

Chapitre 10

Ingénierie des connaissances pour les systèmes d'information*

10.1. Introduction

L'Ingénierie des connaissances constitue depuis de nombreuses années un domaine actif des recherches menées en intelligence artificielle autour de la conception et de la réalisation des systèmes à base de connaissances (SBC). À l'instar de bien d'autres disciplines modélisatrices, elle consiste à concevoir des systèmes dont le fonctionnement permet d'opérationnaliser des connaissances portant sur le traitement ou la résolution d'un problème donné. La résolution (semi-)automatique de problèmes implique deux étapes essentielles : la modélisation du problème et d'une méthode de résolution dans un cadre théorique donné, l'opérationnalisation informatique du modèle obtenu.

Longtemps, le cadre théorique de l'Ingénierie des connaissances fut celui de l'acquisition des connaissances : modélisation psychologique ou empirique des connaissances d'un expert dans le but de les coder dans un système expert (SE). La période actuelle se concentre davantage sur la modélisation conceptuelle du monde : on tente, le plus souvent à partir d'une formulation linguistique du problème –

* Chapitre rédigé par Jean Charlet, Chantal Reynaud et Régine Teulier.

transcriptions d'interviews d'experts, descriptions techniques, notices de maintenance, etc. –, d'élaborer une représentation qualitative et formelle du problème. Il arrive également qu'on obtienne une première représentation du problème à partir d'observations de l'activité faites, par exemple, avec des méthodes issues de l'ergonomie ou de l'ethnométhodologie¹. Définie en ces termes, il paraît naturel qu'une recherche menée sur les SBC comprenne une activité correspondant à l'Ingénierie des connaissances. En effet, il s'agit de définir quelles connaissances sont pertinentes, pour les exploiter en formulant un modèle qualitatif et formel du problème. Classiquement, l'acquisition des connaissances propose des méthodes permettant de modéliser des raisonnements en faisant abstraction du domaine d'activité et en se concentrant sur la façon dont est conduit ce raisonnement – *i. e.* modélisation au « niveau des connaissances » –, des outils pour les opérationnaliser au « niveau des programmes » ou niveau symbolique. Un modèle conceptuel² du problème doit être compréhensible et réappropriable par un spécialiste du domaine³. Un modèle opérationnel, défini au niveau des programmes, traduit en termes algorithmiques les connaissances formalisées du modèle conceptuel. Ce dernier est à la fois la spécification et l'instrument de lecture du système opérationnel.

Les recherches en Ingénierie des connaissances ayant évolué ces dernières années, elles concernent aujourd'hui tout système informatique utilisant des connaissances *pour peu que ces connaissances soient explicitement modélisées en tant que telles*. Par ailleurs, les travaux développés en ce domaine tirent parti d'apports d'autres domaines comme le génie

¹ Le recours au texte, ou corpus, comme matière première n'est pas la pensée unique de l'Ingénierie des connaissances : l'ergonomie, la sociologie sont des vecteurs d'entrée parmi d'autres. Mais deux constats doivent être faits : (a) la formulation linguistique fournit une description qualitative du monde qui contribue à déterminer, en contexte, le contenu de l'information véhiculée (cf. 10.5.4) ; (b) un certain nombre d'avancées méthodologiques dans ce domaine font de l'acquisition des connaissances à partir de corpus une des directions de travail prometteuse de l'Ingénierie des connaissances (cf. 10.3.5).

² Défini comme la description abstraite du comportement d'un système, en termes de *domaine*, *méthodes* de raisonnement, *tâche* dévolue au système (cf. 10.3.1), ce modèle conceptuel recouvre donc une réalité différente de celle communément admise en Systèmes d'information.

³ Il y a une certaine contradiction entre un modèle compréhensible par un expert du domaine et un raisonnement modélisé à un niveau générique où il est fait abstraction du domaine. Ce problème est à l'origine de bien des incompréhensions des experts devant les modèles « génériques » de l'Ingénierie des connaissances. Des réponses sont recherchées dans diverses adaptations des modèles génériques et dans des modèles intermédiaires tels les bases de connaissances terminologiques (cf. 10.3.5).

logiciel et, inversement, les réflexions et techniques d'Ingénierie des connaissances mises au point dans le cadre de la conception de SBC, sont réutilisables dans des contextes plus larges, en particulier au sein des systèmes d'information (*cf.* chapitre 5). Ainsi, tout système, *a fortiori* système d'information, qui manipule et transporte des informations destinées *in fine* à être interprétées par des humains est candidat à être conceptualisé et modélisé avec les méthodes de l'Ingénierie des connaissances. Dans ce cadre, le thème de la réutilisabilité est central pour les systèmes d'information comme pour les SBC. En effet, une discipline qui se veut une ingénierie – des connaissances ou des systèmes d'information – doit apporter, aux utilisateurs, des réponses en termes de méthodes et techniques reproductibles. Pour cette problématique, comme pour d'autres, nous pensons qu'il est nécessaire de revenir sur les fondements des SBC pour tirer tous les bénéfices d'un travail de réflexion à la frontière de deux domaines de recherche.

La 2^e section de ce chapitre nous permettra d'aborder des problèmes épistémologiques de l'Ingénierie des connaissances. La 3^e section nous permettra de nous focaliser sur les acquis de la recherche en Ingénierie des connaissances, en particulier les méthodes de modélisation, puis d'aborder le thème de la réutilisabilité. La 4^e section abordera l'Ingénierie des connaissances d'un point de vue organisationnel et étudiera les moyens que l'on peut mettre en œuvre pour expliciter les processus de traitement de l'information, par l'acteur comme par l'organisation. La 5^e section abordera alors les synergies que l'on peut créer à la croisée de l'Ingénierie des connaissances et de l'ingénierie des systèmes d'information. Nous terminerons sur les perspectives conceptuelles et méthodologiques de telles démarches.

10.2. Épistémologie et définitions

L'Ingénierie des connaissances se situe au carrefour de plusieurs réflexions :

- la linguistique pour étudier la formulation linguistique des connaissances ;
- la terminologie et l'ontologie pour dégager les concepts ;
- la psychologie pour élaborer les méthodes d'élicitation et les modèles d'assistance ; la logique pour élaborer les modèles formels ;
- l'informatique pour les opérationnaliser ;
- l'ergonomie pour interpréter et s'appropriier le comportement du système ;
- la gestion pour concevoir et replacer les systèmes dans leur environnement organisationnel, etc.

La plupart de ces disciplines ressortissent à la sémiotique, la science des signes⁴. Elles se fécondent réciproquement et conduisent à faire évoluer le paradigme. Nous allons donc mobiliser un certain nombre d'entre elles pour aborder successivement les concepts de connaissances, de SBC et de système d'information.

10.2.1. De l'information à la connaissance

Définir la connaissance en soi est une entreprise philosophique qui n'est pas de notre propos ici. Par contre, nous allons essayer de caractériser cette connaissance, d'un point de vue effectif, qui est celui qui nous intéresse. Nous l'aborderons d'abord d'un point de vue dit « épistémologique » puis d'un point de vue « systémique ». Il n'y a pas un point de vue supérieur à l'autre dans notre propos ni de découpage aussi clair à faire entre les caractéristiques qui seront discutées : nous les séparons plutôt en fonction des domaines, Ingénierie des connaissances (et donc Intelligence artificielle) d'un côté, Systèmes d'information de l'autre, qui ont historiquement exploité ces caractéristiques. Enfin, si les caractéristiques attestées ici traduisent des fondements bien acceptés par les deux communautés, elles ne sont pas exactement en accord avec d'autres courants de l'Intelligence artificielle qui font l'hypothèse d'une pensée structurée formellement ou computationnellement. Pour des débats et arguments à ce sujet, nous renvoyons le lecteur à [Bachimont 1996].

10.2.1.1 Point de vue épistémologique

D'un point de vue épistémologique, nous retiendrons trois caractéristiques de la connaissance que l'on peut résumer ainsi [Bachimont 1996, Ganascia 1998, Kayser 1997] :

1. La connaissance est prise ici dans un sens technique rendu nécessaire par le fait que dans le domaine de l'intelligence artificielle, nous cherchons *in fine* à construire des artefacts informatiques.

⁴ La sémiotique est comprise ici dans une acceptation très large et, dans cette hypothèse, il n'est pas étonnant de la voir englober des disciplines qui participent d'une science, l'ingénierie des connaissances, dont le but est d'aider à construire des systèmes informatiques donc systèmes de symboles. Seule, parmi les disciplines citées ici, la gestion se situe dans un cadre applicatif. Elle apparaît ici comme l'aune à laquelle seront mesurés les résultats obtenus, dans l'aptitude qu'auront les SBC construits à s'insérer dans l'environnement organisationnel pour lequel ils ont été conçus (*cf.* 10.5.4).

2. Il y a connaissance et représentation des connaissances quand les manipulations symboliques effectuées par la machine via des programmes, prennent un sens et une justification pour les utilisateurs interagissant avec ces programmes. Les utilisateurs *interprètent* alors le comportement de la machine.

3. La pensée repose sur la médiation externe du signe. La technique (ici l'informatique), par ses outils et ses capacités de mémorisation, permet alors à l'homme de se constituer des connaissances qui évoluent et s'accumulent et n'existeraient pas sans cela⁵.

Ainsi, les deux concepts clés de la connaissance sont *l'interprétation humaine* qui lui donne son existence et, dans notre domaine, *l'outil informatique* qui lui offre son support.

10.2.1.2 Point de vue systémique

D'un point de vue plus « systémique » et parce que les concepts de *données*, *d'information* ont été historiquement abordés par l'informatique et les Systèmes d'information, nous allons nous intéresser à ces concepts, les caractériser les uns par rapport aux autres et constater, qu'ici aussi, il est possible de proposer quelques caractérisations clés qui font consensus [Ermine 1996, Ganascia 1998, Kayser 1988, Le Moigne 1973, Le Moigne 1995, Mèlèse 1990, Poitou 1996, Shannon 1971] et sont respectées par les travaux qui seront cités dans le reste du chapitre. Ainsi, les concepts de *données*, *informations*, *processus*, *connaissances* peuvent être considérés en interactions selon les critères suivants :

a. L'*information* est un concept technique apparu pour les besoins des télécommunications. L'*information* fait appel aux concepts de *codage*, *transmission*, *décodage* et fait référence au nombre potentiel de messages que peut délivrer un système.

b. Une *donnée* est toute information affectant un programme ou un système pour en modifier le comportement. Le *processus* d'utilisation est fixe, au contraire des données qui évoluent et se renouvellent. Mais un programme ou processus peut être lui-même la donnée d'un autre processus.

⁵ On sait que l'écriture a permis de classer les mots en listes, de combiner les listes en tableaux dans un paradigme appelé par J. Goody [1979] *La raison graphique*. De la même manière, d'autres travaux arguent du fait que l'ordinateur permet de se constituer de nouvelles connaissances dans le paradigme de *la raison computationnelle* [Bachimont 1998].

c. Il n'y a pas de frontière donnée/information/processus/connaissance. Nous sommes devant un continuum par rapport à un processus d'action et nous plaçons des étiquettes sur des concepts manipulés par ce processus en fonction des niveaux d'entrée dans celui-ci.

La connaissance peut alors être caractérisée de la façon suivante :

1. Il y a présomption de connaissances, si la faculté d'utiliser l'information à bon escient est attestée. Cette utilisation passe d'abord par une *interprétation* (cf. 10.2.1.1 n°2) puis par une *action*. Ce qui fait dire à P. Lorino [1995] que « l'autonomie d'un acteur est une marge d'*interprétation* pour l'*action* ». Enfin, cette action est d'abord une *réécriture* par l'utilisateur d'un outil, ici l'informatique (cf. 10.2.1.1 n°3).

2. Il y a connaissance quand il y a *contexte* d'utilisation de l'information.

3. La connaissance *a priori* n'existe pas : elle est construite à partir d'un projet propre au modélisateur. Étudier cette construction permet d'essayer d'en dégager des invariants méthodologiques (projet des épistémologies constructivistes [Le Moigne 1995]).

La prise en charge des connaissances par le systèmes d'information va alors être la capacité qu'il aura (*a*) à proposer des données, sources d'*interprétation* par l'utilisateur, (*b*) à prendre en compte et expliciter le *contexte* d'utilisation de ces données et (*c*) à fournir à cet utilisateur les moyens – informatiques – d'*agir* et donc de *réécrire* les résultats de son interprétation. Cette capacité d'interaction homme-machine mettant alors l'accent sur l'utilisateur (savoir, savoir-faire, modes d'interaction) et la prise en compte du *changement* dans l'*action personnelle* et l'*action collective* des organisations.

10.2.2. Des systèmes à base de connaissances

Ces caractéristiques étant admises, il est alors possible d'étendre le champ de nos investigations aux SBC et à leur place dans les organisations. Les SBC manipulent des représentations symboliques selon des prescriptions formalisées lors de la modélisation des connaissances. Ces représentations s'expriment à l'aide de primitives qui renvoient à des notions du domaine, en leur empruntant leur libellé ou terminologie : les primitives sont les termes du domaine⁶. Ces primitives sont manipulées en respectant la grammaire

⁶ Cette affirmation doit être tempérée et surtout précisée : il ne s'agit pas de poser l'équivalence entre les termes du domaine et les primitives conceptuelles qui évoluent dans

du système formel dans lequel elles s'inscrivent. Ainsi, toutes les manipulations effectuées consistent dans la construction syntaxique de représentations mobilisant ces termes : ces représentations peuvent et doivent se rapporter à des expressions interprétables dans le domaine par tout spécialiste. Mais les règles de cette manipulation formelle ne sont pas celles de l'interprétation : le calcul produit des résultats que la rationalité de l'interprétation n'anticipe pas forcément. La combinatoire des expressions suggère alors la possibilité de formuler des inscriptions dont l'interprétation renvoie à de nouvelles connaissances. De même que l'ordinateur ne « voit » pas les images qu'il permet de construire, il ne « pense » pas les nouvelles inscriptions qu'il formule. Mais il permet de voir du nouveau comme de penser autrement (cf. 10.2.1.1, n°3).

Pour concevoir un SBC, il faut donc tenir compte de la manière dont des utilisateurs s'en approprient et lui attribuent du sens en interprétant, sur la foi des primitives empruntées à la terminologie du domaine, les représentations comme des expressions linguistiques de connaissances. Cette interprétation repose sur le contexte d'usage du SBC et son intégration dans un système de pratiques où il prend sens et justification (cf. 10.2.1.1, n°2). Ainsi, il faut voir *les SBC comme des systèmes sémiotiques de manipulation d'inscriptions symboliques, dont le fonctionnement informatique doit permettre à un utilisateur d'interpréter et de comprendre le système dans le cadre de son activité et de ses usages, en utilisant les termes du domaine.*

Un SBC étant un système technique plongé dans un système d'usage, son élaboration est une ingénierie (au sens où un ingénieur élabore un système pour un usage), une ingénierie fondée sur la manipulation de l'inscription symbolique de connaissances. Nous pouvons alors avoir un point de vue opérationnel sur l'Ingénierie des connaissances : *l'Ingénierie des connaissances correspond à l'étude de modèles symboliques ou formels plongés dans des systèmes d'usage ; c'est l'ingénierie informatique et logique de modèles en fonction des usages qu'ils rendent possibles et des appropriations qu'ils permettent.* Ainsi, l'Ingénierie des connaissances peut consister à mettre en perspective des outils formels ou techniques – p. ex. logiques de description, langages de description documentaire (SGML/XML), raisonnement à partir de cas, etc. – avec des concepts d'usage, issus ou non des sciences cognitives – p. ex. le raisonnement classificatoire, la navigation documentaire, l'exploration d'hypothèses dans un domaine donné, etc.

deux registres différents, linguistique et conceptuel. En particulier, les termes du domaine ne peuvent prétendre être instantanément des primitives conceptuelles [Bachimont 2000].

Ces deux points de vue, conceptuel et opérationnel, sont complémentaires et l'Ingénierie des connaissances ne peut mobiliser l'un sans l'autre au risque de perdre sa spécificité, son efficacité et l'objet même de sa recherche : d'un point de vue conceptuel, l'Ingénierie des connaissances trouve l'objet de sa recherche dans des systèmes dont le fonctionnement est interprétable par l'humain en termes de connaissances ; d'un point de vue opérationnel, l'Ingénierie des connaissances intègre ces réflexions dans des systèmes informatiques – *e.g.* systèmes d'information – qu'il faut modéliser, concevoir, bâtir et faire évoluer. Ces deux points de vue ne sont pas indépendants : le système informatique utilisé par un acteur, le système d'information utilisé par une collectivité modifient le comportement et l'organisation de l'acteur comme de la collectivité. Il nous semble alors nécessaire de penser la technique à travers les usages puisque un système est conçu pour une pratique, et les usages à travers la technique puisque les pratiques n'existent qu'à travers les techniques qui les rendent possibles, (*cf.* 10.2.1.1, n°3 et [Charlet 1998]).

10.2.3. Des systèmes dans une organisation

Les SBC focalisant sur l'expertise de spécialistes ont pu, pendant quelques années, négliger les aspects organisationnels jugés parfois lointains de cette expertise. Cependant, penser un système comme une intervention dans un ensemble d'usages et de pratiques amène l'Ingénierie des connaissances à redonner sa place à l'organisation existant par et autour des acteurs. Ainsi l'organisation, que les systèmes d'information avaient pris en compte, notamment à travers la modélisation des flux d'information commence-t-elle à être modélisée autour de la connaissance et des processus organisationnels de gestion de la connaissance.

En prenant un point de vue organisationnel, on peut alors dire comme le souligne Le Moigne [1995] que « la connaissance implique un sujet connaissant et n'a pas de sens ou valeur en dehors de lui. ». Ainsi, on ne peut modéliser la connaissance sans se préoccuper des invariants sociaux-culturels que transporte le sujet connaissant, du contexte d'utilisation de cette connaissance (*cf.* 10.2.1.2, n°2), en particulier des interactions et coopérations d'acteurs dans lesquels elle s'insère et des contextes organisationnels ainsi créés. Ce qu'affirment ainsi les épistémologies constructivistes et que découvrent de leur côté d'autres approches théoriques comme la cognition située : les connaissances sont avant tout connaissances pour l'action (*cf.* 10.2.1.2, n°1), elles sont situées dans un contexte physique, socialisées dans un réseau d'acteurs, interprétées par les capacités perceptives autant que de raisonnement de ceux-ci et prennent place dans les organisations qu'ils ont construites. *Ainsi, L'ingénierie des connaissances, centrée sur l'objet complexe, situé et socialisé qu'est la connaissance, doit participer à une démarche*

d'explicitation globale des processus de traitement de l'information par l'acteur comme par l'organisation pour produire des méthodes reproductibles de gestion des connaissances (cf. 10.2.1.2, n°3)⁷.

À la suite de ces considérations épistémologiques et des enjeux qui en découlent, nous allons détailler les apports de l'Ingénierie des connaissances en termes de méthodes. De façon évidente et vu les problématiques du domaine, ces apports ressortissent principalement à la modélisation des connaissances.

10.3. Modélisation des connaissances

10.3.1. Nature et contenu des modèles

Les travaux en Ingénierie des connaissances sont donc fondés sur la nécessité de modéliser de manière explicite les connaissances (cf. 10.2). Le génie logiciel est, dans ce cadre, un des domaines privilégiés d'interaction avec l'Ingénierie des connaissances pour proposer des méthodes de modélisation. Il s'agit de construire des modèles adaptés à la nature des connaissances à décrire pour pouvoir ensuite représenter ces connaissances dans des formalismes adéquats. En génie logiciel, les aspects statiques, fonctionnels et dynamiques sont spécifiés séparément à l'aide de primitives adaptées et, ceci, selon la plupart des méthodes orientées objets. Ces différents points de vue coopèrent et interagissent. L'ensemble des modèles et leurs interactions constituent la spécification du système à implémenter. Dans le cadre des SBC, trois types de connaissances font l'objet de modèles distincts avec des primitives de modélisation propres : les connaissances du domaine, les tâches et les méthodes. L'ensemble de ces modèles forme alors ce qu'on appelle communément en Ingénierie des connaissances, le modèle conceptuel. L'analyse du domaine et l'analyse des tâches d'un SBC sont à rapprocher des analyses statiques et fonctionnelles effectuées en génie logiciel. En revanche, l'analyse de la dynamique en génie logiciel consiste à étudier les conséquences d'événements extérieurs sur une

⁷ Notons ici que le contexte rend compte de la variabilité entre les individus, de leur différence de comportement devant telle ou telle situation, alors que l'Ingénierie des connaissances cherche à produire des méthodes rendant compte de comportements invariants. C'est toute la difficulté de l'entreprise : figer un contexte d'interprétation pertinent pour un temps suffisamment long au regard de l'entreprise et du système d'information mais prévoir les évolutions qui ne manqueront pas de se produire.

application alors que pour un SBC, il s'agit de rendre compte du contrôle interne du comportement du système à implémenter. En effet, le problème de l'efficacité d'un SBC ne peut être posé uniquement en termes d'efficacité algorithmique : ce problème va aussi se poser par rapport aux connaissances utilisées ; celles-ci devant alors être décrites de manière déclarative en phase d'analyse.

10.3.1.1. Des connaissances du domaine aux ontologies

Les connaissances du domaine d'un SBC sont les connaissances relatives au domaine de l'application qui sont nécessaires pour que les méthodes de raisonnement puissent s'exécuter. Les travaux qui ont porté sur ces connaissances ces dernières années ont eu un double impact. D'une part, ils ont montré l'intérêt de distinguer les connaissances du domaine selon leur nature et de raisonner sur des modèles de connaissances multiples en exploitant les spécificités de chacun d'eux. D'autre part, ils ont montré l'intérêt de disposer de modèles de connaissances structurés, exprimés à l'aide de langages ayant une sémantique bien définie et exprimés à différents niveaux de granularité.

La notion de modèle du domaine trouve son origine dans les travaux sur le raisonnement à partir de modèles développés dans les années 80 [De Kleer 1987, De Kleer 1989, Genesereth 1982, Reiter 1987]. Ces travaux ont montré qu'il était nécessaire de combiner plusieurs types de connaissances pour augmenter l'efficacité de la résolution. La coopération entre plusieurs modèles du domaine a été étudiée dans le cadre des travaux sur les systèmes experts de seconde génération (SESG) [Steels 1985] dont l'objectif premier était de combiner raisonnement heuristique et raisonnement à partir de modèles (ou raisonnement fondé sur des connaissances de surface versus raisonnement fondé sur des connaissances profondes selon Kayser [1988]).

Petit à petit, l'idée d'accorder de l'importance à la modélisation des connaissances du domaine s'est imposée. L'analyse du comportement de systèmes dans des applications réelles complexes a contribué à renforcer la nécessité d'une telle approche. Les SESG sont des systèmes au sein desquels sont explicitement représentés les différents types de connaissances du domaine de l'application pour permettre un raisonnement efficace et robuste, et ce quelle que soit la difficulté du cas traité⁸. Ces connaissances du domaine

⁸ Les travaux sur les SESG [David 1993] traduisaient ainsi les enseignements des systèmes à base de règles, les SE, en modélisant différents types de connaissances pour, principalement, (i) faciliter l'acquisition et la validation des connaissances, (ii) combiner des raisonnements

recèlent des connaissances spécifiques plus ou moins indépendantes des modèles de raisonnement. Ces connaissances constituent alors l'« ontologie » du domaine.

De manière générale, une ontologie⁹ est l'ensemble des objets reconnus comme existant dans le domaine. Construire une ontologie d'un domaine, c'est donc décider quels sont les objets que l'on retient comme existant – *i. e.* décider quels objets possèdent une consistance ontologique et lesquels n'en ont aucune. Construire une ontologie, c'est aussi décider de la manière d'exister de ces objets. En effet, tous les objets d'un domaine n'existent pas de la même façon. Certains sont des *objets physiques*, d'autres ne sont que des propriétés ou des attributs d'autres objets, ou des idées sans réalités matérielles, etc. Une ontologie comprend donc tout ce à quoi il faut penser dans un domaine, tous les objets de pensée d'un domaine. La façon d'y penser ou d'en parler indique le type ontologique de l'objet. Par exemple, un docteur n'est pas un *objet physique*, c'est le *rôle académique* « docteur » tenu par un *objet physique* appelé « être humain ». Ainsi, il faudra soigneusement décider, d'une part, si l'on retient la notion de « docteur » dans l'ontologie du domaine et, d'autre part, quelle est la manière d'être de la notion de « docteur » pour lui donner le type ontologique qui lui revient [Bachimont 2000, Charlet 1996].

Traditionnellement, la logique étudie le discours en tant qu'il permet de dire le vrai et s'efforce de dégager les formes ou manières de dire qui permettent de déterminer la vérité du discours indépendamment de sa matière, c'est-à-dire ce qui est dit. Mais un discours vrai est un discours qui est conforme à ce qui est. C'est ainsi que la logique entretient une forte proximité avec l'ontologie où les lois permettant de dire le vrai reviennent à des lois caractérisant ce qui est. Si bien qu'élaborer une ontologie peut être compris comme la construction d'une théorie logique du monde. L'Ingénierie des connaissances hérite de cette tradition dans la mesure où elle représente ses connaissances symboliquement et formellement. Par conséquent, toute ontologie selon l'IC sera logique ou réductible à la logique. Donc, en reprenant la structure des langages formels du type de la logique du

différents selon le type de problèmes et (iii) conserver des capacités de raisonnement décentes aux limites des modèles construits.

⁹ Le terme « Ontologie » vient de la philosophie où « il désigne la partie de la métaphysique qui s'intéresse à l'Être en tant qu'Être » (Petit Robert). Mais l'Ontologie est habituellement davantage comprise comme une science des étants que comme une science de l'Être en tant qu'Être, c'est-à-dire qu'elle s'intéresse davantage à ce qui existe (les étants ou existants) qu'aux principes de ce qui existe (Être). Cette science, l'Ontologie, produit des ensembles, les ontologies.

premier ordre, une ontologie sera constituée : (a) de classes d'individus – symboles de variables et de constantes dénotant les individus du domaine –, (b) d'associations – relations n -aires et $n - 2$, entre les individus – et (c) d'attributs – symboles de propriétés (relations unaires ou concepts) des individus et associations. En conclusion, faire une ontologie, c'est décider des individus qui existent, des concepts (attributs) qui les caractérisent et des relations qui les relient. On voit par conséquent que la structure du langage logique influe le type d'ontologie élaborée, ainsi que les objets que l'on peut y rencontrer. C'est pourquoi il faut considérer cette caractérisation proposée ici comme utile en pratique mais non comme un choix théorique dans la mesure où il faudrait discuter plus avant le rapport entre logique et ontologie, logique et langage, forme et objet.

Enfin, rappelons encore qu'il faut distinguer l'ontologie d'un domaine de la base de connaissances d'un domaine. Alors que la première recense ce qui existe et le définit par ses propriétés essentielles, elle ne rapporte pas tout ce qu'on peut déduire ou vérifier ; ce que fait la seconde dans le cadre de ce qu'il est utile de représenter. Une ontologie est la base des descripteurs du domaine et de leur définition, une base de connaissances utilise ces descripteurs pour énoncer tout ce qu'il faut savoir sur le domaine¹⁰.

10.3.1.2. Les modèles de raisonnement

Les modèles de raisonnement décrivent de façon abstraite le processus de résolution à mettre en œuvre dans un SBC en termes de tâches et de méthodes, les tâches étant réalisées par des méthodes. Une tâche est une description de ce qui doit être fait dans l'application en termes de buts et de sous-buts. Elle se définit par des connaissances de sortie obtenues à partir des connaissances d'entrée, et ce en fonction de contraintes et ressources disponibles. Pour décrire une résolution de problèmes, on peut mettre en évidence des tâches de différents niveaux de granularité, les tâches de plus bas niveau poursuivant des sous-buts pour les tâches plus générales. Les méthodes décrivent comment un but peut être atteint en termes d'une série d'opérations et d'un ordre de réalisation. Deux types de méthodes sont généralement distinguées : celles qui consistent à décomposer une tâche en sous-tâches de granularité plus fine et celles qui mettent en œuvre une procédure élémentaire qui atteint directement un but.

La distinction explicite faite entre les concepts de tâches et de méthodes n'a pas toujours été adoptée. Elle a été proposée initialement par L. Steels dans son approche

¹⁰ L'ontologie contient l'essentiel et le définitoire, la base de connaissances le propre et le contingent.

« componentielle » [1990]. Cette distinction, aujourd'hui reconnue par l'ensemble des travaux en Ingénierie des connaissances [Klinker 1991, Puerta 1992, Schreiber 1994, Tu 1995], présente l'avantage de décrire séparément le but visé de la façon de l'atteindre. Elle rend ainsi possible la définition explicite de différentes façons d'atteindre un même but par l'association de plusieurs méthodes à une même tâche.

10.3.1.3. La combinaison réfléchie des modélisations domaine et raisonnement

À la suite des travaux sur les SESG, l'intérêt de construire un modèle conceptuel d'une application en combinant réutilisation de composants de modèles de raisonnement et abstraction de connaissances du domaine a été reconnu. Une analyse des connaissances du domaine devint alors nécessaire pour mettre en correspondance les connaissances du domaine et les rôles à jouer par ces connaissances dans le raisonnement [Aussenac-Gilles 1997]. Les connaissances du domaine sont alors décrites à différents niveaux de généralité. L'ontologie du domaine regroupe les entités du domaine et les relations entre ces entités. Le modèle du domaine décrit les connaissances du domaine proprement dites.

L'importance accordée à la modélisation des connaissances du domaine n'a pas eu toutefois un impact immédiat dans les travaux en Ingénierie des connaissances. L'accent a été d'abord mis sur la construction du modèle de raisonnement d'une application, une étape majeure du processus de modélisation. Les connaissances du domaine à acquérir ont alors été considérées comme étant celles nécessaires au modèle de raisonnement pour s'exécuter. Elles étaient désignées par leur rôle dans le raisonnement, ce qui guidait leur acquisition.

Afin de disposer de descriptions de raisonnement réutilisables, les modèles de raisonnement ne réfèrent aux connaissances du domaine qu'indirectement. Seuls les rôles joués par les connaissances du domaine dans le raisonnement sont spécifiés. Les termes utilisés à cet effet (symptômes, hypothèses, diagnostic, etc.) appartiennent au vocabulaire d'ontologies dites de méthodes ou de modèles (*cf.* CommonKADS [Schreiber 1994]).

Ces modèles, comme tout modèle, sont réducteurs. Ils simplifient la vision que l'on peut avoir du système, un des buts étant justement de réduire la complexité en fournissant une représentation synthétique facile à comprendre et permettant de se focaliser sur les aspects importants dans le développement d'un SBC. Ces séparations dans les descriptions offrent trois points de vue différents qui interagissent pour donner une spécification du système au sein d'un modèle cohérent.

10.3.2. Niveau de description des connaissances

Les modèles de connaissances du domaine et du raisonnement sont représentés au « niveau connaissance », un niveau de description situé au-dessus du niveau symbolique, permettant de décrire des connaissances indépendamment des conditions précises de leur implémentation. Par ailleurs, ce sont des modèles abstraits car ils réfèrent à des concepts ou à des classes d'objets, pas directement aux objets d'un domaine d'application.

En revanche, le niveau de généralité auquel les connaissances sont exprimées est variable. Les modèles peuvent être génériques s'ils correspondent à des abstractions communes à plusieurs types d'applications. Ils peuvent être spécifiques à une application s'ils font référence aux concepts propres à une application donnée. La mise en évidence de parties de modèles communes à plusieurs applications peut, en effet, amener à construire des modèles au sein desquels les termes spécifiques aux applications sont remplacés par des termes les généralisant. Ces modèles, plus facilement réutilisables, sont dits génériques.

La construction de modèles génériques concerne à la fois les modèles de connaissances du domaine et les modèles de raisonnement. La réutilisation et l'adaptation d'ontologies déjà constituées peuvent apporter une aide importante dans l'explicitation et la structuration des concepts du domaine propres à une application lorsque celle-ci et les ontologies disponibles concernent le même domaine [Charlet 1996]. La construction d'un modèle de raisonnement d'une application par réutilisation d'éléments de raisonnement d'une classe de problèmes, ou tâches, est également très recherchée. Le modèle de raisonnement d'une application est alors vu comme une instance de modèles de résolution de problèmes génériques sélectionnés dans une bibliothèque, affinés, adaptés [Fensel 1997] ou encore assemblés [Molina 1996].

Le degré de généralité d'un modèle de raisonnement détermine la façon dont sont décrites les connaissances du domaine manipulées. Une description d'un raisonnement spécifique à une application réfère directement aux concepts de l'application. Dans un modèle générique, les connaissances du domaine requises sont dénotées par leur rôle dans le raisonnement. Le choix de noms, parfois accompagnés d'un commentaire sur leur signification, et dénotant précisément la place de connaissances dans le raisonnement, est alors essentiel. C'est sur la base de ces labels fonctionnels que s'établit la connexion avec les connaissances du domaine d'une application lors d'un développement. Selon les approches, des liens plus forts entre les connaissances du domaine et les connaissances de raisonnement ont pu être recherchés. Des travaux ont ainsi porté sur une meilleure caractérisation sémantique [Causse 1993, Le Roux 1994] et/ou syntaxique [Beys 1996] de ces rôles.

10.3.3. Des langages de modélisation

Ces dernières années, de nombreux langages de description des connaissances ont été proposés pour décrire les modèles construits aux différentes étapes du cycle de développement d'un SBC. On distingue les langages de modélisation, des langages de représentation ou d'opérationnalisation des connaissances.

Les langages de modélisation (*cf.* CML au sein de la méthodologie CommonKADS [Schreiber 1994]) offrent des primitives adéquates pour modéliser semi-formellement et de façon structurée des connaissances. Ces notations, qui peuvent être graphiques ou textuelles, sont particulièrement pertinentes pour construire un modèle conceptuel d'une application au niveau connaissances. Leur principale caractéristique est d'être lisible et compréhensible par les différents intervenants humains impliqués dans la conception d'un SBC. A titre d'exemple, la spécification d'une tâche de diagnostic de voitures en CML [Wielinga 1993b] est la suivante :

```

task car-diagnosis;
  task-definition
    goal:    find one solution;
    input:   initial-complaint: complaint that triggered the diagnostic process;
              user-data: additional information supplied by the user;
    output:  solution; system-generated diagnosis;
    spec:    find an explanation for the initial complaint that is confirmed by the
evidence provided by the user data. If no explanation is found, the task should simply fail.
  Task-body
    type:    composite;
    sub-tasks: generate, test;
    additional roles: hypothesis: a potential solution that needs to be tested;
    control structure: car-diagnosis(c:initial-complaint + D:user-data _
h:solution) =
      repeat generate (c:initial-complaint _h:hypothesis)
      until test(h:hypothesis + D:user-data _true)

```

Les langages de représentation permettent le codage symbolique des connaissances, dans un formalisme caractérisant, par des règles ou des structures syntaxiques, les énoncés licites. L'objectif poursuivi est la réduction des ambiguïtés et des imprécisions inhérentes à la modélisation en privilégiant l'utilisation de langages formels comme FORKADS [Wetter 1990], KARL [Fensel 1991], (ML)² [Van Harmelen 1992]. Ces langages utilisent des formalismes logiques pour représenter chaque type de connaissances défini dans la méthode CommonKADS [Wielinga 1992b].

L'opérationnalisation consiste à définir des structures de données informatiques et/ou utiliser des outils de programmation pour simuler ou implanter tout ou partie d'un SBC. Les langages opérationnels associent aux primitives de modélisation les structures de données et les traitements nécessaires à leur mise en œuvre informatique. MODEL-K [Karbach 1991] et OMOS [Linster 1992], associés à la méthode KADS, ainsi que AIDE [Gréboval 1992], LISA [Delouis 1993], ZOLA [Tchounikine 2000] et le langage opérationnel associé au *shell* TASK en sont des exemples. Ces langages permettent de bénéficier rapidement des résultats d'exécution, ce qui aide à préciser et améliorer la représentation initiale [Fensel 1994]. Le choix d'un langage dépend des objectifs que l'on se donne. Ces derniers pouvant évoluer tout au long du cycle de développement d'une application SBC, l'utilisation de plusieurs types de langages peut être nécessaire, chaque type étant adapté à une étape particulière du développement.

10.3.4. Le processus de modélisation

Plusieurs analyses décrivent la construction du modèle d'une application sous la forme d'un processus qui comporte quatre phases [Aussenac-Gilles 1992, Linster 1992] : (1) le recueil de données brutes, (2) la construction d'un schéma de modèle conceptuel, (3) la définition du modèle conceptuel complet, (4) l'implémentation de ce dernier dans une base de connaissances opérationnelle. L'élaboration du modèle conceptuel, qui décrit l'expertise nécessaire au futur SBC pour résoudre un problème, correspond aux phases 2 et 3 de ce processus. Le schéma du modèle conceptuel est une représentation abstraite et structurée de l'expertise. Il se compose d'un schéma des connaissances du domaine et d'un schéma des connaissances du raisonnement. La phase 2 consiste donc à identifier les concepts abstraits qui définissent et structurent les connaissances de l'expert ou du domaine. La phase 3 consiste à enrichir le schéma de toutes les données du domaine d'application. Il s'agit en quelque sorte d'instancier le schéma sur un domaine d'application.

Aider la construction du modèle d'une application peut être vu de deux façons différentes selon que l'accent est mis sur l'obtention d'un modèle exécutable ou sur la construction du modèle conceptuel. Les travaux du premier type offrent des architectures spécialisées dans la réalisation de certaines tâches et/ou de certaines méthodes. Les connaissances du domaine à acquérir sont définies d'après le rôle qu'elles doivent jouer dans le raisonnement défini au sein de ces architectures (aide à la réalisation de l'étape 3 ci-dessus). Les travaux du deuxième type fournissent plus généralement une aide à la construction du modèle conceptuel (étapes 2 et 3) basée sur l'exploitation de contraintes syntaxiques de langages d'expression des connaissances.

10.3.4.1. Des architectures spécialisées

Les travaux qui relèvent de la première approche considèrent que la méthode mise en œuvre dans une application conditionne les types de connaissances nécessaires et que le rôle des connaissances dans la méthode contraint la façon dont les connaissances doivent être représentées et recueillies. C'est l'identification des connaissances nécessaires à la méthode qui guide l'acquisition. Étant donnée une application SBC, cette approche propose donc un séquençement des activités menant à la construction d'une application SBC : (1) identifier la tâche à résoudre et/ou la méthode de résolution de problèmes à mettre en œuvre, (2) sélectionner l'architecture adaptée à la tâche et à la méthode, (3) acquérir les connaissances du domaine nécessaires à la mise en œuvre de la méthode identifiée.

Reignent dans cette catégorie, l'approche des Tâches Génériques de Chandrasekaran [Chandrasekaran 1986], les travaux sur les méthodes à limitation de rôles de Mc Dermott [Marcus 1989] ainsi que les travaux de Musen et son équipe ayant conduit à la réalisation du système PROTEGE [Musen 1989]. L'approche des Tâches Génériques et celle des méthodes à limitation de rôles mettent l'accent sur le concept de méthode de résolution de problème. L'acquisition des connaissances d'une application doit être précédée de l'identification préalable du modèle de raisonnement sous-jacent à l'application. Les travaux de Musen se distinguent des précédents en mettant l'accent sur le concept d'application. Le but est de disposer d'outils plus proches du concepteur dans la mesure où ils manipulent le vocabulaire du domaine d'une application et non un vocabulaire orienté méthode, décrivant les connaissances au travers de leur rôle dans un raisonnement, indépendamment de tout domaine d'application.

Cette première approche qui a conduit à la construction d'architectures et d'outils spécialisées – *e.g.* SALT [Marcus 1989] – présente l'avantage de guider très efficacement l'étape 3 d'instanciation d'un modèle conceptuel. Les schémas de modèles conceptuels sont prédéfinis et guident l'acquisition des connaissances du domaine. Néanmoins, ces architectures ne sont utilisables que si (a) le concepteur a auparavant effectué une analyse fine de l'application à développer, (b) qu'il a pu identifier précisément le raisonnement sous-jacent et (c) qu'il existe une architecture mettant en œuvre le raisonnement identifié. Ce dernier point est d'autant plus primordial que les architectures proposées sont peu flexibles. Dans l'approche à limitation de rôles, les méthodes sont relativement simples mais ne peuvent être modulées pour une application donnée. L'approche des Tâches Génériques rencontre les mêmes difficultés même si c'est à un degré moindre dans la mesure où le concept de tâches génériques est de granularité plus fine et que les méthodes de raisonnement d'une application sont configurées par composition de tâches génériques.

10.3.4.2. Des architectures génériques

Palliant ce manque de souplesse, d'autres approches proposent des langages de modélisation conceptuelle totalement indépendants de toute méthode, tâche ou domaine. Elles proposent des guides méthodologiques pour mieux maîtriser le développement de gros projets SBC. Ces méthodes ressemblent beaucoup dans leurs principes à celles qui existent en génie logiciel car elles accordent une grande importance à la modélisation. Dans les deux cas, il s'agit de gérer les cycles de développement et de construire un ou des modèle(s) du système à concevoir. La réalisation d'applications est vue comme un processus de transformation de modèles. Ces méthodologies proposent un ensemble de primitives épistémologiques devant servir de base conceptuelle à la modélisation et définissent des structures correspondant à des formes génériques à donner aux expressions qui forment des connaissances. Ces principes sont généraux dans la mesure où ils s'appliquent quels que soient la tâche, le domaine, et le raisonnement mis en œuvre dans l'application à développer.

La méthodologie de développement de SBC la plus connue en Europe est CommonKADS, développée au sein du projet Esprit KADS-II [Wielinga 1993a]. Ce projet avait pour objectif d'intégrer les apports d'une première version de la méthode, KADS-I [Wielinga 1992a], avec ceux d'autres travaux du domaine, en particulier ceux sur les composants de l'expertise de Luc Steels [1990, 1993]. La structuration des connaissances proposée par CommonKADS fournit un cadre intéressant pour guider la représentation du schéma d'un modèle conceptuel (phase 2). Cependant, la méthodologie n'aide pas à déterminer le contenu du schéma de modèle à construire. Ce travail est délicat, largement basé sur l'expérience. Se voulant intégrer et synthétiser les résultats de tout un ensemble d'approches de modélisation, CommonKADS permet la construction du modèle d'expertise aussi bien par abstraction des données que par réutilisation en offrant une bibliothèque de composants de modélisation, de granularité très différente, jugés réutilisables. Le concepteur d'une application SBC dispose ainsi de briques de base pour construire son modèle mais sans réel mode d'emploi.

Si les architectures relevant de la première approche mettent l'accent sur la modélisation des connaissances de raisonnement, la méthode CommonKADS propose une caractérisation des connaissances pouvant servir de base à la modélisation en ne privilégiant a priori aucun type de connaissances. Par ailleurs, la démarche de construction du modèle d'une application n'est pas figée. Deux processus de construction de modèles s'opposent [Van Heijst 1997b] :

- un processus de construction de modèle ascendant (*bottom-up constructive process*) dans lequel une structure est appliquée sur des connaissances déjà élicitées ;

– un processus de construction de modèle descendant (*top-down constructive process*) dans lequel un modèle générique est sélectionné ou construit, puis instancié avec les connaissances propres à l'application étudiée.

Ces deux processus sont vus comme les extrêmes d'un continuum d'approches, allant d'une faible réutilisation de modèles à une forte réutilisation de modèles.

Le premier processus est difficile à mettre en œuvre. Il suppose que le concepteur acquiert l'ensemble de l'expertise sans aucune idée *a priori* sur le modèle de l'application et les connaissances utiles pour construire l'application. La construction du modèle de l'application se fait alors sur la base de l'analyse de cette expertise, par un processus d'abstraction. Le concepteur recherche les traits caractéristiques de l'expertise, dégage les tâches réalisées, les actions exécutées. L'analyse de l'expertise doit permettre d'exprimer l'ensemble des connaissances en des termes plus généraux et de construire un modèle de l'application. L'outil MACAO [Lépine 1996] est une des références de cette approche.

Le deuxième processus est théoriquement plus facile et plus rapide à mettre en œuvre car la construction du schéma du modèle conceptuel est effectuée à partir de modèles (ou de parties de modèles) réutilisés. L'acquisition des connaissances est ensuite aisée ; elle consiste à collecter l'ensemble des connaissances de l'application correspondant aux rôles (on parle de remplissage de rôles), ou aux concepts décrits dans le raisonnement. En revanche, la mise en œuvre de ce processus suppose l'existence de bibliothèques. Il suppose également deux problèmes résolus : celui de l'indexation des composants réutilisables (ou comment retrouver les composants réutilisables) et celui lié à leur configuration (ou comment adapter les composants réutilisés). Apporter des solutions à ces problèmes est aujourd'hui un des enjeux majeurs en modélisation. Ces dernières années, beaucoup d'efforts ont été réalisés pour permettre de construire un modèle d'une application par réutilisation.

10.3.5. La réutilisation

Les travaux sur la réutilisation se sont développés selon deux axes : d'abord en termes de réutilisation des modèles de raisonnement comme nous venons de le voir, puis à la suite de travaux sur les ontologies (*cf.* 10.3.1.1) qui ont relancé la problématique par rapport aux connaissances du domaine.

Pour revenir aux modèles de raisonnement, la réutilisation peut se concevoir de différentes manières. Une première façon consiste à réutiliser des modèles d'expertise

prédéfinis. Cette idée a été initialement proposée dans le cadre du projet KADS-I. Des modèles d'interprétation ont été définis [Breuker 1987]. Ils contiennent des descriptions de raisonnements adaptées à la réalisation de tâches spécifiques et exprimées dans une terminologie indépendante de tout domaine d'application. Le choix d'un modèle d'interprétation est guidé par un arbre de décision basé sur une taxinomie des types de tâches, une version étendue et modifiée de celle de Clancey [1985], description dérivée de celle de Hayes-Roth [1983]. Les critères de choix portent sur la structure du système au centre de l'application analysée, le type de solution attendue au travers du raisonnement à mettre en œuvre, la nature des connaissances du domaine utilisées. Cette approche est séduisante mais l'application d'un modèle d'interprétation à un domaine pose deux types de problèmes. D'une part, une application met souvent en œuvre plusieurs raisonnements auxquels correspondent plusieurs modèles d'interprétation. Il faut alors pouvoir distinguer explicitement ces différents raisonnements. D'autre part, la réutilisation de modèles génériques prédéfinis rend nécessaire la mise en œuvre de processus d'adaptation à une application donnée, un travail difficile et long à réaliser. En effet, si la réutilisation a pour but d'abaisser les coûts de développement, l'expérience montre que la réutilisation de méthodes générales, très largement applicables et donc peu contraignantes par rapport à la structure des connaissances du domaine manipulées, a un coût important, alors que les méthodes, d'application plus limitée, sont plus facilement réutilisables [Talbi 2000]. Cette constatation a ouvert la voie à des approches plus flexibles, au sein desquelles les éléments réutilisés et adaptés sont de granularité plus fine. Il ne s'agit plus de réutiliser des modèles génériques complets de raisonnement mais des éléments de raisonnement. Les critères de sélection des composants génériques restent proches de ceux permettant la sélection des modèles d'interprétation. Ils portent sur les buts (ou sous-but) que l'on cherche à atteindre et les caractéristiques des connaissances du domaine. Les buts sont rapprochés des compétences de méthodes ou mécanismes de résolution. Les caractéristiques des connaissances du domaine sont rapprochées des caractéristiques que doivent posséder les connaissances nécessaires au raisonnement pour s'exécuter [Reynaud 2000].

Par rapport aux ontologies, les réflexions sur la réutilisation ont alimenté de longs débats [Charlet 1996, Guarino 1997, Van Heijst 1997a]. La communauté française est particulièrement active dans ce domaine, en particulier au niveau du groupe « Terminologie et intelligence artificielle » (TIA), au sein duquel ontologies, terminologies, bases de connaissances terminologiques, entre autres « produits » de la recherche, sont caractérisées par rapport aux besoins de la Terminologie et de l'Ingénierie des connaissances. Il en ressort que la réutilisation simple – voire simpliste – des ontologies espérée au début des années 90 a disparue. À l'inverse, une convergence de vue

se fait jour sur quelques points maintenant essentiels que nous reprenons ici du point de vue des méthodes et des ressources terminologiques [Charlet 2000b] :

1. Les textes (documentation technique, ouvrages, textes d'interviews, etc.) étant reconnus comme étant la source essentielle à l'élaboration des ontologies et connaissances du domaine¹¹, la construction d'ontologies doit suivre des méthodologies précises organisées selon plusieurs étapes permettant de passer d'une expression linguistique des connaissances à une ontologie computationnelle utilisée dans un artefact informatique [Bachimont 1996].

2. La réutilisabilité des ontologies est alors limitée puisque la normalisation sémantique, nécessaire au passage du linguistique au formel, ne peut s'effectuer que dans la perspective d'une tâche donnée : une ontologie étant, par essence, toujours spécifique à une tâche. Dans cette hypothèse, la réutilisabilité est recherchée à travers des critères systématiques d'adaptabilité d'ontologies d'une tâche à une autre [Bachimont 2000].

3. L'Ingénierie des connaissances se rapproche de la linguistique et élabore, avec ce domaine, des ressources terminologiques intermédiaires qui doivent permettre de préciser de nouvelles méthodologies et outils d'acquisition d'ontologies et autres connaissances nécessaires aux SBC [Bourigault 2000]. Dans ce cadre, les bases de connaissances terminologiques sont un produit prometteur [Condamines 2000].

Finalement, les recherches liées à la réutilisation ont atteint une certaine maturité en Ingénierie des connaissances. La confrontation avec les recherches en Systèmes d'information devrait permettre de mettre en place des collaborations rapides autour de la problématique des ontologies versus des référentiels, de travailler à moyen terme sur la réutilisation des modèles d'expertise au regard des *design patterns* et, plus tard, d'appréhender les problématiques de prise en compte de la dimension collective et de l'évolution des connaissances dans les organisations (*cf.* 10.5.4). Déjà au niveau des Systèmes d'information, les réflexions de l'Ingénierie des connaissances permettent de proposer des modélisations pertinentes pour l'ingénierie des composants (*cf.* chapitre 5, § 5.3.3).

10.4. Le contexte organisationnel et son impact sur les Systèmes d'information et l'Ingénierie des connaissances

¹¹ Nous réitérons ici la remarque de la note 1 mais nous nous plaçons ici sciemment dans l'hypothèse où le corpus est le lieu privilégié où s'expriment des connaissances stabilisées et consensuelles.

Parallèlement à cette évolution de l'Ingénierie des connaissances, s'est développé, dans les entreprises, un mouvement de complexification des connaissances et de leur traitement. La façon de voir la stratégie de l'entreprise en a ainsi été modifiée.

10.4.1. Les organisations prennent conscience de l'importance stratégique des connaissances

On peut considérer que, depuis très longtemps, toute l'activité des entreprises consiste à fédérer des connaissances et des savoir-faire d'acteurs en vue d'obtenir la production collective de biens ou de services. De sorte que l'on peut dire que, dans les organisations, on traite de la connaissance, des savoir-faire et des apprentissages depuis toujours. La question de l'acquisition, de l'utilisation, de l'agencement des connaissances pour pouvoir agir est même posée explicitement, puisqu'autrefois le compagnonnage et à présent le management – omniprésent à tous les échelons des organisations – est un véritable exercice de « faire-faire », obligeant le manager à résoudre des problèmes à partir des connaissances et des savoir-faire de ses collaborateurs, s'interrogeant sur les attentes de ses clients, fournisseurs, partenaires... depuis les méthodes de la gestion de production qui traitent des problèmes d'apprentissage, d'évolution de savoir-faire, de partage collectif des savoirs, jusqu'au *marketing* qui utilise les « représentations » du produit ou des concepts publicitaires et qui évalue les « valeurs » des clients pour proposer de nouveaux produits.

La nouveauté est la prise de conscience que l'entreprise possède des connaissances et des savoir-faire qui ne se réduisent pas aux compétences des individus et que ceci fait partie de ses premières richesses. Ces connaissances se transmettent certes d'individu à individu mais aussi de collectif à individu et de collectif à collectif. Le concept d'apprentissage est ainsi renouvelé et un concept comme celui de l'apprentissage organisationnel tente de rendre compte de ces phénomènes. Ces concepts deviennent centraux dans la préoccupation qu'ont les entreprises de savoir évoluer très vite dans un environnement changeant sous la pression économique. Certains auteurs voient les organisations elles-mêmes comme un système d'interprétation collectif et ils en caractérisent le fonctionnement avec des concepts liés à la cognition : c'est le cas par exemple d'Argyris [1995] qui propose le concept d'organisation apprenante. Celle-ci est définie comme sachant limiter ses routines défensives et capable de ce fait d'un apprentissage organisationnel en « double boucle » qui modifie non seulement le comportement organisationnel ayant eu des effets néfastes mais aussi plus profondément toutes les conceptions qui avaient produit ce comportement.

Il se précise une approche « cognitive » du management et de la stratégie d'entreprise. Les dirigeants et cadres sont de plus en plus invités à avoir une attitude critique, donc consciente et explicite de leurs modèles interprétatifs de l'environnement et de la réalité intérieure de l'entreprise. Cette démarche vise à ce qu'ils puissent mieux échanger et construire collectivement ces modèles interprétatifs, à savoir les ajuster les uns avec les autres, à pouvoir et savoir les modifier et les faire évoluer. L'objectif étant toujours de mieux adapter leur comportement et celui de l'entreprise à une complexité croissante et à une concurrence accentuée. Cette perception ou connaissance de l'environnement, interne comme externe, est alors au centre de la dynamique organisationnelle [Nonaka 1997].

10.4.2. Les connaissances dans les organisations

Les connaissances évoluent dans la pratique d'un même acteur comme dans celle de l'organisation. On peut considérer qu'elles suivent un développement en plusieurs étapes, qui correspond à différentes natures de connaissances organisées dans un continuum [Teulier-Bourgine 1997]. D'abord encapsulées dans un savoir-faire humain, elles peuvent ensuite être explicitées par l'acteur. À cette étape essentielle, elles peuvent, à travers l'action collective, être échangées, critiquées et partagées. Aussi ne faut-il pas s'étonner que, dans les entreprises, on se serve facilement de la distinction connaissances tacites versus connaissances explicites. Après avoir été explicitée, partagée la connaissance peut être reformulée et se figer, s'institutionnaliser en informations, normes, standards. Par ce processus, elle « s'impose » à tous et peut être utilisée avec peu d'efforts cognitifs des acteurs. Puis, les standards et normes deviennent moins adaptés et il faut un nouvel effort cognitif pour les interpréter, les appliquer et les faire évoluer. On peut considérer que, d'une certaine façon, l'acquisition des connaissances s'est intéressée dans ses débuts au savoir-faire encapsulé chez l'acteur (en l'occurrence l'expert ou encore la première étape du processus que nous venons de décrire), et que les systèmes d'information traitaient traditionnellement des informations bien établies et codifiées, normes et standards (c'est-à-dire la dernière étape du processus d'institutionnalisation).

Les connaissances qui intéressent les acteurs dans l'entreprise, les gestionnaires, les informaticiens et les ingénieurs de la connaissance sont des connaissances pour l'action¹².

¹² Nous n'opposons pas connaissances théoriques et connaissances pour l'action puisque la connaissance est indubitablement liée à l'action (cf. 10.2.1.2 n°1). Le sens de cette affirmation est à replacer dans le contexte organisationnel de l'entreprise où ce sont les

C'est uniquement ce rapport des connaissances avec l'action qui justifie *in fine* les pratiques des uns et des autres. Il s'agit d'un savoir comme intelligence pratique du monde, celui qui est orienté vers et immergé dans l'action et qui n'est pas dépouillé de « la gangue de la contingence » qu'évoquent Delbos et Jorion [1984] ; ce qui le rend sensiblement différent du savoir discursif qui vise à une validité omni-contextes. Il est donc multiforme, difficile à capter, à organiser et à assister par des artefacts. L'objectif des gestionnaires, en particulier quand ils s'intéressent aux connaissances, est de modifier le comportement des acteurs pour faire évoluer le comportement de l'organisation. On comprend alors l'intérêt de la distinction introduite par Argyris [Argyris 1995] entre le savoir « applicable » ou savoir pratique et le savoir « actionnable » : ce dernier comprend le savoir pratique plus ses conditions d'application et le savoir qui a servi à le créer ; ce qui permet d'introduire les conditions de sa remise en cause à travers un processus d'apprentissage en double boucle. L'objectif des informaticiens et des ingénieurs de la connaissance pour leur part est d'agir en concevant et en développant des artefacts qui à leur tour devront s'intégrer dans l'activité de leurs utilisateurs. Leur rapport au lien connaissances-action se situe donc à deux niveaux : (1) ils doivent « lire » un processus organisationnel pour en extraire informations et connaissances pertinentes à modéliser alors que les connaissances pour l'action sont difficiles à isoler de l'action parce qu'elles y sont immergées ; et (2) ils doivent concevoir un outil qui va transformer la tâche des utilisateurs et infléchir les processus organisationnels alors que vouloir infléchir l'action, les comportements, les pratiques des acteurs, en particulier à travers leurs connaissances, est particulièrement complexe (cela amène à intervenir dans un *contexte situé* – *i. e.* un environnement physique, cognitif et social qui structure et organise lui aussi les échanges et l'action).

Une des difficultés les plus notables est d'appréhender l'action et la cognition collective [Teulier 2000], vitales pour les organisations : c'est dans ces processus collectifs que les connaissances se construisent [Hatchuel 1996] et trouvent leur dynamique au sein du tissu organisationnel et de l'activité de production.

10.4.3. La place des connaissances dans cette évolution vers la complexification des systèmes d'information

connaissances pour lesquelles une action est clairement identifiée qui intéressent *a priori* les gestionnaires.

Dans le contexte économique déjà évoqué, les aspects organisationnels de l'entreprise se modifient en profondeur et les exigences vis-à-vis des systèmes d'information se modifient également : on leur demande de plus en plus dans cette tension vers une adaptation à caractère vital. Il s'agit bien du lien, des interactions, des dépendances mutuelles entre la stratégie de l'entreprise, sa structure et la technologie de l'information, ce que Tardieu et Guthmann [1991] ont appelé « le triangle stratégique », reprenant en cela l'expression d'un rapport de la *Sloan School* du MIT.

Cette exigence croissante vis-à-vis des systèmes d'information est observable concrètement dans les organisations. Les mêmes données doivent être traitables de multiples manières (le bon de commande prépare la facture qui se répercute sur la gestion des stocks), saisies de différents postes de l'entreprise (le commercial passe ses commandes de son PC portable), centralisées, décentralisées, utilisées modifiées par de multiples groupes, réseaux, etc. le tout en temps réel¹³. Ces systèmes d'information sont donc de plus en plus complexes ; ils doivent par ailleurs être de plus en plus accessibles aux non-informaticiens. Outre toutes ces caractéristiques, on demande maintenant aux nouvelles générations de systèmes d'information d'opérer des traitements à valeur ajoutée par rapport aux versions précédentes : donc de plus en plus de faire émerger du sens, c'est-à-dire d'interpréter les données et d'opérer des rapprochements entre elles à partir de grands systèmes de gestion de bases de données (SGBD) servant de multiples applications. C'est dans ce contexte qu'il faut apprécier la place des connaissances dans les systèmes d'information de demain.

Reprenons un point de vue « cognitif » et souvenons-nous que l'essentiel du traitement de l'information fait par l'être humain étant un traitement qualitatif, (environ 80 à 90%), on a commencé, en informatisant, à mettre sur machine le quantitatif, c'est-à-dire ce qui était facilement identifiable, *stockable*, traitable. Cette étape est en quelque sorte acquise. On approche maintenant du traitement d'autres objets, supports de la cognition comme l'image, le texte, les connaissances parce qu'il s'agit de plus en plus « d'interpréter » les données et les résultats des traitements, tâche qui était réservée jusqu'ici à l'utilisateur. En conséquence, on demande des traitements de plus en plus sophistiqués, on atteint ce qui tient du traitement qualitatif chez l'être humain [Pomerol 1997]. Or ceci se modélise différemment et c'est un autre métier de le conceptualiser,

¹³ Ces traitements se mettent en place dans les entreprises, au travers des progiciels de gestion intégrés (PGI) dont la paramétrisation est difficile, au point même que ce sont souvent les entreprises qui doivent s'adapter et non l'inverse.

même s'il ne s'agit « que » de fournir des outils de navigation, d'accompagner l'utilisateur dans ce type de traitement qualitatif.

Cette tendance se traduit concrètement par le fait que les utilisateurs exigent des outils d'utilisation intuitive, des interfaces conviviales. Ce qui est visé ici, c'est le gain de temps, et pour cela, un confort d'utilisation maximum et un coût cognitif minimum d'accès. L'information n'est plus rare : elle devient surabondante et très accessible ; le problème est de la trouver, de la sélectionner, de l'interpréter. Les organisations en accumulent de plus en plus ; elles veulent faire émerger du sens de ces grandes masses d'informations, exploiter celles qui leur sont propres ou bien utiliser celles qui, rapprochées de leurs savoir-faire spécifiques peuvent leur donner des atouts concurrentiels. Cette demande s'est déjà concrétisée par une évolution des systèmes interactifs d'aide à la décision (SIAD) [Lévine 1989, Zaraté 2000] et des *Executive Information System* (EIS) [Olson 1992, Turban 1993] ou, plus récemment, des bases de données exploitées et fédérées sous la forme de *data warehouses*. Cette disponibilité accrue de l'information, des technologies et la réactivité supplémentaire demandée à l'entreprise, tendent à accroître aussi le travail collectif. Les systèmes d'information doivent donc, non seulement, accélérer les échanges, mais aussi favoriser le travail de groupe. Or la conceptualisation du travail collectif est en quelque sorte en retard sur les moyens techniques, et il reste une demande forte d'application d'aides à la conception en matière de collecticiels [Lewkowicz 2000], d'outils pour travailler avec le Web. Les travaux collectifs ont surtout été vus à travers l'échange et le partage de documents. Les modèles conceptuels des processus collectifs restent, pour une large part, à construire.

10.4.4. Les systèmes à base de connaissances dans les organisations

Que nous apprend l'expérience récente mais néanmoins conséquente de l'introduction des SE et SBC dans les entreprises jusqu'à présent ? Cette expérience mériterait qu'on s'y intéresse de façon plus approfondie puisque ce sont les expériences concrètes de modélisation et d'informatisation des connaissances qui ont été les plus développées dans les organisations. Nous pouvons tirer quelques leçons de ces décennies en faisant d'abord le constat, assez unanimement partagé, que l'une des limites organisationnelles les plus criantes des SBC est d'avoir été appliqués à des « poches » de savoir facilement identifiables mais très localisées.

Dans ce contexte d'évolution économique et d'évolution organisationnelle, les SE et plus généralement les SBC ont permis des avancées notables concernant la nature, la modélisation, la représentation et le traitement des connaissances. Tout d'abord une

focalisation sur la notion d'expertise dans les entreprises a suscité une réflexion assez généralisée sur ce thème. Au-delà des travaux de psychologie cognitive sur l'intelligence ou d'approches de l'intelligence artificielle sur des expertises très locales et très isolées comme celle du joueur d'échec, il fallait une approche fine de l'expertise à l'œuvre dans les entreprises avec toutes ses modalités différentes d'exercice, avec ses liens de différentes natures avec les autres savoirs et connaissances de l'entreprise ; ceci n'aurait pu être fait sans ces centaines d'expériences de réalisation de SBC. Elles n'ont d'ailleurs été possibles que parce qu'elles correspondaient (au-delà de l'effet de mode qui ne suffit pas à lui seul à créer un tel mouvement) à un réel besoin d'approche de cette expertise et d'automatisation de traitements qualitatifs de l'information. Dans la même démarche, outre la focalisation sur l'expertise, toute une approche modélisatrice des savoir-faire et des connaissances dans les organisations accumulait les premières explorations concrètes sur les connaissances considérées au-delà de la compétence des individus. En effet, parallèlement à la réflexion des gestionnaires sur la connaissance qui aboutissait à distinguer les connaissances explicites, implicites, leur transmission etc., parallèlement à celle des penseurs comme Morin [1986] qui écrivait « la connaissance de la connaissance », le courant qui construisait des SBC aboutissait à mettre sur pied et à proposer – de façon balbutiante au départ – une Ingénierie de la Connaissance, c'est-à-dire une approche de la connaissance (et de ses processus) comme objet identifiable, objet d'ingénierie c'est-à-dire de méthodes, de modèles compréhensibles, échangeables et reproductibles. Le concept de connaissance, aux mains des philosophes depuis des siècles, des pédagogues depuis quelques décennies, arrivait chez les sociologues, chez les ingénieurs et chez les gestionnaires ; les ingénieurs ne s'y intéressant qu'en vue de leur automatisation, introduisant une étape draconienne d'explicitation et de formalisation pour le passage en machine. Celui-ci pèse lourd en termes d'efforts et de temps nécessaires mais son caractère fortement contraignant a aussi permis des avancées importantes dans cette approche de la connaissance.

Que demande-t-on actuellement aux SBC ? D'être totalement intégrés aux systèmes d'information, échangeant avec eux les données, ayant des interfaces communes, bref d'être totalement transparents pour l'utilisateur ; plus, de rester limités à des modules circonscrits, intégrés dans des systèmes d'information, d'être vite développés, faciles à maintenir... En effet les SBC, intégrant des applications plus larges et entrant dans la pratique des utilisateurs, ont « naturellement » rejoint d'autres parties du système d'information. En même temps qu'il prend connaissance des résultats du SBC, l'utilisateur veut pouvoir visionner toutes les données qui ont permis à celui-ci d'élaborer des solutions, un conseil. Il souhaite aussi naviguer « intelligemment » dans ces données, les présenter sous des formes différentes, les rapprocher d'autres données à

choisir parmi plusieurs sources. Bien entendu, il n'est pas question de saisir toutes ces données, le SBC doit être relié à toutes les bases de données de l'entreprise.

10.5. Articuler SBCs et systèmes d'information

10.5.1. L'utilisateur au centre de la conception

L'évolution des enjeux dans les organisations que nous venons de décrire tend à mettre l'utilisateur au centre de la conception. Plusieurs spécialités de l'informatique ont évolué de façon apparemment disjointe vers cette situation. Des travaux comme ceux des interfaces homme-machine (IHM) ou comme ceux du courant « conception centrée utilisateur » [Norman 1986] ont été des étapes importantes. Actuellement dans le domaine de la conception des systèmes d'information, l'Ingénierie des besoins pose explicitement la question des besoins de l'utilisateur et propose comme thème de recherche la construction d'une ingénierie permettant de formaliser cette prise en compte [Grosz 2000]. On peut considérer aussi que la prise en compte des besoins de l'utilisateur est sous-jacente à d'autres courants. La conception à l'aide de patrons, par exemple, à travers la volonté de rendre les applications réactives, cherche à prendre en compte l'ensemble des problèmes qui se manifesteront quand l'application sera en usage : il s'agit, à travers les patrons, de collectionner les symptômes d'inadéquation potentiels d'une application pour pouvoir y répondre à l'avance. De même, les méthodes utilisant les cas d'utilisation (*use cases*) tentent par une collection de situations représentatives d'approcher ce que souhaitera l'utilisateur final et surtout de prévoir l'ensemble des situations dans lesquelles il se trouvera alors que lui-même peine à les imaginer à l'avance. L'Ingénierie des connaissances s'est focalisée, pour sa part, sur la résolution de problème, en modélisant celle-ci de façon suffisamment générique pour englober tous les cas de figure pouvant advenir. Aucune de ces approches n'est, à elle seule totalement satisfaisante. Toutes, par contre, vont dans la même direction et ont à interagir.

Mettre l'utilisateur au centre de la conception revient à l'accompagner dans ses processus de réflexion et de manipulation des données. En effet, les systèmes d'information pénètrent de plus en plus ce qui était du ressort de l'utilisateur dans le partage des tâches entre l'homme et la machine, et cette amplification de leurs rôles nécessite des modélisations du métier. Ce qui est difficile à cadrer, complexe et coûteux et aboutit à une évolution profonde de la conception de systèmes d'information. Pour accompagner les métiers de l'organisation, les concepteurs informatiques doivent non seulement concevoir et réaliser des outils adaptés, mais ils doivent aussi accompagner les utilisateurs dans l'évolution de leur métier, dans le changement organisationnel [Prekas

1999]. Il s'agit là, de les aider, à des degrés divers, à réfléchir à leurs propres concepts. C'est un travail de construction des connaissances bien connu de l'Ingénierie des connaissances (*cf.* 10.3.4).

Il faut situer cette prise en compte croissante des besoins des utilisateurs dans un contexte plus global. La tendance générale d'utilisation et de développement d'outils génériques, comme les tableurs, traitements de textes, *Web* ou navigation hypermédia est la tendance de fond. De ce point de vue, la réponse au problème de la prise en compte des besoins utilisateurs, s'organise en trois niveaux : (a) les systèmes d'assistance ou conseillers conçus et développés spécifiquement pour un ensemble d'utilisateurs, (b) les systèmes conçus comme des aides spécialisées à partir d'outils génériques et (c) les systèmes nécessitant surtout un paramétrage d'outils génériques. À chacun de ces niveaux, les concepteurs ont, à des degrés divers, à prendre en compte les besoins de l'utilisateur ; que ce soit en termes d'échanges, de fiabilité, de sécurité et de convivialité ou par rapport à des contraintes organisationnelles.

Les acquis de l'Ingénierie des connaissances pour cette prise en compte des besoins utilisateurs sont (a) la connaissance et l'expérience de modélisation des processus cognitifs, (b) l'expérience de modélisation de la tâche, (c) le fait de savoir que recueillir la connaissance est en fait en grande partie la reconstruire, (d) le fait de savoir que l'outil futur modifie la tâche et son environnement.

Pour l'Ingénierie des connaissances, s'orienter vers une plus grande prise en compte de l'utilisateur revient à élargir sa vision et à prendre l'utilisateur non seulement dans sa dimension d'être raisonnable et résolvant des problèmes, mais aussi d'être social engagé dans des échanges et des interactions. Ce qui oblige à modéliser l'utilisateur - et pas seulement l'expert - comme acteur organisationnel engagé dans sa tâche et ses échanges. L'utilisateur est alors un acteur participant à des travaux collectifs et est immergé dans une organisation imposant ses contraintes, ses procédures et ses sources d'information.

La prise en compte des besoins de l'utilisateur et de la capacité de conception de celui-ci dans l'évolution de son métier est un challenge aussi bien pour l'Ingénierie des connaissances que pour les Systèmes d'information. Chacune des spécialités ayant des acquis dans ce domaine : que ce soit, par exemple, la prise en compte du changement organisationnel pour l'Ingénierie des besoins ou l'élaboration coopérative des concepts du domaine pour l'Ingénierie des connaissances.

10.5.2 Vers des systèmes d'information coopératifs

Les systèmes informatiques peuvent être considérés comme coopératifs, parce qu'ils coopèrent avec leur utilisateur [Rosenthal-Sabroux 1996], problématique abordée précédemment, ou parce qu'ils coopèrent entre eux et sont largement communicants et interopérants – c'est l'objet de l'activité d'intégration de systèmes [Meynadier 1998] –, ou encore parce qu'ils assistent la coopération d'agents humains et produisent ainsi un système coopératif impliquant des hommes et des machines [Zacklad 1994]. Ces trois aspects des propriétés coopératives des systèmes d'information sont souvent présents simultanément et leurs effets se renforcent. C'est parce qu'on peut échanger des données qu'on peut concevoir des artefacts qui coopèrent avec l'utilisateur et c'est parce qu'on met l'utilisateur au centre de la conception qu'on est conduit à l'assister dans ce qui est essentiel pour lui : ses travaux collaboratifs.

L'intégration de systèmes existants, non conçus initialement en vue de fonctionner ensemble, a pour objectif de les faire coopérer malgré leur hétérogénéité et d'obtenir des propriétés et une efficacité nouvelles. Comme le souligne Meynadier [1998], dans les années 80 « *alors que la majorité de l'informatique des organisations restait dans un mode transactionnel en systèmes centralisés auxquels se sont peu à peu connectés les ordinateurs personnels, les grands systèmes faisant l'objet de projets d'intégration de systèmes préfiguraient les architectures réparties et client-serveur des années 90* ». Ces années 90 ont vu la généralisation des systèmes distribués selon le mode client-serveur, généralisant aussi de fait l'intégration de systèmes. Outre les aspects techniques de l'interfaçage des différents systèmes, l'une de difficultés de l'intégration est de rendre transparent à l'utilisateur l'hétérogénéité et la complexité interne des systèmes, donc de rendre cohérentes les significations des données de différentes bases de données et leur utilisation par différentes applications. Lorsque des SBC sont dans une chaîne d'intégration, ils ont donné lieu à une modélisation du domaine et à un dialogue avec la maîtrise d'ouvrage. De ce fait, ils participent souvent à la conception d'une interface unique et à l'élaboration d'un schéma global de la chaîne d'intégration utilisant les concepts, contraintes et objectifs du métier impliqué.

Au-delà de ces échanges et de cette communication généralisés, la coopération entre les agents tend aussi à se généraliser, poussée par les technologies et tirée par les besoins d'adaptation et de réactivité des entreprises. Le fait que chaque poste de travail soit connecté à un ensemble de systèmes d'assistance et d'information, dans et hors de l'organisation, rend nécessaire une nouvelle approche du système d'information centrée sur des situations d'utilisation et des savoir-faire partagés d'utilisateurs. Il s'agit de caractériser conceptuellement les types de connaissances et de processus cognitifs qui sont en jeu dans le travail collectif, d'explicitier et de caractériser les besoins des utilisateurs non seulement dans leur fonctionnement cognitif individuel mais aussi dans leurs travaux

coopératifs. L'imbrication des tâches des uns et des autres et le nombre de tâches réalisées en commun augmentant, l'enjeu devient alors de savoir organiser le travail en commun, de l'assister et le faciliter. Pour cela, il faut concevoir des systèmes informatiques spécialement destinés à cette tâche. Assister la coopération d'agents humains oriente toute la conception de ces systèmes par la prise en compte des besoins de collectifs de travail identifiés.

Sur le plan conceptuel, on peut considérer que les assistances logicielles au travail collaboratif sont basées sur la coopération au sens de Zacchary et Robertson [1990] : « La coopération est une forme particulière d'interaction dans laquelle les agents coopérants partagent des buts et agissent de concert avec eux sur une certaine période de temps ». La coopération est donc vue comme une résolution collective de problème. L'origine des travaux conceptuels sur la coopération [Falzon 1991] se fonde sur la modélisation de la résolution de problème, telle que proposée par Newell et Simon [1972]. Il s'agit donc d'une origine commune forte avec l'Ingénierie des connaissances.

De leur côté et depuis les années 80, les travaux des communautés CSCW¹⁴ ont exploré le support de travaux de groupe et particulièrement d'échanges en s'appuyant, surtout les premières années, sur la technologie des réseaux et sur la sociologie pour comprendre les caractéristiques du travail collectif. Ils ont montré assez vite [Bannon 1993] que l'introduction efficace de ces outils ne doit pas reposer uniquement sur la technologie mais aussi sur une bonne compréhension des fonctionnements collectifs et organisationnels. Ces travaux ont produit des logiciels de type *groupware* qui servent de systèmes de médiation entre agents pour les décisions ou l'élaboration de solutions collectives et des logiciels de type *workflow* reposant sur des protocoles de circulation des informations dans l'organisation. Par ailleurs, les gestionnaires et informaticiens de gestion, produisaient à travers la conception des *Group Decision Support System* [Olson 1992, Turban 1993] des avancées sur l'assistance à la prise de décision collective.

Au départ les travaux de la communauté CSCW ont surtout insisté sur les modes d'échanges et de dialogue entre les acteurs. L'étude de situations de travail réelles, complexes et difficile à « assister » ont amené le CSCW à collaborer avec l'ergonomie cognitive, la psychologie cognitive ou de l'anthropologie et à étudier finement des processus comme les dialogues ou la répartition des rôles. Des sciences comme l'ergonomie cognitive ont ainsi apporté une modélisation du dialogue, des stratégies développées par les acteurs dans les échanges coopératifs comme, par exemple, les

¹⁴ *Computer Supported Cooperative Work* ou travail coopératif assisté par ordinateur (TCAO).

modèles d'usage du langage [Clark 1996, Clark 1993] ou les phénomènes de vigilance mutuelle [Rognin 1998]. Ces travaux intéressent l'Ingénierie des connaissances dans une acception large, ils éclairent des processus cognitifs interférant avec des tâches de résolution de problème, et se révèlent particulièrement importants pour concevoir des environnements globaux de travail collectif comme les salles de marché ou d'aiguilleurs du ciel ou de centre d'urgences [Salembier 1994].

Mais de façon plus globale, avec la sophistication grandissante des applications, les travaux sur la coopération rencontrent un besoin croissant de modélisation. Et les méthodes et outils de l'Ingénierie des connaissances s'avèrent pertinents pour modéliser ces situations de résolution collective de problème [Zacklad 1994]. Ainsi des travaux proposant d'agir notamment sur la mise en valeur des savoirs échangés [Lewkowicz 1999] et proposant de garder la trace du dialogue argumentatif, enrichissent des méthodes de conception venues du *design rationale* par l'utilisation de KADS. Un des nouveaux enjeux des travaux sur la coopération étant la modélisation des processus de conception collective [Darses 1996, Hatchuel 1994]. Une ingénierie des processus de construction collective des connaissances est de fait en train de se construire.

En résumé, le développement des moyens technologiques permet d'aborder la conception d'assistances prenant en compte la complexité de l'acteur organisationnel dans tous les aspects de sa cognition située, engagée dans l'action collective. Cette complexité nécessite cependant un travail pluridisciplinaire ainsi qu'un attirail conceptuel et méthodologique fondé sur différentes disciplines dans une démarche d'ingénierie des connaissances. C'est ainsi qu'on peut éviter le risque d'un point de vue technologique, finalement réducteur.

10.5.3. Vers une ingénierie des composants

Concevoir un système d'information ou un SBC est complexe. De multiples décisions sont prises tout au long du cycle de développement pour aboutir à un système final dont le modèle associé est, dans la pratique, impossible à justifier. C'est pour répondre à ce problème que l'ingénierie des composants est, à l'heure actuelle, une activité en plein essor. L'emploi de *cas de figures remarquables* peut permettre plus de facilité de compréhension et une qualité du système final plus grande sans avoir à donner plus de justification. La conception, vue ainsi comme un assemblage de composants prédéfinis, s'en trouve facilitée.

Cette aide apportée aux concepteurs d'applications peut prendre différentes formes. Dans le domaine des systèmes d'information, les développements actuels voulant favoriser la réutilisation se tournent vers l'usage de patrons. Qu'il s'agisse de descriptions abstraites de regroupements prédéfinis et remarquables de classes collaborant à la réalisation d'un ou de plusieurs services particuliers, ou de fragments de modèles de processus, les patrons correspondent à de nouvelles formes de composants permettant la réutilisation de décisions de conception. Par ailleurs, le concept de *frameworks* qui définit un ensemble de classes (au sens de la programmation objet) collaborant pour réaliser une tâche particulière, a été créé pour permettre de modéliser l'architecture du système à construire. Les patrons proposent une vision globale de l'application à construire en définissant les collaborations entre les classes et les objets ainsi que les flux de contrôle (cf. chapitre 5). Ces travaux constituent, à l'heure actuelle, un véritable enjeu d'une démarche de conception objet et sont également exploités dans les phases d'Ingénierie des besoins. Mettant en œuvre des mécanismes éprouvés, leur utilisation doit contribuer à augmenter la qualité de la conception des modèles objets et la qualité des produits ; cette dernière étant largement dépendante de celle du processus qui en assure l'ingénierie.

Ces approches rencontrent l'Ingénierie des connaissances à plusieurs niveaux :

– Quand il s'agit d'identifier les composants réutilisables et de les sélectionner, il est nécessaire de prendre en compte le contexte d'utilisation de ces composants, que ce soit au niveau de leur définition comme au niveau de l'application à construire. L'*approche par analyse du domaine* (cf. chapitre 5) considère qu'il faut mettre en œuvre un processus de modélisation et de structuration de ce domaine. On retrouve alors les problématiques de réutilisation des SBC (cf. 10.3.5). Les éléments du domaine sont décrits « indépendamment » de la tâche à effectuer, elle-même décrite à travers d'éléments de raisonnements abstraits, les MRP. Les difficultés méthodologiques apparaissent évidemment dans la résolution de la tension entre une hypothèse d'interaction limitée telle que développée dans le cadre du projet KADS [Breuker 1987]¹⁵ et une réutilisation obtenue à travers des critères systématiques d'adaptabilité d'ontologies d'une tâche à une autre : la première hypothèse, séduisante, est irréaliste telle quelle, alors que l'adaptabilité d'ontologies est en tout état de cause un problème difficile et pas encore réellement pris en charge dans des outils mettant en œuvre les méthodologies développées (cf. 10.3.5.).

¹⁵ Depuis, cette hypothèse est sujette à débat et remise en cause par ceux-là même qui l'avaient posée [Guarino 1997, Van Heijst 1997a, Van Heijst 1997b].

– La modélisation objet a fédéré ses modèles dans un unique langage de modélisation, UML (*Unified Modelling Language*) qui permet de représenter les artefacts logiciels. Ce langage respecte un métamodèle, lui-même décrit en UML. La nécessité de prendre en compte d'autres modélisations comme la modélisation des organisations a amené la définition d'un niveau de modélisation supérieur, le MOF (*Meta-Object Facility*). UML y est un des métamodèles supportés [Bézivin 2000]. L'identification des différents modèles, de leurs relations et de leurs transformations pour aboutir à la modélisation d'un artefact logiciel sont des enjeux importants pour construire un cadre méthodologique cohérent. Des travaux récents comme ceux de J. Bézivin [1998, 2000] envisagent ces problèmes du point de vue de l'Ingénierie des connaissances et des ontologies où les clés de la réussite sont la définition des bons niveaux d'abstraction et le partage de catégories conceptuelles.

10.5.4 Vers des systèmes de gestion des connaissances

Problématique émergente, la gestion des connaissances est un questionnement incontournable pour l'Ingénierie des connaissances comme pour les Systèmes d'information. Il n'est pas de notre propos ici de définir ou redéfinir ce qu'est la gestion des connaissances. On se reportera, pour le fond, au livre de J.-L. Ermine [1996] et pour un descriptif de différentes approches, au volume *Capitalisation des connaissances* de ce même traité [Zacklad 2000]. Par contre, nous voudrions donner quelques clés pour décrypter les rapports qu'entretiennent et doivent entretenir l'Ingénierie des connaissances et la gestion des connaissances :

1. La demande d'intégration des SBC au sein du système d'information de l'entreprise (cf. 10.4.4) est une tendance de fond inéluctable, au point que l'on peut affirmer aujourd'hui qu'il ne peut pas exister dans l'entreprise un SBC isolé.

2. Cette intégration n'est pas étrangère au concept de système de gestion des connaissances : un système de gestion des connaissances doit appréhender la connaissance dans toutes ses dimensions en permettant les usages et les interprétations nécessaires et, d'un autre côté, être un système d'information informatisé, robuste, en relation avec les autres parties du système d'information de l'organisation, prenant en compte la dimension collective et évolutive de la connaissance [Ermine 1996].

3. Par rapport aux caractéristiques de l'Ingénierie des connaissances développées ici et aux définitions de la connaissance et des SBC qui en découlent (cf. 10.2), l'Ingénierie des connaissances est un des principaux lieux de réflexions et de développements méthodologiques pour réifier le concept de gestion des connaissances. En effet, l'objet

central est dans les deux cas la connaissance avec les difficultés qu'il y a à la définir mais avec les fondements et avancées théoriques maintenant attestés qui permettent quelques caractérisations et réalisations (*cf.* 10.2 et [Ermine 1996]). Ces apports conceptuels et modélisateurs intéressent aussi bien les concepteurs de systèmes d'information que les gestionnaires. Mais, il ne s'agit pas de réduire l'Ingénierie des connaissances à la gestion des connaissances ou inversement : l'Ingénierie des connaissances a d'autres sujets de préoccupation que la gestion des connaissances en entreprise, la gestion des connaissances ayant, de son côté, des problématiques spécifiques.

Ceci étant, par rapport à une gestion des connaissances essayant de prendre en compte la connaissance dans le cadre d'un système complexe se pliant difficilement à une démarche analytique [Le Moigne 1995], l'Ingénierie des connaissances propose ses méthodes et, justement, son savoir-faire [Barthès 1999, Charlet 2000a]. Les apports de l'Ingénierie des connaissances sont ainsi patents dans :

- la fourniture de ressources linguistiques [Bourigault 2000], que ce soit pour la construction d'index structurés permettant l'indexation et la récupération pertinente de documents divers dans des documentations techniques ou des textes scientifiques [Bourigault 1999] ;

- La création de ressources formelles permettant une meilleure indexation et récupération des expériences passées, en particulier dans le cadre de développements d'ontologies, structures formelles indispensables à la création et la maintenance de nombreux thesaurus « métiers » et pivots des représentations pour des systèmes d'information de plus en plus complexes et prenant en charge des informations de plus en plus hétérogènes (*cf.* les *data warehouse* [Zurfluh 2000]) ;

- La création d'environnements pour améliorer la mise en contexte des textes disponibles dans un système de gestion des connaissances et pour permettre une meilleure exploitation par l'homme, en particulier pour développer des ontologies permettant de baliser (XML) les informations contenues dans les textes [Benjamins 1999, Boury-Brisset 1999]. Par rapport à ce point, l'ingénierie documentaire, *ensemble des techniques numériques de conception, structuration, transformation, visualisation, de documents intervenant en particulier pour structurer les documents et en améliorer leur exploitation*, couplée aux réflexions de l'Ingénierie des connaissances, fournit les outils privilégiés pour mettre en place des systèmes de gestion des connaissances : en effet, la langue naturelle et les genres textuels des documents utilisés dans les entreprises, que ce soit au niveau de leur structure interne (organisation logique, découpage, etc.) ou externe (dossiers, regroupements, archivages, etc.) permettent d'exprimer l'information dans son contexte de création et de la proposer à l'utilisateur dans la forme qui permet

l'interprétation / la réinterprétation, *in fine*, la compréhension [Brunie 2000, Charlet 1999] ;

Finalement, l'ensemble des points et recherches abordés dans cette section (*cf.* 10.5) et visant à articuler la problématique des SBC et les systèmes d'information participent des réflexions de l'Ingénierie des connaissances pour intégrer et gérer les connaissances au sein du Système d'information.

10.6. Conclusion et Perspectives

Les nouvelles technologies produisent un contexte qui change profondément les rapports des acteurs et des organisations à l'information. Ce changement d'ère permet l'accès très large à de multiples sources d'information hors de l'entreprise, la connexion multiple des utilisateurs entre eux et nécessite des outils d'utilisation facile et intuitive. À terme, le contenu sémantique et la sophistication des fonctionnalités des applications ne peuvent que s'accroître, donnant, de ce fait, de plus en plus d'importance au traitement de la connaissance. Du point de vue de la recherche aussi bien pour les Systèmes d'information que pour l'Ingénierie des connaissances, l'enjeu est du côté conceptuel et du côté du lien avec les sciences humaines. Les technologies maîtrisées, c'est le contenu qui prime et il s'agit alors de savoir fournir aux métiers de l'entreprise les outils dont ils ont besoin. Pour les concepteurs de systèmes d'information, il y a donc une mutation de métier à opérer, en intégrant non seulement les aspects techniques des nouvelles technologies, mais en faisant le virage conceptuel intégrant le traitement de la connaissance et une analyse fine des besoins des utilisateurs et des contextes organisationnels.

L'acquisition des connaissances a joué un rôle d'investigation de premier plan : elle a accumulé des acquis importants en se consacrant à ce qui a été longtemps considéré comme une des difficultés principales de l'intelligence artificielle, en même temps que le centre de son activité, la connaissance. Même si ces travaux apparaissaient, dans un premier temps, spécialisés, obscurs et que l'ampleur de la tâche paraissait démesurée, les résultats accumulés permettent à présent de mieux cerner les enjeux dans les organisations, de se situer en tant qu'Ingénierie de la connaissance. Cette maturité permet de poser les hypothèses de recherche avec plus de validité sur un domaine balisé par les premiers résultats. Ainsi, comme nous l'avons vu, les principaux acquis de l'Ingénierie des connaissances se situent au niveau de la modélisation – *e.g.* le modèle conceptuel – et, plus fondamentalement, au niveau des ontologies, brique de base de toute

conceptualisation. Par ailleurs, la dimension sociale apportée par des disciplines comme la psychologie, l'ergonomie ou la sociologie ont permis de replacer ces modèles au regard des besoins des utilisateurs.

Parmi les enjeux qui se posent à l'Ingénierie des connaissances, face aux Systèmes d'information, deux nous semblent fondamentaux : (a) comprendre les impératifs organisationnels et les problèmes de travail coopératif comme d'évolution des organisations que cela pose et (b) réussir l'intégration de l'Ingénierie des connaissances au sein des Systèmes d'information, que ce soit pour y intégrer des SBC ou pour modéliser directement cette connaissance au sein des Systèmes d'information.

Concernant les aspects organisationnels, l'Ingénierie des connaissances collabore avec d'autres disciplines comme la gestion, la sociologie des organisations, l'ergonomie. Centrée sur l'objet complexe, situé et socialisé qu'est la connaissance, elle peut participer à une démarche d'explicitation globale des processus de traitement de l'information et des connaissances, par l'acteur et par l'organisation. Il s'agit de travailler avec les managers de toutes spécialités ayant une action sur la connaissance. Le travail coopératif utilisant l'artefact comme support se développe de plus en plus dans ce contexte. L'assister intelligemment suppose de construire des modèles conceptuels de l'activité coopérative aussi complexes et aussi variés que ceux des raisonnements individuels. Des bibliothèques de modèles d'activité coopératives restent à élaborer ; élaboration dans laquelle l'Ingénierie des connaissances a un rôle déterminant à jouer.

En quelques années, l'acquisition des connaissances a su devenir une Ingénierie des connaissances. Modélisant un objet et des processus complexes, elle a pour objectif de s'intégrer encore plus dans les ressources des concepteurs de systèmes d'information par la proposition de méthodes et d'outils reproductibles : l'Ingénierie des connaissances continue et doit continuer à produire des concepts et des méthodes reproductibles, évaluables, chiffrables pour modéliser à plusieurs niveaux, concevoir et de développer des systèmes spécialisés, des SBC. Elle a un rôle particulier à jouer pour construire des modèles et des outils favorisant et accompagnant la dynamique des connaissances dans l'organisation notamment lors des processus de changement organisationnel. Enfin, les recherches liées à la réutilisation ont atteint une certaine maturité en Ingénierie des connaissances. Leur confrontation avec les recherches en Systèmes d'information devrait permettre d'instaurer des collaborations rapides autour de la problématique des ontologies versus des référentiels, de travailler à moyen terme sur la réutilisation des modèles d'expertise au regard des *design patterns* et de développer des approches de modélisation cohérentes sinon similaires [Tort 2000].

Remerciements

Nos premiers remerciements vont à Bruno Bachimont qui est principalement à l'origine des réflexions épistémologiques développées ici. Nous remercions par ailleurs Georges Grosz qui nous a apporté ses réflexions sur les Systèmes d'information et l'Ingénierie des besoins. Enfin, merci à Camille Rosenthal-Sabroux et Corine Cauvet, de nous avoir permis d'inscrire cette réflexion dans le contexte, fertile, de la coopération entre Ingénierie des connaissances et Systèmes d'information.

Bibliographie

- [Argyris 1995] Argyris C., *Savoir pour agir*, Intereditions, 1995.
- [Aussenac-Gilles 1992] Aussenac-Gilles N., Krivine J.-P., Sallantin J., Éditorial, *Revue d'Intelligence Artificielle* vol. 6, 1992, p. 7-18. Special issue on Knowledge Acquisition.
- [Aussenac-Gilles 1997] Aussenac-Gilles N., Reynaud C., Tchounikine P., Associer un type de raisonnement à un domaine : une question de rôle ?, M. Zacklad, Ed., *Journées Ingénierie des Connaissances*, Roscoff, 1997.
- [Bachimont 1996] Bachimont B., « Herméneutique matérielle et artéfacture : des machines qui pensent aux machines qui donnent à penser. Critique du formalisme de l'intelligence artificielle » Thèse de philosophie, École Polytechnique, 1996.
- [Bachimont 2000] Bachimont B., Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en Ingénierie des Connaissances, in *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, J. Charlet, M. Zacklad, G. Kassel et D. Bourigault, Eds., 2000, Eyrolles.
- [Bannon 1993] Bannon L. J., Schmidt K., CSCW : four characters in search of a context., in *Readings in Groupware and computer-supported cooperative work*, B. R. M., Ed, 1993, Morgan Kaufmann.
- [Barthès 1999] Barthès J.-P., Dieng R., Kassel G., Dossier « Mémoire d'entreprise », in *Bulletin de l'AFIA* 1999, p. 34-64.
- [Benjamins 1999] Benjamins V. R., Fensel D., Decker S., Gomez Perez A., (KA)2 : building ontologies for the Internet : a mid-term report, *International Journal of Human-Computer Studies* vol. 51, 1999, p. 687-712.

- [Beys 1996] Beys B., Benjamins V. R., Van Heijst G., Remedying the Reusability - Usability Trade-off for Problem Solving Methods, B. Gaines et M. Musen, Eds., *KAW'96*, University of Calgary, 1996, Banff: SRDG Publications, p. 2-1/2-20.
- [Bézivin 1998] Bézivin J., Who Is Afraid of Ontologies?, *International Workshop on Model Engineering with CDIF (OOPSLA '98)*, Vancouver, Canada, 1998.
- [Bézivin 2000] Bézivin J., De la programmation par objets à la modélisation par ontologies : aspects de l'évolution actuelle du génie logiciel et de la technologie des composants, in *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, J. Charlet, M. Zacklad, G. Kassel et D. Bourigault, Eds., 2000, Eyrolles p. 209-22.
- [Bourigault 1999] Bourigault D., Charlet J., Construction d'un index thématique de l'Ingénierie des Connaissances, *IC'99*, École Polytechnique, Palaiseau, 1999.
- [Bourigault 2000] Bourigault D., Jacquemin C., Construction de ressources terminologiques, in *Ingénierie des langues*, J.-M. Pierrel, Ed, 2000, Hermès, Traité IC2.
- [Boury-Brisset 1999] Boury-Brisset A.-C., Gestion des connaissances d'une mémoire corporative construite autour d'ontologies, R. Teulier, Ed., *journées Ingénierie des Connaissances*, École Polytechnique, Palaiseau, 1999.
- [Breuker 1987] Breuker J., Wielinga B., Van Someren M., *et al.*, « Model Driven Knowledge Acquisition: Interpretation Models » *Deliverable A1, Esprit Project 1098, Memo 87, VF Project Knowledge Acquisition in Formal Domains*, 1987.
- [Brunie 2000] Brunie V., Bachimont B., Morizet-Mahoudeaux P., Modélisation des connaissances structurelles documentaires pour la conception d'un dossier médical hypertextuel, in *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, J. Charlet, M. Zacklad, G. Kassel et D. Bourigault, Eds., 2000, Eyrolles.
- [Causse 1993] Causse K., Heuristic Control Knowledge, N. Aussenac, G. Boy, M. Linster, J.-G. Ganascia et Y. Kodratoff, Eds., *Knowledge Acquisition for Knowledge Based Systems, EKAW'93*, Heidelberg, 1993, Springer-Verlag, p. 183-99.
- [Chandrasekaran 1986] Chandrasekaran B., Generic Tasks in Knowledge-Based Reasoning: High level Building Blocks for Expert System Design, *IEEE Expert* vol. 1, 1986, p. 23-30.
- [Charlet 1998] Charlet J., Bachimont B., De l'acquisition à l'ingénierie des connaissances : applications et perspectives, *Assises du PRC 13*, Lyon, 1998.
- [Charlet 1996] Charlet J., Bachimont B., Bouaud J., Zweigenbaum P., Ontologie et réutilisabilité : expérience et discussion, in *Acquisition et ingénierie des connaissances, Tendances actuelles*, N. Aussenac-Gilles, P. Laublet et C. Reynaud, Eds., 1996, Cépadués-Editions, Toulouse.
- [Charlet 1999] Charlet J., Daigne M., Leroux V., Ingénierie des patrimoines informationnels de l'établissement de santé, *Document numérique* vol. 3, 1999. Numéro spécial « Gestion des documents et gestion des connaissances ».

- [Charlet 2000a] Charlet J., Zacklad M., Kassel G., Bourigault D., Eds., *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, 2000a, Eyrolles.
- [Charlet 2000b] Charlet J., Zacklad M., Kassel G., Bourigault D., Ingénierie des connaissances : recherches et perspectives, in *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, J. Charlet, M. Zacklad, G. Kassel et D. Bourigault, Eds., 2000b, Eyrolles p. 1-22.
- [Clancey 1985] Clancey W., Heuristic Classification, *Artificial Intelligence* vol. 27, 1985, p. 289-350.
- [Clark 1996] Clark H. H., *Using language*, Cambridge University Press, 1996.
- [Clark 1993] Clark H. H., Brenann S. E., Grounding in communication, in *Readings in Groupware and computer-supported cooperative work*, B. R. M., Ed, 1993, Morgan Kaufmann.
- [Condamines 2000] Condamines A., Aussenac-Gilles N., Entre textes et ontologies formelles : les bases de connaissances terminologiques, in *Capitalisation des connaissances*, M. Zacklad et M. Grundstein, Eds., 2000, Hermès, Traité IC2.
- [Darses 1996] Darses F., Falzon P., La conception collective : une approche de l'ergonomie cognitive, in *Coopération et Conception*, D. Terssac, Ed, 1996, Octares, Friedberg.
- [De Kleer 1987] De Kleer J., Williams B., Diagnosing multiple Faults, *Artificial Intelligence* vol. 32, 1987, p. 97-130.
- [De Kleer 1989] De Kleer J., Williams B., Diagnosis with behavioral Modes, *IJCAI*, Detroit, 1989, p. 1324-30.
- [Delbos 1984] Delbos G., Jorion P., *La transmission des savoirs*, Éditions de la Maison des sciences de l'homme, Paris, 1984.
- [Delouis 1993] Delouis I., « LISA : un langage réflexif pour la modélisation du contrôle dans un système à base de connaissances » Thèse de doctorat, Université Paris-Sud, 1993.
- [Ermine 1996] Ermine J.-L., *Les systèmes de connaissance*, Hermès, 1996.
- [Falzon 1991] Falzon P., Cooperative Dialogues, in *Distributed Decision Making : Cognitive Models for Cooperation Work*, R. J., B. B. et L. J., Eds., 1991, John Wiley and sons ltd, London p. 145-89.
- [Fensel 1997] Fensel D., The Tower-of-Adapter Method for Developing and Reusing Problem-Solving Methods, *EKAW'97*, Sant Feliu de Guixols, Catalonia, Spain, 1997, p. 97-112.
- [Fensel 1991] Fensel D., Angele A., Landes D., KARL: a Knowledge Acquisition and Representation Language, *11th International conference Expert Systems and their Applications*, Avignon, 1991, p. 513-25.

- [Fensel 1994] Fensel D., Van Harmelen F., A comparison of languages which operationalize and formalize KADS Models of Expertise, *The Knowledge Engineering Review* vol. 9, 1994, p. 105-46.
- [Ganascia 1998] Ganascia J.-G., *Le Petit Tresor. Dictionnaire de l'informatique et des sciences de l'information*, Édition Flammarion, 1998.
- [Genesereth 1982] Genesereth M.-R., Diagnosis using Hierarchical Design Models, *AAAI*, 1982, p. 278-83.
- [Gréboval 1992] Gréboval C., « AIDE : Une approche et une architecture pour rendre opérationnels des modèles conceptuels » Thèse de doctorat, Université de technologie de Compiègne, 1992.
- [Grosz 2000] Grosz G., Rolland C., Ingénierie des besoins, in *Ingénierie des systèmes d'information*, C. Rosenthal-Sabroux et C. Cauvet, Eds., 2000, Hermès, Traité IC2. Chap. 4 (ce volume).
- [Guarino 1997] Guarino N., Understanding, building and using ontologies, *Int J Human-Computer Studies* vol. 46, 1997.
- [Hatchuel 1994] Hatchuel A., Apprentissages collectifs et activités de conception, *La Revue Française de gestion*, 1994.
- [Hatchuel 1996] Hatchuel A., Coopération et conception collective. Variété et crises des rapports de prescription, in *Coopération et conception*, D. Terssac, Ed, 1996, Octares, Friedberg.
- [Hayes-Roth 1983] Hayes-Roth F., Waterman D. A., Lenat D. B., *Building Expert Systems*, Addison-Wesley, New-York, 1983.
- [Karbach 1991] Karbach W., VoB A., Schuxkey R., Drouven U., MODEL-K: Prototyping at the Knowledge Level, *The Expert Systems and their Applications*, Avignon, 1991, p. 501-11.
- [Kayser 1988] Kayser D., Le raisonnement à profondeur variable, *Journées du Greco PRC-IA*, Toulouse, France, 1988, Teknea, p. 103-13.
- [Kayser 1997] Kayser D., La représentation des connaissances, , 1997.
- [Klinker 1991] Klinker G., Bholá C., Dallemagne G., Marquès D., Mc Dermott J., Usable and Reusable Programming Constructs, *Knowledge Acquisition* vol. 3, 1991, p. 117-36.
- [Le Moigne 1973] Le Moigne J.-L., *Les systèmes d'information dans les organisations*, PUF, Paris, 1973.
- [Le Moigne 1995] Le Moigne J.-L., *Les épistémologies constructivistes*, PUF, 1995.
- [Le Roux 1994] Le Roux B., « Eléments d'une approche constructive de la modélisation et de la réutilisation en acquisition des connaissances » Thèse de doctorat, Université Paris 6, 1994.
- [Lépine 1996] Lépine P., Aussenac-Gilles N., Modélisation de la résolution de problèmes : comparaison de KADS et MACAO sur une application juridique, in

- Acquisition et ingénierie des connaissances : tendances actuelles*, N. Aussenac-Gilles, P. Laublet et C. Reynaud, Eds., 1996, Cépaduès-éditions p. 131-50.
- [Lévine 1989] Lévine P., Pomerol J.-C., *Systèmes interactifs d'aide à la décision et systèmes experts*, Hermès, Paris, 1989.
- [Lewkowicz 1999] Lewkowicz M., Zacklad M., Memo-net, un collecticiel utilisant la méthode de résolution de problème DIPA pour la capitalisation et la gestion des connaissances dans les projets de conception, R. Teulier, Ed., *Journées Ingénierie des Connaissances*, Palaiseau, 1999.
- [Lewkowicz 2000] Lewkowicz M., Zacklad M., Une approche de la capitalisation des connaissances : l'analyse des processus de prise de décision collective, in *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, J. Charlet, M. Zacklad, G. Kassel et D. Bourigault, Eds., 2000, Eyrolles.
- [Linster 1992] Linster M., « Knowledge Acquisition Based on Explicit Methods of Problem Solving » Ph. D. Dissertation, D 386, Univ. of Kaiserslautern (G), 1992.
- [Lorino 1995] Lorino P., *Comptes et récits de la performance*, Les éditions d'organisations, 1995.
- [Marcus 1989] Marcus S., Mc Dermott J., SALT: a Knowledge Acquisition Tool for Propose and Revise Systems, *Artificial Intelligence* vol. 39, 1989, p. 1-37.
- [Mélèse 1990] Mélèse J., Au sujet de l'information, in *Approche Systémique des Organisations*, , 1990, Les éditions de d'Organisation, Paris p. 15-24.
- [Meynadier 1998] Meynadier J.-P., *Ingénierie et intégration des systèmes*, Hermès, 1998.
- [Molina 1996] Molina M., Shahar Y., Problem Solving Method Reuse and Assembly: From Clinical Monitoring to Traffic Control, B. R. Gaines et M. A. Musen, Eds., *KAW'96*, Alberta, Canada, 1996, SRDG Publications.
- [Morin 1986] Morin E., *La méthode : la connaissance de la connaissance*, Le Seuil, 1986.
- [Musen 1989] Musen M. A., Conceptual Models of Interactive Knowledge Acquisition Tools, *Knowledge Acquisition* vol. 1, 1989, p. 73-88.
- [Newell 1972] Newell A., Simon H., *Human Problem Solving*, Prentice-Hall, New jersey : Englewood Cliffs, 1972.
- [Nonaka 1997] Nonaka I., Takeuchi H., *La connaissance créatrice*. D. B. Université, Ed., , Bruxelles, 1997.
- [Norman 1986] Norman D. A., in *User-centered system design*, D. A. Norman et S. W. Draper, Eds., 1986, Lawrence Erlbaum associates.
- [Olson 1992] Olson D. L., Courtney J. F., *Decision support models and expert systems*, JR Maxwell MacMillan international editions, 1992.
- [Poitou 1996] Poitou J.-P., La gestion des connaissances comme condition et résultat de l'activité industrielle, *Intellectica* vol. 1, 1996, p. 185-202.

- [Pomerol 1997] Pomerol J.-C., Artificial intelligence and human decision making, *European Journal of Operational Research* vol. 99, 1997, p. 3-25.
- [Prekas 1999] Prekas N., Loucopoulos P., Rolland C., *et al.*, Using patterns as a mechanism for assisting the management of knowledge in the context of conducting organisational change, *10th international conference and workshop on database and expert systems application (DEXA '99)*, Florence, Italy, 1999, p. 110-22.
- [Puerta 1992] Puerta A. R., Egar J., Tu S. W., Musen M. A., A multiple-Method Knowledge Acquisition Shell for the Automatic generation of Knowledge Acquisition Tools, *Knowledge Acquisition* vol. 4, 1992, p. 171-96.
- [Reiter 1987] Reiter, A theory of Diagnosis from first principles, *Artificial Intelligence* vol. 32, 1987, p. 57-95.
- [Reynaud 2000] Reynaud C., Tort F., Diriger la réutilisation de composants à l'aide d'ontologies, in *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, J. Charlet, M. Zacklad, G. Kassel et D. Bourigault, Eds., 2000, Eyrolles.
- [Rognin 1998] Rognin L., Working in control Rooms cooperation and communication in shared Workspaces, F. Darses et P. Zaraté, Eds., *COOP'98*, Cannes, 1998, INRIA.
- [Rosenthal-Sabroux 1996] Rosenthal-Sabroux C., « Contribution méthodologique à la conception de systèmes d'information coopératifs : prise en compte de la coopération homme-machine » Habilitation à diriger les recherches, Université Paris Dauphine, 1996.
- [Salembier 1994] Salembier P., Assistance coopérativeaux activités complexes : l'exemple de la régulation du trafic aérien, in *Systèmes coopératifs : de la modélisation à la conception*, B. Pavard, Ed, 1994, Octares.
- [Schreiber 1994] Schreiber G., Wielinga B., Akkermans H., Van de Velde W., Anjewierden A., CML: The CommonKADS Conceptual Modelling Language, L. Steels, G. Schreiber et W. V. d. Velde, Eds., *EKAW'94*, 1994, Springer-Verlag, p. 1-25.
- [Shannon 1971] Shannon C.-E., Weaver W., Information, in *The Mathematical Theory of Communication*, , 1971, University of Illinois Press p. 8-16. 12e édition (édition originale, 1949).
- [Steels 1985] Steels L., Second Generation Expert Systems, *Future Generation in Computer Systems* vol. 1, 1985.
- [Steels 1990] Steels L., Components of Expertise, *AI Magazine* vol. 11, 1990.
- [Steels 1993] Steels L., The componential Framework and its Role in Reusability, in *Second Generation Expert Systems*, J. M. David, J. P. Krivine et R. Simmons, Eds., 1993, Springer verlag p. 273-8.
- [Talbi 2000] Talbi N., Laublet P., Construction d'un modèle générique d'expertise à partir d'un exemple : une solution pragmatique, in *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, Jean, Charlet, M. Zacklad, G. Kassel et D. Bourigault, Eds., 2000, Eyrolles.

- [Tardieu 1991] Tardieu H., Guthmann B., *Le triangle stratégique : stratégie, structure et technologie de l'information*, Éditions d'Organisations, 1991.
- [Tchounikine 2000] Tchounikine P., Istenes Z., Trichet F., ZOLA : un langage permettant une approche flexible de l'opérationnalisation du modèle conceptuel d'un système à base de connaissances, in *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, J. Charlet, M. Zacklad, G. Kassel et D. Bourigault, Eds., 2000, Eyrolles.
- [Teulier 2000] Teulier R., Quelles connaissances actionnables pour la cognition collective ?, in *Ingénierie des pratiques collectives. La cordée et le quatuor*, M.-J. Avenier, Ed, 2000, L'harmattan.
- [Teulier-Bourgine 1997] Teulier-Bourgine R., Les représentations, médias de l'action stratégique, in *La stratégie chemin faisant*, M.-J. Avenier, Ed, 1997, Economica.
- [Tort 2000] Tort F., Teulier R., Grosz G., Charlet J., Ingénierie des besoins, ingénierie des connaissances : similarités et complémentarités des approches de modélisation, P. Tchounikine, Ed., *journées Ingénierie des Connaissances*, Toulouse, 2000.
- [Tu 1995] Tu S., Eriksson H., Gennari J., Shahar Y., Musen M., Ontology-Based Configuration of Problem-Solving Methods and Generation of Knowledge Acquisition Tools: Application of PROTEGE II to Protocol-Based Decision Support, *Artificial Intelligence in Medicine* vol. 7, 1995, p. 257-89.
- [Turban 1993] Turban E., *Decision Support and Expert Systems, Management Support Systems*, MacMillan, 1993.
- [Van Harmelen 1992] Van Harmelen F., Balder J., (ML)2 : A Formal Language for KADS Models of expertise, *Knowledge Acquisition* vol. 4, 1992, p. 127-61.
- [Van Heijst 1997a] Van Heijst G., Schreiber A. T., Wielinga B. J., Roles are not classes: a reply to Nicola Guarino, *Int J Human-Computer Studies* vol. 46, 1997a.
- [Van Heijst 1997b] Van Heijst G., Schreiber A. T., Wielinga B. J., Using Explicit Ontologies in Knowledge Based Systems Development, *Int J Human-Computer Studies* vol. 46, 1997b, p. 183-292.
- [Wetter 1990] Wetter T., First Order Logic Foundations of the KADS Conceptual Model, in *Current trends in Knowledge Acquisition*, B. Wielinga, Ed, 1990, IOS Press, Amsterdam p. 356-75.
- [Wielinga 1992a] Wielinga B., Schreiber G., Breuker J., KADS: A Modelling Approach to Knowledge Engineering, *Knowledge Acquisition* vol. 4, 1992a, p. 5-54.
- [Wielinga 1992b] Wielinga B., Van de Velde W., Schreiber G., Akkermans H., The CommonKADS Framework for Knowledge Modelling, *KAW'92*, Banff, 1992b.
- [Wielinga 1993a] Wielinga B., Van De Velde W., Schreiber G., Akkermans H., « Expertise Model Definition Document » *Kads I/M2/UVa/026/1.1*, 1993a.
- [Wielinga 1993b] Wielinga B., Van De Velde W., Schreiber G., Akkermans H., Towards a Unification of Knowledge Modelling Approaches, in *Second Generation of*

Expert Systems, J. M. David, J. P. Krivine et R. Simmons, Eds., 1993b, Springer-Verlag p. 299-335.

[Zachary 1990] Zachary W. W., Robertson S. P., Introduction to Cognition, computation and cooperation Cognition, in *Computing and cooperation*, S. P. Robertson, W. Zachary et J. B. Black, Eds., 1990, Ablex Publishing Corporation.

[Zacklad 1994] Zacklad M., groupe COOP, MadeIn'COOP : préliminaires pour le développement d'une méthodologie de modélisation de la coopération homme-machine, *Ergo-IA*, Biarritz, 1994, p. 406-21.

[Zacklad 2000] Zacklad M., Grundstein M., Eds., *Capitalisation des connaissances*, 2000, Hermès. Traité IC2.

[Zarató 2000] Zarató P., Advances for Decision Support Systems, *Journal of the Spanish Royal Academy of Sciences*, 2000. À paraître en juin 2000.

[Zurfluh 2000] Zurfluh G., Guessarian I., Bellahsene Z., *et al.*, Entrepôt de données pour l'aide à la décision, in *Ingénierie des systèmes d'information*, C. Rosenthal-Sabroux et C. Cauvet, Eds., 2000, Hermès, Traité IC2. Chap. 7 (Ce volume).

Index

- approche par analyse du domaine,33
- base de connaissances,11
- besoins de l'utilisateur,27,30
- bibliothèque de composants,18
- bibliothèque de modèles,14
- changement organisationnel,28
- cognition collective,24
- Computer Supported Cooperative Work*
Voir CSCW
- connaissance,5,6,22
- connaissances du domaine,9,14
- connaissances explicites,23
- connaissances pour l'action,23
- connaissances tacites,23
- contexte d'utilisation,6,8
- contexte situé,24
- coopération,30,31
- CSCW,31
- data warehouse*,26,35
- design pattern*,28,32,37
- domaine,2
- donnée,5
- dynamique organisationnelle,22
- épistémologies constructivistes,6,8
- épistémologique,4
- frameworks*,32
- génie logiciel,9,17
- gestion des connaissances,33
- groupware*,31
- information,5
- ingénierie,3,28
- Ingénierie des besoins,27,29,32
- ingénierie des composants,32

- Ingénierie des
 - connaissances,1,3,7,8,29,34
 - ingénierie documentaire,,35
 - interface homme-machine,27

- langage de modélisation,14
- langage de représentation,15
- langage d'opérationnalisation *Voir*
 - langage opérationnel
- langage opérationnel,14,15

- médiation externe,4
- méthode,2,12,16
- modèle,9
- modèle conceptuel,2,9,12,16
- modèle d'interprétation,19
- modèle de raisonnement,12,14,19
- modèle du domaine,10,12
- modèle générique,13
- modèle spécifique,13

- ontologie,10,11
- ontologie de méthodes,13
- ontologie de modèles,13
- ontologie du domaine,11,12
- organisation apprenante,22

- patron *Voir design pattern*
- primitive,6

- représentation des connaissances,4
- réutilisabilité *Voir réutilisation*

- réutilisation,3,13,19,21,32
- réutilisation de composants,12
- rôle,12,13,14,16

- savoir actionnable,23
- savoir applicable,23
- savoir-faire,21,23
- SBC,1,6,7,8,26
- schéma du modèle conceptuel,16
- SE,1,26
- SGBD,25
- système coopératif,29
- système d'information,3,23,24,30
- système de gestion de bases de données
 - Voir* SGBD
- système de gestion des connaissances,34
- système expert *Voir* SE
- systèmes à base de connaissances *Voir*
 - SBC
- systémique,4,5

- tâche,2,12,16
- terme,6
- travail collaboratif,30
- travail collectif,30,31
- travail de groupe *Voir groupware*

- utilisateur,29

- workflow*,31

Table des matières

10.1. Introduction.....	1
10.2. Épistémologie et définitions.....	3
10.2.1. De l'information à la connaissance.....	4
10.2.1.1 Point de vue épistémologique.....	4
10.2.1.2 Point de vue systémique.....	5
10.2.2. Des systèmes à base de connaissances.....	6
10.2.3. Des systèmes dans une organisation.....	8
10.3. Modélisation des connaissances.....	9
10.3.1. Nature et contenu des modèles.....	9
10.3.1.1. Des connaissances du domaine aux ontologies.....	10
10.3.1.2. Les modèles de raisonnement.....	12
10.3.1.3. La combinaison réfléchie des modélisations domaine et raisonnement.....	13
10.3.2. Niveau de description des connaissances.....	14
10.3.3. Des langages de modélisation.....	15
10.3.4. Le processus de modélisation.....	16
10.3.4.1. Des architectures spécialisées.....	17
10.3.4.2. Des architectures génériques.....	18
10.3.5. La réutilisation.....	20
10.4. Le contexte organisationnel et son impact sur les Systèmes d'information et l'Ingénierie des connaissances.....	22
10.4.1. Les organisations prennent conscience de l'importance stratégique des connaissances.....	23
10.4.2. Les connaissances dans les organisations.....	24

50 Ingénierie des systèmes d'information

10.4.3. La place des connaissances dans cette évolution vers la complexification des systèmes d'information.....	26
10.4.4. Les systèmes à base de connaissances dans les organisations.....	27
10.5. Articuler SBCs et systèmes d'information.....	29
10.5.1. L'utilisateur au centre de la conception.....	29
10.5.2 Vers des systèmes d'information coopératifs.....	31
10.5.3. Vers une ingénierie des composants.....	34
10.5.4 Vers des systèmes de gestion des connaissances.....	35
10.6. Conclusion et Perspectives.....	37
Remerciements.....	39