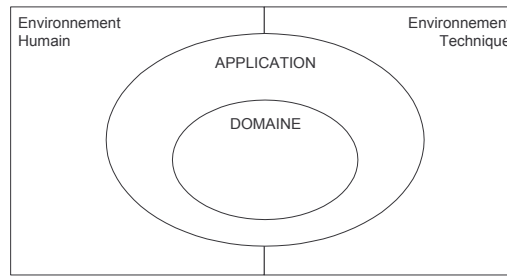


Figure 7 : un logiciel vu comme un système en interaction avec son environnement.

Lien avec les modèles d'activité : aide à la mise en contexte. Les modèles d'activité obtenus par la méthode MKSM recensent les individus participant aux grandes fonctions d'une organisation et les ressources utilisées. Or les logiciels étudiés vont servir de support à une ou plusieurs de ces fonctions. Ces modèles nous permettent alors d'identifier les utilisateurs potentiels d'un logiciel, ainsi que leur rôle par rapport à celui-ci : on en déduit donc les différents acteurs humains en interaction avec le système. Les ressources explicitent alors les contraintes techniques à prendre en compte : on obtient ainsi les acteurs techniques imposés par l'environnement. En plus de cette relation très forte avec OMT, on constate que le modèle d'activité aide le concepteur à expliciter ce que Rumbaugh appelle *les concepts de l'application* (cf figure 6). En particulier la première définition des besoins est grandement facilitée par la connaissance du contexte dans lequel l'application rendra des services. Nous ne développeront cependant pas ce point car notre expérience ne nous permet pas de préciser la nature de cette aide.

Lien avec les modèles statiques et dynamiques de la connaissance : modèle du domaine. MKSM est évidemment parfaitement adaptée à la construction d'un modèle d'expertise. Les réseaux sémantiques qui modélisent les concepts du domaine d'expertise se traduisent directement en classes OMT. Le passage d'un modèle à l'autre est déjà largement abordé dans la littérature et des outils commerciaux tel que Ilog Server fournissent même des solutions d'implémentation pour les relations entre les classes. Le modèle objet du domaine peut donc être déduit directement du ou des modèles d'expertise de MKSM.

Les modèles de tâches décrivent les méthodes de résolution de problèmes utilisées par les experts du domaine, et comment les informations décrites par les réseaux sémantiques sont exploitées. Ils explicitent de ce fait les grandes fonctionnalités que le système devra éventuellement proposer aux utilisateurs. Les tâches permettent donc de construire les bases du modèle fonctionnel du domaine de OMT. Cette approche informelle permet d'intégrer une expertise dans un cycle de développement classique où les méthodes de résolution définies avec les experts peuvent être codées en dur.

Aide à la conception de l'ergonomie. Les tâches utilisées dans la méthode MKSM ont pour origine des recherches en ergonomie cognitive, et sont donc par nature adaptées à l'aide à la spécification de l'IHM du système. En plus de l'identification des principales fonctionnalités, elles constituent un guide dans la conception de l'interface en apportant des critères de choix pour l'ergonomie [Scapin90], [Sébillote94].

5. Propositions pour une approche systémique de OMT

L'intégration de OMT dans une méthode globale de gestion des connaissances telle que MKSM nous a amené à une réflexion sur le découpage en phases indépendantes couramment utilisé dans les méthodes de génie logiciel. Les applications envisagées pour assurer un support de la gestion des connaissances au sein d'une organisation sont souvent simples dans leur principe mais difficiles à réaliser. Les modèles OMT proposés dans les différentes phases apportent une aide précieuse dans la conception du logiciel. Mais le principe même de la répartition en phases soulève des difficultés d'ordre conceptuelle. En s'inspirant de l'approche systémique de MKSM, nous proposons donc une approche différente applicable aux méthodes de génie logiciel orientées objet. Cette approche est ensuite appliquée pour remettre en perspective les différents modèles de OMT.

5.1 Le logiciel vu comme un système complexe

La méthode OMT conserve les trois phases considérées comme classiques : Analyse, Conception, Implémentation. L'idée fondamentale consiste dans cette approche à séparer la spécification du *Quoi* de la description du *Comment*, et à différencier le codage de la spécification. Cependant, dans la pratique, la frontière entre le quoi et le comment s'avère très difficile à déterminer, et le découpage en phases indépendantes n'est pas forcément toujours conceptuellement satisfaisant.

Ainsi James Rumbaugh émet des réserves sur le découpage en phases successives. Il explique ce choix en partie par la nécessité d'opter pour une présentation linéaire de sa démarche mais ne s'en satisfait pas : *gardez à l'esprit que le développement est (un processus) aussi bien itératif que récursif. Il a une nature fractale difficile à décrire en utilisant un média linéaire tel qu'un article ou un livre.* [Rumbaught95]. Dans le même article il dit

clairement qu'il n'est pas habituellement nécessaire de maintenir une séparation nette entre l'analyse et la conception. Cette problématique est aussi évoquée par Bertrand Meyer dans [Meyer96]. D'après lui, un des avantages des méthodes et langages orientés objets réside dans la possibilité qu'ils nous apportent d'atténuer, voir même de supprimer les distinctions habituelles entre les phases d'analyse, de conception et d'implémentation, qui posent toujours des problèmes qu'il qualifie d'adaptation d'impédance. Ce terme d'électronicien évoque les difficultés de passage d'une phase à l'autre. Pour lui, les technologies objet sont plus fondamentalement un ensemble d'outils pour penser/réfléchir sur des systèmes complexes.

Cette notion de complexité est à notre avis au centre des préoccupations des méthodes de génie logiciel orientées objet. L'approche objet est en effet reconnue aujourd'hui comme un paradigme incontournable pour concevoir certaines catégories de logiciels : on considère implicitement que l'approche objet est indispensable lorsque l'application développée franchit un certain degré de complexité. L'approche objet est en effet considérée comme plus adaptée que l'approche fonctionnelle pour modéliser le monde réel. Elle permet ainsi une meilleur adéquation lorsque les logiciels abordent des problèmes plus complexes et moins formalisés. Mais les applications qui ont en premier largement bénéficié de la programmation par objet sont celles où les interfaces graphiques et donc les besoins d'interaction étaient prédominants. On retrouve en cela la problématique des systèmes complexes dont on donnera deux définitions :

- *La complexité d'un système vient plutôt de ce que son évolution dépend d'un grand nombre de degrés de liberté et de leurs interactions avec un environnement évolutif.* [Erceau96]
- *La complexité d'un système est caractérisée par deux facteurs : le nombre des éléments qui le constitue d'une part et le nombre des inter-relations d'autre part.* [Le Moigne90]

La deuxième définition est de plus caractéristique d'un logiciel objet. Un logiciel objet est le résultat de la composition d'un ensemble d'objets. Mais les interactions entre ces objets sont au moins aussi importantes que les objets pris individuellement pour expliquer le comportement global du logiciel. Il semble alors intéressant d'aborder le logiciel comme un système complexe sur lequel on peut appliquer les raisonnements systémiques.

Une méthode de génie logiciel a pour but de donner des outils intellectuels pour faire face à la complexité des logiciels. Elle donne un cadre conceptuel qui aide dans la démarche de conception en autorisant une approche plus abstraite des problèmes abordés. Pour aborder le génie logiciel sous l'angle systémique on reprend le modèle du microscope de MKSM. Le logiciel vu comme un système est selon nous un sous-système du système d'information. En suivant la démarche de MKSM, on choisit de considérer une méthode de génie logicielle orientée objet comme un outil conceptuel qui permet de modéliser les connaissances construites au fur et à mesure de l'étude du logiciel. Vu sous cet angle le découpage en phases successives et indépendantes est remplacé par un autre microscope qui va nous permettre d'organiser une méthode de génie logiciel en 9 points de vue. Contrairement à l'approche en phases on ne veut plus découper le raisonnement en étapes indépendantes, mais donner plusieurs fenêtres d'observation sur un même système afin d'en appréhender le fonctionnement malgré sa complexité. On ne cherche plus à progresser linéairement de l'analyse vers la conception, mais à compléter mutuellement et parallèlement plusieurs points de vue. L'ensemble de ces points de vue permettent par recoupement d'assimiler tous les aspects du système dans toute sa complexité.

5.2 Le microscope : une approche systémique du génie logiciel.

Partons maintenant du principe que l'on aborde un logiciel comme un système complexe, et qu'une méthode de génie logiciel orientée objet peut être perçue comme la construction d'un système de connaissances sur un système complexe. On peut alors tenter de définir un microscope pour une méthode de génie logicielle orientée objet quelconque. On verra ensuite comment aborder la méthode OMT suivant cette approche.

Un système de connaissances peut être décomposé en trois points de vue selon le triangle sémiotique (figure 4). La connaissance a dans ce contexte la définition suivante : *La connaissance est un ensemble d'informations qui prennent une signification dans un contexte.*

Voir un logiciel sous cet angle amène alors à une définition similaire pour une méthode de génie logiciel orientée objet : *Une méthode de génie logiciel orientée objet est un ensemble d'outils intellectuels permettant d'explicitier la sémantique du code par rapport au contexte du logiciel.*

Mais on veut maintenant étudier le système logiciel selon ces trois points de vue. On va pour cela appliquer sur chacun d'eux le triangle systémique qui décompose l'étude d'un système en trois points de vue. En final, une méthode de génie logiciel permet de présenter les connaissances sur un logiciel perçu comme un système complexe selon neuf points de vue. On peut maintenant proposer une redistribution des différents modèles de la méthode OMT selon ces neuf points de vue (on garde les noms des phases de OMT pour ne pas les renommer inutilement) :

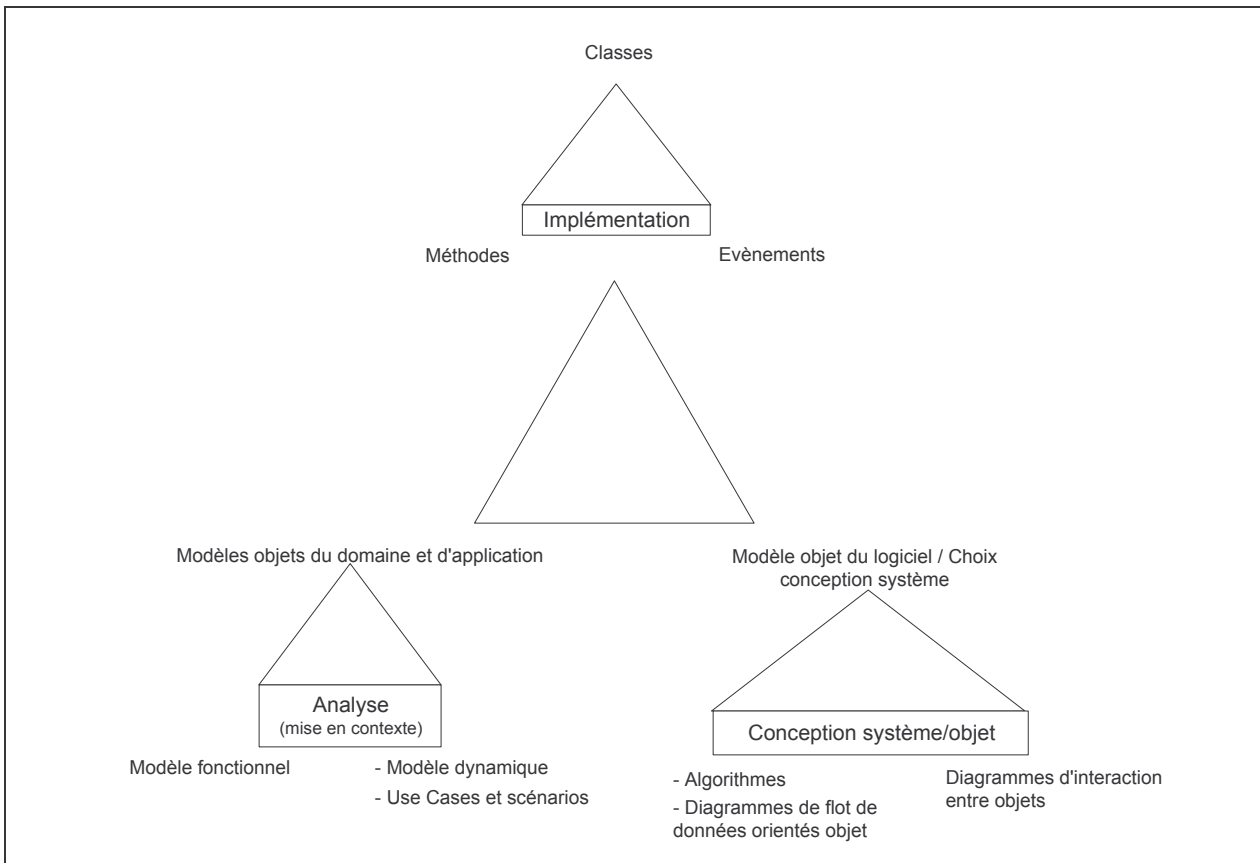


Figure 8 : OMT à travers le microscope.

Il n'est pas étonnant à ce stade de remarquer des similitudes avec OMT. Il ne s'agit pas de remettre en cause des années d'expériences mais de proposer un changement dans le découpage en phases séquentielles indépendantes. Notre but est de positionner les outils proposés par OMT dans une nouvelle perspective qui à notre avis répond à un certain nombre de difficultés ressenties dans l'application de la méthode, et d'améliorer son intégration dans une démarche plus globale de gestion des connaissances.

6. Application aux logiciels scientifiques

MKSM fournit deux perspectives pour la problématique des logiciels scientifiques.

6.1 Un cadre méthodologique pour la modélisation des connaissances

Pour les bibliothèques de composants logiciels, il est délicat d'identifier le module le mieux adapté au problème posé, puis d'en contrôler l'exécution. Nous considérons que l'approche du projet SHERPA ([Rechenmann92], [Chaillot94]) constitue une solution théorique satisfaisante à cette problématique. Nous retenons *Power Classes* (produit commercialisé par Ilog - [Ilog94]) comme un outil opérationnel pour la mettre en œuvre.

Toutefois, cette approche est basée sur l'élaboration d'un modèle objet et d'un modèle de tâches. Notre expérience de gestion des connaissances sur des projets de R&D qui impliquent plusieurs experts nous a montré que la modélisation objet n'est pas suffisante et qu'on ne peut faire l'économie d'une mise en *contexte* du domaine et de l'activité. En effet, les différents experts n'envisagent pas un phénomène physique de la même manière, les intitulés qu'ils utilisent pour les grandeurs physiques sont ambigus etc.. Plus généralement, le sens qu'ils associent à l'information qu'ils partagent dépend du contexte : l'expérimentation, la théorie, la réalisation et la technologie. Aussi, proposons nous d'utiliser MKSM dont le cadre méthodologique est cohérent avec cette démarche : le modèle objet du domaine et le modèle de tâches correspondent respectivement (au formalisme près) aux réseaux sémantiques et aux tâches du point de vue de la signification du microscope. L'utilisation de MKSM consiste alors à rajouter deux étapes en amont des modèles d'expertise : la mise en contexte du domaine par la description des processus physiques pris en compte et la mise en contexte de l'activité métier par l'analyse fonctionnelle.

6.2 Restitution de la sémantique des paramètres des codes

La mise en contexte du domaine et de l'activité métier n'est pas seulement utile l'acquisition des connaissances. Nous proposons de présenter à l'utilisateur les trois points de vue du microscope MKSM. Les données et le logiciel constituent le point de vue information. Les concepts du modèle objet décrivent l'aspect structurel de la signification de ces données, et les tâches décrivent l'aspect fonctionnel de la signification du logiciel. Pour restituer à l'utilisateur toute la connaissance de ce système, il faut donc également lui fournir la mise en contexte : décrire les phénomènes physiques qui sont pris en compte en identifiant les grandeurs physiques utilisées dans les codes sur ces processus ; décrire l'activité qui est concernée par le composant logiciel. Le microscope permet également de positionner les éléments de l'ontologie du domaine et du modèle de tâches. Les concepts sont rattachés aux processus, les tâches sont associées aux fonctions qu'elles permettent de réaliser.

Une application hypermedia [Millerat96] a été réalisée pour exploiter un tel microscope dans le domaine de la conduite des centrales nucléaires. Depuis une tâche donnée (opération de conduite) on peut accéder à la fonction de sûreté qu'elle assure, fonction à partir de laquelle on visualise en rouge les processus qu'elle inhibe et en vert ceux qu'elle déclenche. Inversement, en sélectionnant l'un de ces processus, on accède de la même manière aux fonctions puis aux tâches qui le maîtrisent.

Cette utilisation des modèles de mise en contexte avec les modèles d'expertise (ontologie objet du domaine et modèle de tâches) permet de repositionner ces derniers par rapport aux concepts physiques et métiers qui sont familiers à l'utilisateur. Nous travaillons actuellement à l'application de cette approche pour gérer les paramètres d'appel des logiciels scientifiques pour aider le physicien à valuer les concepts du flot d'entrées/sorties entre les tâches.

Conclusion

Nous avons tenté de montrer comment l'approche MKSM constitue un complément intéressant de la modélisation objet. Pour ce faire nous nous sommes appuyés sur un cadre théorique général : le microscope de la connaissance. Nous avons ensuite abordé les aspects techniques de l'interface entre la méthode MKSM et la méthode OMT et nous avons donné les perspectives actuelles de nos travaux par rapport aux logiciels scientifiques.

Il ressort de notre expérience que les modèles de mise en contexte (processus du domaine et analyse fonctionnelle de l'activité) qui sont préconisés par MKSM sont très utiles comme prérequis pour aborder le travail de construction d'une ontologie du domaine avec un modèle objet. À partir de ce constat, il nous semble que leur utilisation est également intéressante pour restituer cette modélisation aux utilisateurs des logiciels scientifiques. Nous nous employons donc à concrétiser cette approche pour laquelle certains points restent en suspend : en particulier, comment positionner dans ce cadre théorique le mécanisme d'inférence des systèmes classificatoires.

Références

- [Chaillot94] *Sélection rationnelle de méthodes dans un environnement réactif de résolution de problèmes*. M. Chaillot, E. Crochon. 9^{ème} congrès RFIA, Paris, janvier 1994, pp 293-303.
- [Ducournau95] *Les systèmes classificatoires*. R. Ducournau. Rapport de recherche. LIRM Université de Montpellier II. RR-LIRM-95038. 1995.
- [DeRosnay75] *Le microscope, vers une vision globale*, De Rosnay J. Collection Points, Le Seuil, Paris, 1975
- [Erceau95] *Complexité et conception des systèmes complexes, vers une ingénierie des systèmes complexes*. J. Erceau. CP2I, Journée « La complexité », juillet 1995.
- [Ermine93] J-L Ermine : *Génie logiciel et génie cognitif pour les systèmes à base de connaissances*, Collection Tec et Doc, Lavoisier (1993)
- [Ermine96a] *MKSM, méthode de gestion des connaissances*. J-L Ermine, M. chaillot, Ph. Bigeon, B. Charreton, D. Malavieille. À paraître.
- [Ermine96b] *Les systèmes de connaissances*. J-L. Ermine. Hermès, à paraître.
- [Euzenat94] *KR and OOL co-operation based on semantics non reducibility*. J. Euzenat. Proc 11th ECAI Workshop on integrating object-orientation and knowledge representation, Amsterdam (NL), 1994.
- [Ilog94] *Ilog Power Classes : Reference manual 1.3*. Ilog SA. 1994.
- [LeMoigne77] *La théorie du Système Général, théorie de la modélisation*. J-L Le Moigne. P.U.F., Paris, 1977, 3^{ème} édition mise à jour, 1990
- [LeMoigne90] *La modélisation des systèmes complexes*, J-L Le Moigne. Afcet Systèmes, Dunod, Paris, 1990.
- [Meyer96] *The conceptual perspective*. B. Meyer. *Computer*, Janvier 1996.
- [Millerat96] *Knowledge management for modelling nuclear power plants control in incidental and accidental states* P. Millerat, J-L Ermine, M. Chaillot. À paraître dans *Computational Engineering in System Applications*, 1996.
- [Rechenmann92] *A development shell for knowledge-based pde solver*. F. Rechenmann, B. Rousseau. E.N. Houstis, J.R. Rice, *Expert systems in scientific computing*. Elsevier Science Publishers. 1992.
- [Rumbaugh94] *Getting started. Using uses cases to capture requirements*. J. Rumbaugh. *Journal of Object Oriented Programming*, Vol. 7, No. 5, p8-12 et p23, Septembre 1994.
- [Rumbaugh95] *OMT : The development process*. J. Rumbaugh. *Journal of Object Oriented Programming*, Vol. 8, No. 2, p8-16 et p76, Mai 1995.

- [Scapin90] "Organising Human Factor Knowledge for the Evaluation and Design of Interfaces". D.L. Scapin. *International Journal of Human-Computer Interaction*, Vol. 2, No. 3, p203-229, 1990.
- [Shannon49] *The Mathematical Theory of Information*, Shannon C. E., Weaver W. Urbana, University of Illinois Press, 1949, trad. : Théorie mathématique de la communication, Retz - CEPL, Paris , 1975
- [Sebillote94] *From users' task knowledge to high level interface specification*. S. Sebillote. *International Journal of Human-Computer Interaction*, Vol. 6, No. 1, 1994.