
Adaptation de KADS pour la construction de Systèmes à Base de Connaissances explicatifs

Philippe MARTIN

*INRIA Sophia Antipolis - Projet ACACIA
2004, Route des Lucioles - B.P. 93 - 06902 Sophia-Antipolis Cedex (France)
e-mail : phmartin@sophia.inria.fr - tél. : 93 65 76 45*

RÉSUMÉ : Pour aider le cognicien à construire un système à base de connaissances capable de fournir de bonnes explications, nous étendons dans cet article le Modèle de Coopération de la méthodologie KADS. Ce modèle, qui fait partie du Modèle Conceptuel de KADS, permet de spécifier à un haut niveau la coopération entre le futur système et ses utilisateurs. Nous proposons de construire ce modèle sous la forme de deux modèles d'expertise : le «Modèle d'Expertise de la Coopération» et le «Modèle d'Expertise de la Communication». Ils sont similaires au classique «Modèle d'Expertise (de résolution de problèmes)» mais modélisent d'autres expertises : 1) le partage du travail avec l'utilisateur pour le premier modèle 2) les problèmes complexes des échanges de données avec l'utilisateur pour le second, ce qui inclut l'interprétation de données ou de requêtes ainsi que la présentation de données ou d'explications.

Grâce à ces deux nouveaux modèles, il est possible de spécifier les connaissances nécessaires au futur système pour générer de bonnes explications. Dans cette optique, nous détaillerons surtout le «Modèle d'Expertise de la Communication». Une bibliothèque de modèles d'interprétations pourraient être construite pour aider le cognicien dans l'extraction et l'organisation de connaissances pour ces deux modèles. Nous donnons à titre d'exemple, quelques structures d'inférences assez générales sur les tâches d'interprétation des requêtes et de présentation des réponses.

MOTS-CLES : acquisition des connaissances, KADS, explications, modèles d'expertise, modèles d'interprétation.

1. Introduction

Dans un Système à Base de Connaissances (SBC), la génération de bonnes explications implique de nombreuses tâches complexes et interdépendantes : choisir le type de coopération avec l'utilisateur, gérer le dialogue, interpréter les requêtes, choisir les éléments de réponses et les structurer, puis les présenter à l'utilisateur sous forme de phrases et/ou de graphiques. De plus, pour être utiles ou efficaces, ces tâches doivent prendre en compte les buts et les connaissances de l'utilisateur.

Générer de bonnes explications nécessite donc une expertise certaine. Dans la méthodologie d'acquisition des connaissances KADS, le Modèle Conceptuel - aboutissement de la phase d'analyse des connaissances nécessaires au futur SBC - comprend un Modèle d'Expertise (de résolution de problèmes) et un Modèle de Coopération. Le premier, structuré en quatre niveaux avec le langage KCML, permet de bien modéliser l'expertise de résolution nécessaire au SBC, mais le second ne permet pas de spécifier de la même manière l'expertise nécessaire pour les explications. Dans [De Greef et Breuker 92], il est proposé de construire un modèle supplémentaire, le «Modèle de Communication», pour décrire les fonctions d'interactions avec les utilisateurs. Nous allons plus loin : *ce nouveau Modèle de Communication et le Modèle de Coopération devraient, lorsque les connaissances qu'ils doivent contenir sont suffisamment nombreuses, être structurés comme des modèles d'expertise*, i.e. en employant le langage KCML.

Dans la seconde section, nous montrons l'intérêt de ce choix, et les éléments que peuvent exploiter ces deux nouveaux modèles d'expertise. Puis en troisième section, nous précisons comment construire le modèle de communication comme un modèle d'expertise.

2. Deux nouveaux modèles d'expertise

Classiquement, construire le *Modèle de Coopération* consiste :

- 1) à définir sommairement les tâches de transfert, i.e. les tâches de transmission ou de réception de données ou de résultats avec l'utilisateur, ce qui inclut les explications et les dialogues; pour cela, le cogniticien précise qui, de l'utilisateur ou du système, a l'*initiative* de l'échange, et quel est le *type d'«ingrédients»* transférés : «information», «connaissance», «talent», etc. [de Greef et Breuker 92]. Pour spécifier les fonctions d'interactions de ces tâches de transfert, il est donc nécessaire de construire **en plus**, un «Modèle d'Expertise de Communication» (gérant par exemple l'interprétation de requêtes, la présentation de réponses et la gestion de dialogue).
- 2) à répartir le travail prévu entre l'utilisateur et le système en définissant par un algorithme, l'alternance prévue entre les tâches de transfert et les sous-tâches de résolution lors de la résolution d'un type de problème.

Or cette répartition du travail peut dépendre des buts, connaissances et aptitudes de l'utilisateur, des priorités des divers objectifs à atteindre, et des diverses connaissances de résolution disponibles. Elle peut aussi être le résultat d'interactions avec l'usager.

L'expertise impliquée peut donc être grande et nécessiter un «Modèle d'Expertise de la Coopération» pour être spécifiée. Nous utiliserons désormais l'abréviation «ME-Coopération». De même, nous noterons «ME-Communication» le «Modèle d'Expertise de Communication» et «ME-Résolution» le classique Modèle d'Expertise (de résolution).

Pour gérer le partage du travail ou tout autre fonction coopérative, les tâches du ME-Coopération appellent et/ou orientent les tâches de résolution et celles de communication en leur fixant ou mettant à jour des objectifs, i.e. respectivement des buts de résolution et des buts de discours. Par exemple, pour une tâche de dialogue, le ME-Coopération peut choisir les «*messages*» à faire passer auprès de l'utilisateur ou fixer plus ou moins les *sujets ou concepts* à aborder et développer. Il doit aussi déterminer si au cours du dialogue l'utilisateur a modifié l'*objectif central* et si de nouveaux sujets doivent être abordés. En effet, les tâches du ME-Communication ne peuvent assurer une certaine coopération que relativement à des objectifs pré-établis. Donc on a le schéma suivant pour le Modèle Conceptuel.

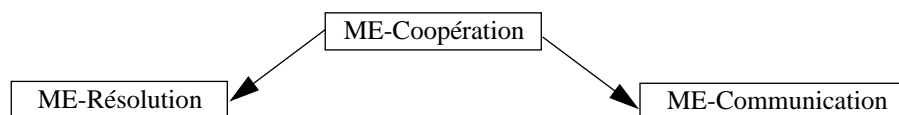


Figure 1 : le ME-Coopération contrôle les deux autres modèles d'expertise du Modèle Conceptuel

Les extractions et modélisations de chacune de ces trois expertises apportent informations et contraintes aux deux autres. Durant la phase de conception, chacun de ces modèles est affiné et complété en fonction des contraintes matérielles et logicielles (e.g. les interfaces avec l'utilisateur, pour les stratégies coopératives et/ou explicatives). Durant la phase d'analyse, la descriptions des langages de communication système-utilisateur peut demeurer à un haut niveau d'abstraction en s'appuyant, par exemple, sur les concepts de média séquentiels (textes, sons, etc.) et spatiaux (images, graphiques).

Cette présentation du ME-Coopération montre que la partie la plus importante de l'expertise liée aux explications doit se modéliser dans le ME-Communication. Nous allons voir comment dans la prochaine section. Mais tout d'abord, afin de mieux définir ces deux modèles, et donc d'aider à les construire, voici une liste de connaissances que leur tâches peuvent exploiter pour des explications ou autres buts coopératifs similaires :

- le *ME-Résolution* fournit des descriptions explicites, structurées et de haut niveau sur les connaissances de résolution de problèmes du SBC. Comme montré dans [Chandrasekaran et al. 89], ce type de descriptions est indispensable aux tâches explicatives pour abstraire les *données d'un problème* et les *traces de résolution antérieures*, raisonner dessus, et ainsi fournir des explications compréhensibles par l'utilisateur. Ce travail

d'abstraction est facilité dans KADS par la conservation de l'architecture du Modèle Conceptuel durant la phase de conception. On peut trouver dans [Baumewerd et al. 91] et [Sprenger 92] des méthodes permettant d'exploiter le ME-Résolution pour l'interprétation des requêtes de l'utilisateur et la détermination du contenu des explications.

- des *descriptions explicatives sur les éléments du ME-Résolution* sont également nécessaires : définitions, figures, «schémas explicatifs» i.e. manières particulières d'expliquer un élément (concept, inférence, tâche, etc.), connaissances profondes telles des réseaux causaux justificatifs, etc. Dans KADS, la modélisation des tâches de résolution est assez indépendante des concepts de l'expertise. Les schémas explicatifs associés à ces tâches peuvent donc soit conserver cette généralité, soit au contraire expliquer la tâche avec les concepts du domaine de l'expertise.

- le «*Modèle de l'Utilisateur*» (MU) stocke relativement à chaque type d'utilisateur, un certain nombre d'informations (suivant les besoins explicatifs ou coopératifs) : buts, centres d'intérêts, types de questions, connaissances sur le domaine, préférences, etc. Pour un suivi précis de l'utilisateur, le MU peut être mis à jour après chaque interaction avec celui-ci. La construction de MU reste cependant surtout du domaine de la recherche.

- enfin, l'*historique des échanges d'informations ou dialogues avec l'utilisateur*, ainsi que l'*historique des générations d'explication* (buts poursuivis et choix faits pour chaque génération) sont indispensables pour poursuivre des explications antérieures, répondre à des questions sur celles-ci, ou encore raisonner sur les dialogues [Paris 92] [Cawsey 91].

Cette liste de «*connaissances de support*» pour les *explications* n'est pas exhaustive mais tous ses éléments sont nécessaires et doivent être exploités au mieux afin de fournir des explications complètes ou adaptées à l'interlocuteur. Les connaissances ou mécanismes liés à ces exploitations doivent donc être acquis ou définis selon les besoins en explications. Les structures d'inférences de ces tâches d'exploitation auront donc des «méta-classes» sur des éléments de cette liste (les méta-classes, ou «rôles», permettent d'abstraire les rôles des concepts impliqués dans une inférence).

3. Le Modèle d'expertise de Communication

Le ME-Communication est essentiellement composé de l'expertise relative

- 1) à l'interprétation de données ou de requêtes ainsi que la présentation de données ou d'explications;
- 2) à la gestion du dialogue. Comme les tâches de dialogues appellent celles d'interprétation et de présentation, voyons tout d'abord les connaissances qu'elles impliquent.

3.1. L'expertise de dialogue

Pour la gestion du dialogue, il faut inclure dans le ME-Communication, et de façon plus ou moins développé suivant les besoins en explications :

- au niveau *domaine* : un modèle de dialogue (conventions, règles de construction et de coopération), les protocoles de «prise de parole», les concepts de «topiques» (sujets ou contenus, introduits par l'utilisateur ou le système au cours du dialogue), et tous les autres *concepts* et relations nécessaires aux niveaux inférence et tâche, par exemple ceux associés à l'exploitation des Modèles Utilisateurs.
- aux niveaux *inférence, tâche, et stratégie* : les divers mécanismes *gérant* les topiques, la «prise de parole», l'exploitation des connaissances de support pour les explications, et l'application d'objectif (i.e. quels sous-topiques ou nouveaux topiques introduire pour satisfaire au mieux l'objectif fixé, en fonction des connaissances de l'utilisateur et de ses attentes; l'objectif pouvant être donné par le ME-Communication ou être issu d'une requête de l'utilisateur). Ceci inclut :
 - 1) les *explications spontanées* (i.e. données pour prévenir des requêtes prévisibles),
 - 2) l'*adaptation* aux nouveaux buts et connaissances de l'utilisateur,
 - 3) le *maintien* et l'exploitation du contexte du dialogue (concepts précédemment utilisés de part et d'autre, thèmes abordés, etc.).

Une gestion simple mais assez efficace du dialogue est proposée dans [Cawsey 91]. Elle utilise une méthode de planification hiérarchique. Nous allons voir plus loin, les diverses modélisations possible d'un tel contrôle et les connaissances qu'il met en oeuvre. Notons que pour être utiles aux tâches de dialogue, les tâches de présentation doivent savoir réagir aux interruptions et pouvoir répondre à des questions sur des explications déjà présentées.

3.2. L'expertise de présentation

Pour aider le cognicien à extraire et organiser les connaissances de résolution en vue de la construction du ME-Résolution, KADS propose une bibliothèque de *modèles d'interprétation*, i.e. essentiellement des structures d'inférences commentées (cf [Breuker (ed) 87]). Nous proposons de telles structures d'inférences pour aider à modéliser l'*interprétation de données* venant de l'utilisateur et la *présentation d'information* à celui-ci. Ainsi peut être spécifiée l'essentiel de l'expertise associée aux deux grandes classes de tâches de transfert : réception de données / transmission de données.

Les sources de connaissances des structures d'inférence sont désormais notées SC.

3.2.1. Structure d'inférence de l'interprétation de données ou de requête

L'*interprétation* consiste à produire, à partir de données exprimées dans un langage compréhensible par l'utilisateur, une *structure de données* dont le format est compréhensible par la machine. Cette transformation peut être triviale (e.g. transformation d'un chiffre en son code binaire; association d'une commande de menu à la procédure qui l'exécute), de difficulté moyenne (e.g. requête exprimée dans un langage de requêtes) voire très complexe (e.g. interprétation d'une phrase en langage naturel).

Dans les cas complexes, l'interprétation (SC «*transformer/étendre/sélectionner*») dans la figure ci-dessous s'accompagne d'une recherche des *références implicites* dans les données («*étendre*»), et d'une suppression des éléments inutiles ou redondants («*sélectionner*»). Puis, la structure de donnée ainsi obtenue, est éventuellement *retransformée* en un langage compréhensible de l'utilisateur pour lui faire part de l'interprétation qui a été faite de ses données. D'où la structure d'inférence suivante, utilisant les termes conseillés par KADS :

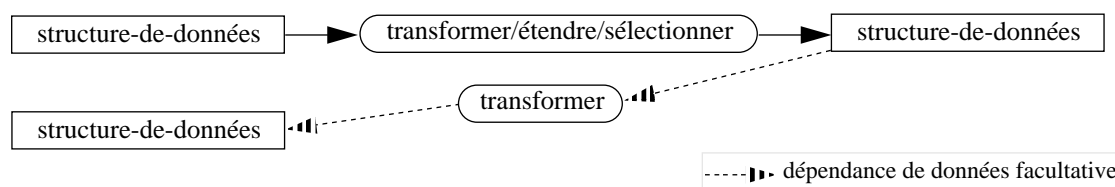


Figure 2: L'interprétation de données ou de requête

Lorsqu'il suit cette structure d'inférence, le cognicien doit définir ou acquérir pour les niveaux **domaine et inférence** de son ME-Communication, les différents *concepts et méthodes* relatifs

- 1) aux *divers* langages utilisés : celui permettant à l'utilisateur d'exprimer sa requête, celui qui en représente le sens en machine et, s'il est différent du premier, celui nécessaire pour communiquer à l'utilisateur l'interprétation de sa requête (pour chaque langage, il faut définir son lexique, sa syntaxe et éventuellement sa sémantique et sa pragmatique);
- 2) aux transformations d'un langage vers un autre;
- 3) à l'*utilisation* des «connaissances de support», e.g. l'historique du dialogue et le MU pour la recherche des éléments implicites. Notez que les éléments du ME-Résolution sont à la base des lexiques des langages utilisés. La mise en correspondance des éléments de la requête avec ceux du ME-Résolution participe à la fois à l'interprétation de cette requête et à l'apport d'éléments de réponses [Baumewerd et al. 91] [Sprenger 92];
- 4) à la mise à jour de quelques connaissances de support : historique de Présentation et éventuellement MU et historique de Dialogue.

Au niveau **tâche**, le cognicien définit le contrôle de l'enchaînement de ces méthodes en fonction de la souplesse attendue dans la transformation, de l'interactivité avec l'utilisateur, etc.

3.2.2. Structures d'inférence de la présentation de données ou d'explications

Le but de la «présentation» est ici de fournir dans un langage compréhensible par l'utilisateur, des *informations* sur des objets du système, généralement ceux du ME-Résolution, e.g. connaissances ou résultat de raisonnement. La structure d'inférence que nous proposons ci-dessous (figure 3), traite le cas le plus complexe, celui de la génération d'explications. Elle a donc pour entrée une demande d'explication sur un objet connu par le système. S'il s'agit d'une explication spontanée, cette requête est issue d'une tâche du système, sinon elle est la formalisation d'une requête de l'utilisateur (cf figure 2).

L'objet à expliquer est soit fourni également, soit généré ou recherché à partir de la requête : dans la figure 3, SC «résoudre» si une résolution est nécessaire, SC «sélectionner» si l'objet est une connaissance d'un modèle d'expertise.

A partir de ces données, la tâche de présentation consiste à rechercher des éléments de réponse dans le modèle d'expertise concerné par la requête (donc généralement dans le ME-Résolution et les descriptions explicatives associées à ses éléments) puis à les assembler pour créer une **structure intentionnelle** combinant les effets que chaque «expression» doit avoir sur l'interlocuteur. Globalement, toutes les techniques actuelles de génération de discours travaillent ainsi pour atteindre des **buts du discours**.

Certaines techniques opèrent une *pré-sélection* de contraintes ou de connaissances à appliquer pour la sélection des éléments de réponse. Ceci est détaillé dans [Suthers 91] sous le nom de «détermination de la perspective» (choix des concepts, propriétés et relations qui serviront de base à l'explication). Cette *pré-sélection* peut s'effectuer avant la *sélection* ou au cours de celle-ci (alternance des étapes). Comme la sélection et l'assemblage peuvent également être alternés, ces inférences ont été réunies dans une seule SC («sélectionner/assembler»). Nous verrons plus loin une structure d'inférence plus précise, notamment sur cette «sélection».

La *structure intentionnelle* doit ensuite être *raffinée* (SC «transformer/étendre/raffiner») avant d'être donnée en entrée à un générateur de «phrases» qui la *transforme* pour que son «transfert» sur le support physique soit compréhensible et naturel pour l'utilisateur.

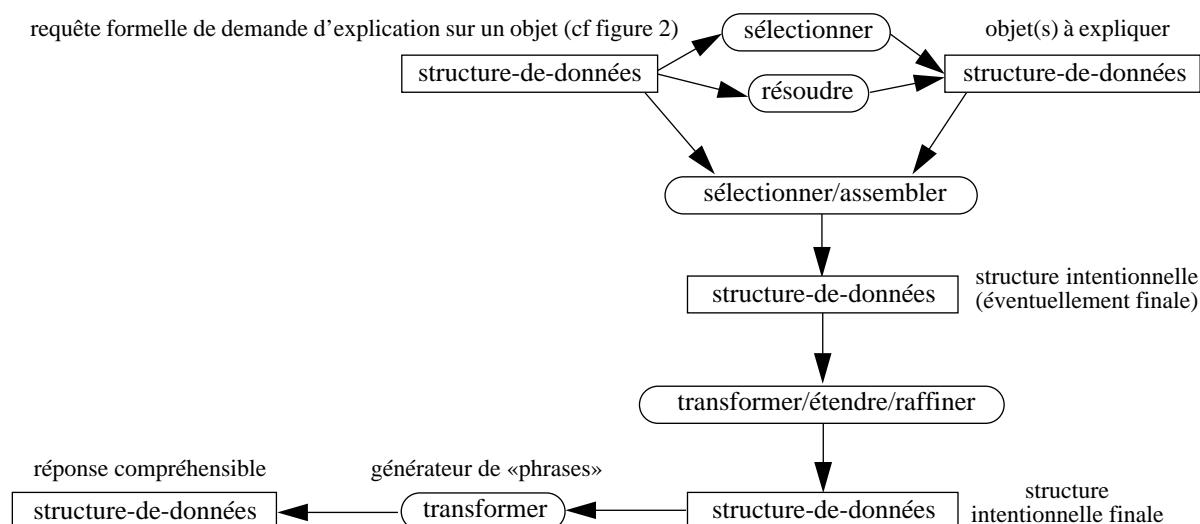


Figure 3: Structure d'inférence générale de la génération d'explications

Au cours de chacune de ces trois grandes phases, tous les éléments des *connaissances de support* que nous avons vues, peuvent être exploités. Ils peuvent également être mis à jour, exceptés bien sûr ceux du ME-Résolution.

Ces **trois phases** sont nécessaires. La dernière est utile, pour des raisons de simplicité et d'efficacité puisque les techniques de planification de discours, qui créent la structure intentionnelle, ne doivent pas gérer en plus les niveaux de détails inférieurs : lexicale, grammaticale, choix d'expressions ou de tournures de phrase adaptés à l'utilisateur. Ceci est le rôle du **générateur de «phrases»**, lequel peut également afficher des graphiques (ou des

images, etc.) si *le langage destiné à l'utilisateur* admet l'inclusion ou la gestion de ces graphiques et si donc, ils sont inclus dans la structure intentionnelle.

De plus, la structure intentionnelle ne peut être générée d'un seul coup ([Lemaire 92], [Suthers 91]). Après le *mécanisme de planification descendante* généralement utilisé pour déterminer cette structure, des *mécanismes dirigés par les données* sont nécessaires pour la corriger et l'étendre. Ceci, en fonction de *règles pédagogiques*, des *connaissances* de l'utilisateur ou de ses *préférences*, etc.

Pour élaborer cette description et ainsi la modélisation des tâches explicatives, nous devons introduire la notion «d'opérateurs». Tous les mécanismes actuels de création et de raffinement de cette structure font appel à des «**opérateurs**» : sortes de règles avec des conditions de déclenchement, pouvant s'appeler entre elles, et dirigées par les buts ou par les données. Par exemple, des opérateurs dirigés par les données peuvent être sensibles au type des éléments de la structure intentionnelle ou à leur agencement.

Voici à titre d'exemple, une paraphrase du code de l'opérateur «persuader» de [Moore et Paris 91]. Les mots clefs et les noms d'opérateurs sont en italique.

Pour qu'un système s puisse *persuader* un utilisateur u d'accepter pour *but* de *réaliser* un acte a, il doit vérifier que a est une *étape* dans l'accomplissement d'au moins un *but* commun à s et à u puis *motiver* a par chacun des buts communs à s et à u.

Les opérateurs permettent d'obtenir l'adaptabilité des processus d'explication à divers contextes et en donnent des représentations courtes et claires (car déclaratives). Comme les opérateurs représentent actuellement la meilleure façon d'aborder la complexité de la génération d'explications, voyons comment ils peuvent être modélisés dans KADS.

1) Certains opérateurs sont indépendants du domaine de l'application et peuvent donc être modélisés comme des *méthodes d'inférences* (pour plus de détails sur ces dernières, cf [Breuker (ed) 87] [Weilinga et al. 92]). Des recherches sont faites pour créer une bibliothèque de telles méthodes indépendantes du domaine [Safar 92].

2) Les autres opérateurs, et les concepts explicatifs qu'ils utilisent, doivent être modélisés dans le *niveau domaine* du ME-Communication. Dans ce cas, les *méthodes d'inférences* des SC ont pour but de *contrôler* l'application de un ou plusieurs opérateurs de domaine.

Dans les deux cas, les SC peuvent être plus ou moins complexes : les méthodes d'inférences d'une SC peuvent être *simples* et la SC **appelée plusieurs fois** successivement; c'est alors le contrôle au niveau tâche qui est plus compliqué. Nous laissons au concepteur ce choix de modélisation.

Nous allons nous appuyer sur ce principe de modélisation d'opérateur pour présenter, avec la figure 4 ci-dessous, une structure d'inférence possible *pour la construction et le raffinement de la structure intentionnelle* (centres des travaux actuels sur la génération d'explications). Il ne s'agit pas d'une solution idéale, mais d'une *synthèse dans une formalisation KADS* de travaux actuels **combinant** diverses méthodes de génération de discours : [Suthers 91], [Zukerman 91], [Lester et Porter 91], [Lemaire et Safar 91].

Toutes les étapes possibles dans la génération ont été retenues. Bien que toutes ces étapes soient **utiles** pour avoir de «bonnes» explications, la plupart des travaux actuels en omettent un certain nombre. Lorsque les quatre travaux de référence ci-dessus divergent sur l'ordre de certaines étapes, celui-ci n'est pas précisé. Ce sera alors **au concepteur de le définir dans le niveau tâche** en fonction de son application et de ses exigences en explications. Dans [Lemaire et Safar 91] les auteurs conseillent un opportunisme total, ce qu'ils réalisent en gérant les opérateurs via un blackboard. Le concepteur peut adopter cette solution de la souplesse optimale, mais du plus long temps de génération, en modélisant au niveau tâche une gestion de l'opportunisme.

Cette structure d'inférence est *destinée à des explications complexes* i.e. naturelles, adaptées à l'utilisateur et à ses buts, etc. Le concepteur peut donc éventuellement n'en prendre qu'une sous-partie *pour des explications plus simples*.

Enfin, comme il s'agit d'une synthèse, l'**utilité** des divers opérateurs à employer et surtout la **manière de les construire** ne sera **pas détaillée** ci-dessous. On se reportera pour cela aux auteurs cités. *Seule leur mise en oeuvre dans le cadre de KADS et de l'architecture proposée sera discutée.*

Dans cette figure 4, pour des raisons de clarté, des appellations plus précises que «structure-de-données» ont été données aux rôles. Les noms des SC ont aussi été explicités.

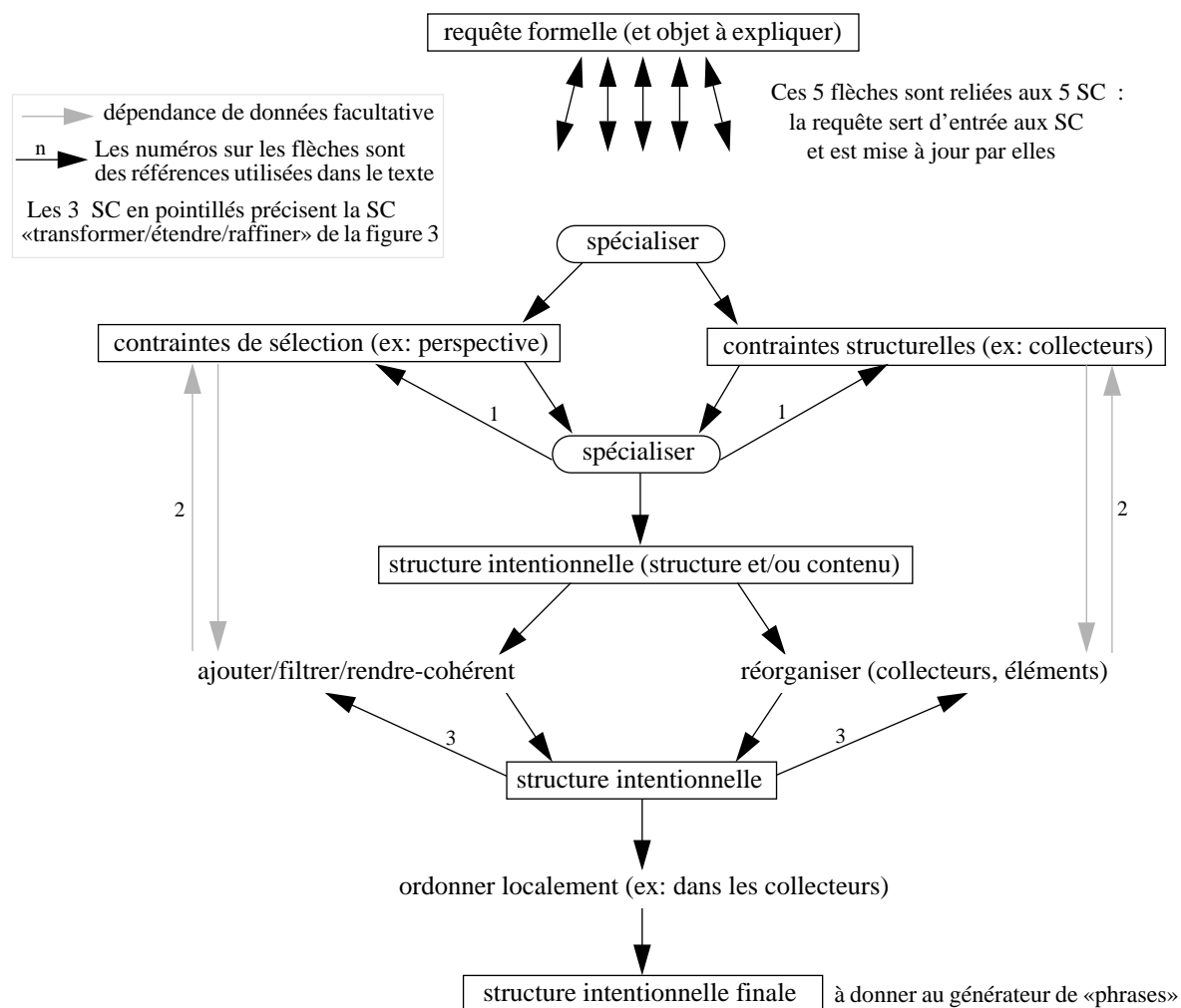


Figure 4: Précisions sur la construction de la structure intentionnelle

En génération d'explication, les opérateurs sont déclenchés pour atteindre un but ou sous-but de discours. Dans la structure d'inférence suivante, les *cing* SC trouvent ces buts dans la requête formelle d'où partent donc *cing* flèches. Ces flèches sont *bi-directionnelles* car chaque SC peut modifier la requête ou surtout la spécialiser pour la SC suivante. Dans la structure d'inférence précédente, cette notion de mise à jour de but n'apparaît pas explicitement car la requête formelle est incluse et propagée dans les «structure-de-données».

Les trois dernières SC en pointillés, ou plus exactement groupes de SC, remplacent le groupe de SC «transformer/étendre/raffiner» de la figure 3.

Les deux SC «spécialiser» remplacent «sélectionner/assembler» : elles sélectionnent des connaissances dans le niveau domaine du ME-Présentation et dans les «connaissances de support pour les explications», les utilisent et les adaptent en fonction du contexte : l'utilisateur, le contenu de la requête, etc. Le terme «spécialiser» est donc plus juste que «sélectionner». Ces SC, ou inférences, fournissent ainsi des *contraintes structurales et informatives et des éléments de réponses*. L'assemblage de ces derniers est implicite compte-tenu des contraintes structurales.

- **La première SC «spécialiser» symbolise la phase de recherche de contraintes structurelles ou informatives**, souvent liées aux données et préalable à la phase de planification. Plus précise que «contraintes structurelles» est l'idée de «collecteurs» («clusters» chez [Lester et Porter 91]). Chaque collecteur rassemble des éléments d'explication nécessaires pour atteindre un but de l'explication. La première inférence peut donc fournir en tant que *plan initial* de l'explication une liste de collecteurs contenant des éléments. Exemple de plan initial : «Motivation», «Introduction», «Mécanismes». Plans et éléments peuvent être ordonnés mais ces derniers le sont rarement à l'issue de cette première phase. Cette prédétermination d'éléments et de contraintes mériterait d'être plus détaillée, mais elle dépend trop du type de la requête et de l'application pour cela. On peut se reporter à [Suthers 91] ou [Lester et Porter 91] pour plus de détails sur la détermination de la «*perspective*».

- **La seconde SC «spécialiser» représente la phase d'expansion de buts (planification)**. Les opérateurs de planification sélectionnent, adaptent et assemblent des éléments de réponse, en tenant compte de *contraintes structurelles et de sélection*. Les **flèches «1»** sur la figure 4 montrent que les contraintes peuvent être mises à jour après chaque application d'opérateur. Ces flèches forment deux boucles, ce qui rappelle que cette seconde SC peut soit modéliser toute la planification, soit une inférence beaucoup plus simple mais appelée plusieurs fois successivement par le niveau tâche.

Les **flèches «2»** grisées montre que le niveau tâche peut revenir à une phase de planification après une phase de correction. La série d'appels aux opérateurs de planification et leur alternance avec les opérateurs de correction sont donc gérés au niveau tâche, selon un mécanisme et des règles que choisira le concepteur : opportunisme total guidé par des heuristiques ([Lemaire et Safar 91]), alternance liées au opérateurs, alternance stricte, etc.

Les flèches «1» et «2» permettent d'utiliser la même structure d'inférence quels que soient les différents séquençements des phases de planification et de correction. Voici quelques justifications de la nécessité de ces séquençements :

1) il doit être possible de lancer une *phase de correction au cours de la planification* pour corriger les erreurs le plus tôt possible (donc après chaque application importante d'un opérateur ou d'un groupe d'opérateur);

2) de façon à être interruptible, la planification doit commencer à «*présenter*» des *explications alors que tout l'arbre de leur génération n'est pas encore développé*;

3) la phase de planification elle-même peut s'effectuer en deux passes, l'une pour déterminer le contenu des explications et l'autre pour structurer ce contenu (mais à notre connaissance, seul [Zukerman 91] opère ainsi).

Il y a donc un très grand nombre de structures de tâches possibles, i.e. de moyens de gérer ce **contrôle** : structures de contrôle classiques, blackboard, etc. Aucune ne peut être jugée meilleure que les autres; celle de l'**opportunisme** total étant probablement la solution la plus complète mais la plus coûteuse en temps d'exécution. Le contrôle doit inclure la gestion des *retours arrière* : exploration d'une autre branche de génération d'explication lorsque celle en cours apparaît inadéquate (grâce à des *heuristiques d'évaluation*). Cette *exploration* peut se faire en «largeur d'abord» (e.g. [Lemaire et Safar 91]) ou en «meilleur d'abord», ce qui est la solution la plus commune.

- **Les opérateurs de correction améliorent le contenu et l'organisation des éléments d'explication qui constituent la structure intentionnelle**. De nombreux exemples de ces opérateurs sont donnés dans les quatre travaux de référence ici synthétisés.

La structure intentionnelle est associée à une *mémoire des buts poursuivis pour sa génération* (e.g. arbre de génération de [Moore et Paris 91]) et suit les *contraintes structurelles* définies plus haut. Par exemple, ces contraintes peuvent être des collecteurs, lesquels peuvent être triés, divisés, ajoutés, etc., tandis que leurs éléments peuvent changer de collecteur, être supprimés, etc.

Dans la figure 4, ces diverses inférences ont été regroupées par type d'action, mais il y a en réalité autant de SC que d'applications d'opérateurs différents. Les deux boucles (**flèches «3»**) permettent de modéliser les divers séquençements des applications de ces

divers types d'opérateurs. C'est au niveau tâche de gérer ces séquencements, éventuellement de manière totalement opportuniste.

Comme le signalent les flèches grisées, ces applications se font sous des contraintes et modifient ces contraintes *pour orienter* l'application future d'autres opérateurs.

- *Lorsque plus un opérateur ne peut s'appliquer sur (la partie à développer de) la structure, ou que le niveau tâche ne le permet plus, commence une phase d'ordonnement local des éléments de cette structure intentionnelle.* Selon [Lester et Porter 91], ce peut être à trois niveaux : «entre intentions, entre propositions ou entre rôles d'une proposition». Les opérateurs d'ordre ou d'ordonnement peuvent être selon [Suthers 91] des méthodes de : *traversée de graphes* (i.e. suivi de chaîne temporelle, causale, etc.), *d'exploitation de relations pédagogiques* (e.g. dans l'explication d'un concept, les définitions ou les généralisations sont en premier tandis que les illustrations ou les particularisations sont en dernier), *de dérivation de focus* (i.e. comment placer tel élément compte-tenu que tel autre est plus important), etc.
- La (*partie de*) structure intentionnelle ainsi développée peut alors être donnée au générateur de phrase (cf figure 3).

En résumé, pour modéliser l'expertise de présentation, le **niveau domaine** du ME-Communication doit non seulement contenir les différents concepts et relations relatifs aux langages (lexique, syntaxe, etc.) et aux «connaissances de support pour les explications» mais aussi les opérateurs qui exploitent ces connaissances. Les niveaux **inférence, tâche et stratégie** contrôlent les applications de ces opérateurs. Pour plus de détails sur ces derniers, on se référera aux auteurs cités ci-dessus; une synthèse des techniques actuelles de génération d'explication peut se trouver dans [Suthers 91], [Lemaire 92] ou [Martin 92]).

Nous n'avons pas mentionné dans les spécifications sur les divers contrôles possibles, ci-dessus, le niveau stratégie. En fait, pour reprendre sa définition, celui-ci doit adapter dynamiquement, i.e. en fonction du contexte ou des échecs, des structures de tâches - ces dernières étant les stratégies préfixées du niveau tâche. Cependant, dans le cas du ME-Communication, l'essentiel de ce travail est déjà effectué soit par le niveau tâche qui opère une gestion dynamique sur les opérateurs, soit par le ME-Coopération qui gère les problèmes de haut niveau de la communication : changement de l'objet de la transaction, des objectifs à atteindre, du type de dialogue, etc.

4. Conclusion

Pour aider le concepteur à modéliser l'expertise de coopération et d'explication nécessaire à la construction d'un SBC explicatif, nous avons transformé le classique Modèle de Coopération de KADS en deux nouveaux modèles d'expertise et spécifié les connaissances qu'ils pouvaient inclure d'une part et exploiter d'autre part.

De plus, nous avons donné des modèles d'interprétations pour les principales tâches du ME-Communication. Ce sont des exemples encore assez généraux, qui mériteraient donc d'être précisés par plusieurs modèles d'interprétations plus spécifiques. Les recherches sur l'interprétation de données, sur la génération d'explication et sur la gestion de dialogue, ainsi exprimées au knowledge level [Newel 82] pourraient plus aisément être exploitées pour des implémentations.

Même si l'expertise de coopération est rarement importante pour des SBC classiques, le ME-Coopération offre pour la spécifier et stocker, un cadre propre, modulaire et distinct de l'expertise de résolution de problèmes.

Dans [Martin 92] nous avons complété cette étude en donnant une méthode pour acquérir de manière assez complète, les «connaissances de support pour les explications». Cette méthode est basée sur les éléments des modèles d'interprétation KADS : elle recense à un niveau assez général, les «connaissances de support» relatives à chacun de ces types d'éléments. Cette méthode peut ainsi être utilisée avec n'importe quel modèle d'interprétation et donc lors de l'extraction de n'importe quel type d'expertise.

5. Remerciements

Nous tenons à remercier Rose Dieng, Alain Giboin, Olivier Corby et Sofiane Labidi pour leurs commentaires sur les versions précédentes de cet article.

6. Références

- > AAAI 91: *Proceedings of the AAAI'91 Workshop on Comparative Analysis of Explanation Planning Architectures*, Anaheim, California, July 1991.
- > EXPL 92: *Actes des deuxièmes journées «Explication» du PRC-GDR IA du CNRS*. Sophia-Antipolis, 17-19 Juin 1992.
- [Baumewerd et al. 91] A. Baumewerd-Ahlmann, P. Jaschek, J. Kalinski, and H. Lehmkuhl. Embedding Explanations into Model-Based Knowledge Engineering - Improved Decision Support in Environmental Impact Assessment. In *Proceedings of the 1st Int. Conference on Knowledge Modeling and Expertise Transfer*, pp 269–284, Sophia-Antipolis, France, April 1991.
- [Breuker (ed) 87] J. Breuker (ed). Model Driven Knowledge Acquisition: Interpretation Models. Deliverable A1, Esprit Project 1098 Memo 87, VF project Knowledge Acquisition in formal domains.
- [Cawsey 91] A. Cawsey. Generating Interactive Explanations. In AAAI 91.
- [Chandrasekaran et al. 89] B. Chandrasekaran, M. C. Tanner et J. R. Josephson. Explaining Control Strategies in Problem Solving. In *IEEE Expert (1989)*, pp 9-24.
- [de Greef et Breuker 92] P. de Greef, J. Breuker. Analysing system-user cooperation in KADS. In *Knowledge Acquisition (1992) 4*, pp 89-108.
- [Lemaire et Safar 91] B. Lemaire, B. Safar. Some Necessary Features for Explanation Planning Architectures: Study and Proposal. In AAAI 91 pp 15-26.
- [Lemaire 92] B. Lemaire. Aspects constructifs de la production d'une explication : l'architecture ESMERALDA. In EXPL 92 pp 165-177.
- [Lester et Porter 91] J. C. Lester, B. W. Porter. An Architecture for Planning Multi-Paragraph Pedagogical Explanation. In AAAI 91 pp 27-41.
- [Martin 92] P. Martin. La méthodologie d'acquisition des connaissances KADS et les explications. *Rapport de stage de D.E.A. informatique, Nice-Sophia Antipolis*, projet ACACIA, INRIA Sophia Antipolis.
- [Moore et Paris 91] J. D. Moore, C. L. Paris. The EES Explanation facility: its tasks and its architecture. In AAAI 91 pp 65-79.
- [Newell 82] A. Newell. The Knowledge Level. In *Artificial Intelligence 18 (1982)*, pp 87-127.
- [Paris 92] C. L. Paris. Systèmes Experts Explicatifs. In EXPL 92 pp 3-23.
- [Safar et al. 92] B. Safar, P. Berthault, J. Sylvestre. Place des explications dans la conception d'une interface intelligente entre une base de données et un usager. In EXPL 92 pp 221-231.
- [Suthers 91] D. D. Suthers. Task-Appropriate Hybrid Architecture for Explanation. In AAAI 91 pp 80-94.
- [Sprenger 92] M. Sprenger. Explanation strategies for KADS-based expert systems (revised version). In rapport GMD : DIAMOD 10, juillet 1992.
- [Wielinga et al. 92] B. Wielinga, G. Schreiber, and J. Breuker. KADS: a modelling approach to knowledge engineering. In *Knowledge Acquisition (1992) 4*, pp 136-145
- [Zukerman 91] I. Zukerman. Failure Prevention: an Alternative Approach to Content Planning. In AAAI 91 pp 80-94.