

# Modèle d'utilisation et modèles de tâches pour l'assistance à l'utilisateur basée sur l'expérience : le cas d'un système d'information audiovisuelle

## Use Model and Tasks Models for Experienced-based Users' Helping: Application in an Audiovisual Information System\*

Yannick Prié<sup>1,2</sup>

Alain Mille<sup>2</sup>

Jean-Marie Pinon<sup>1</sup>

<sup>1</sup> Laboratoire d'Ingénierie des Systèmes d'Information — LISI

Bât 404. INSA, 20 avenue Albert Einstein F-69621 VILLEURBANNE

<sup>2</sup> Equipe Raisonnement à Partir de Cas, LISA

CPE-Lyon, 43 Boulevard du 11 Novembre 1918 F-69616 VILLEURBANNE

Yannick.Prie@{insa-lyon.fr,cpe.fr} / am@cpe.fr / pinon@lisi.insa-lyon.fr

### Résumé

Nous présentons dans cet article un modèle original de description de systèmes informatiques en modèle d'utilisation et modèles de tâches, qui permettent de décrire l'activité concrète d'un utilisateur en cas d'utilisation, base de l'apprentissage pour l'aide à l'utilisateur basée sur l'expérience. Comme illustration des ces principes dans un système d'information audiovisuel, nous présentons tout d'abord le modèle d'utilisation de SESAME, les Strates-Interconnectées par les Annotations, avec notamment la notion originale de graphes potentiels contextuels. Un modèle de la tâche centrale de description de SESAME, la tâche d'annotation, est ensuite décrit. Une discussion sur les possibilités d'apprentissage dans les cadres mis en place vient conclure cet article.

**Mots-Clef** Audiovisuel, modélisation de connaissances, modèle de tâches, modèle d'utilisation, cas d'utilisation, expérience, description de contextes.

### Abstract

In this article we present an original model for describing computing systems with use and tasks models, that allow to describe the user's real activity with use cases. These use cases are a base for learning from experience and for helping users. As an illustration for these principles in an audiovisual information system, we first present the use model of SESAME: the Annotation-Interconnected Strata, introducing the notion of potential contextual graphs. We then present a model of the annotation task, which is central to SESAME. The last part of this article is devoted to discussion about learning possibilities in the frames we have set-up

**Keywords** Audiovisual, knowledge modeling, task models, use model, use cases, experience, context description.

---

\*Ce travail s'inscrit dans le cadre du projet Sesame financé par le CNET, consultation thématique 96-ME-17. This work is part of the Sesame project granted by the French Center for Telecommunication Research under contract 96-ME-17.

# 1 Introduction

Deux objectifs complémentaires ont guidé la rédaction de cet article: le premier objectif est de présenter une nouvelle façon d'étudier l'assistance informatique à la tâche d'un utilisateur exploitant les services d'un système informatique, le second est de montrer comment notre proposition de modèle de représentation de séquences audiovisuelles (Strates-IA [15]) s'adapte et se justifie pleinement selon cette nouvelle façon de voir les choses.

Un système informatique<sup>1</sup> *idéal* serait en effet un système qui répondrait au mieux aux questions que lui poserait un utilisateur pour lui faciliter sa *tâche*. Un tel système ne serait donc plus une composition d'applications informatiques capables de répondre à un nombre limité de questions prédéfinies (par exemple dans un système comptable), mais une collection de services offerts pour répondre à des *classes de questions* correspondant à des *classes de tâches*. La notion d'exploration du système informatique s'est par ailleurs imposée comme moyen privilégié de l'utilisateur dans sa quête d'assistance à sa propre tâche. L'indexation de séquences audiovisuelles pour faciliter leur recherche, leur analyse, leur réutilisation, etc. s'inscrit naturellement dans cette tendance et le modèle des Strates-IA [15], [13] a été conçu pour faciliter ce type de services, mais également pour servir de support à l'assistance à différentes tâches, y compris les tâches qui ne seraient pas prédéfinies mais induites par des *cas d'utilisation* concrets. Afin de formaliser la manière dont le modèle des Strates-IA permet d'atteindre l'objectif de favoriser l'assistance à la tâche, en particulier par *capitalisation de connaissances*, nous structurons dans la première partie les connaissances qui peuvent être mobilisées dans le cadre d'un outil informatique pour assister la tâche de l'utilisateur, avant de situer et d'illustrer cette approche sur un exemple. Dans une seconde partie, nous utilisons la structuration précédemment définie afin de présenter d'abord le modèle d'utilisation des Strates-IA, puis la manière dont ce modèle de base est mobilisé autour de la tâche principale d'annotation. La dernière partie est consacrée à une discussion mettant en perspectives les potentiels d'apprentissage par introspection, sur la base de l'expérience.

## 2 Modéliser utilisation et tâches pour aider l'utilisateur

### 2.1 Définitions

Nous présentons ci-après quelques définitions nécessaires à la formalisation d'une assistance basée sur l'expérience à la tâche d'un utilisateur d'un outil informatique.

**Outil informatique** Un outil informatique est un logiciel exploité pour une classe de tâches. Il est caractérisé par les *objets de base* qu'il fournit à *l'utilisateur*. Ces objets de base peuvent être décrits par un *modèle d'utilisation*.

**Utilisateur** Un utilisateur est un être humain exploitant les possibilités offertes par un outil informatique dans le cadre d'une tâche. La tâche considérée peut être une sous-tâche d'une tâche beaucoup plus générale. Elle correspond à un sous-but que l'utilisation de l'outil informatique permet d'atteindre plus facilement. Une tâche peut être décrite par un *modèle de tâche*.

**Modèle d'utilisation** Un modèle d'utilisation est intrinsèquement lié à l'outil exploité. Il décrit les objets de base de l'outil informatique (propriétés, signatures des méthodes) ainsi que les relations qu'ils entretiennent. Un modèle d'utilisation minimal existe par définition (l'utilisation d'un outil sans mode d'emploi explicite minimal est exclue), et on considère qu'il n'existe qu'un seul modèle d'utilisation pour un outil informatique, celui-ci étant défini au moment de la conception de cet outil.

---

1. Système informatique: expression très large recouvrant l'ensemble des matériels et logiciels mobilisés pour fournir des services à un utilisateur.

**Modèle de tâche** Un outil peut être associé explicitement à une tâche prédéfinie, c'est à dire à la réalisation par l'utilisateur d'un but prédéfini. Cette association nécessite un modèle de tâche décrivant une façon particulière d'utiliser l'outil dans un but particulier. A cette occasion, des concepts nouveaux, construits à partir de ce qu'il est possible d'exprimer avec le modèle d'utilisation, peuvent être décrits et permettre d'aider *spécifiquement* l'accomplissement de la tâche pour laquelle le modèle a été bâti. Le modèle de tâche décrit les nouveaux concepts et les relations qu'ils entretiennent. Il peut exister de très nombreux modèles de tâche pour un outil particulier, et ceux-ci peuvent être prédéfinis au moment de la conception mais aussi au moment de l'exploitation de l'outil informatique<sup>2</sup>.

**Notion d'assistant informatique** Un assistant informatique est un logiciel qui assiste l'exploitation d'un outil informatique dans le cadre d'une tâche particulière. Il exploite le modèle de tâche concerné. Un assistant informatique *de base* assiste l'utilisation d'un outil selon son modèle d'utilisation. Un assistant informatique couvre tout ou partie d'un modèle d'utilisation ou de tâches.

**Cas d'utilisation** Un cas d'utilisation est une structure décrivant une utilisation particulière de l'outil informatique pour un but particulier. Un *but* s'exprime par la *requête*<sup>3</sup> de l'utilisateur. Complété par la description de la *réponse* à la requête, un cas d'utilisation est une instance concrète d'une utilisation d'un système. Un cas d'utilisation sera d'autant plus facile à (ré-)utiliser que requête et réponse s'expriment dans un même langage et, d'une manière générale, font référence aux objets décrits dans les modèles d'utilisation et/ou de tâche. Par définition, la notion même de cas d'utilisation fait partie du modèle d'utilisation.

**Explications d'un cas d'utilisation** Les éléments d'un cas d'utilisation sont des instances d'objets décrits dans le modèle d'utilisation ou/et dans un modèle de tâche. Les différents éléments du cas d'utilisation profitent des relations existantes (inter- ou intra-modèles) entre les objets de ces modèles, qui fournissent des *explications* à leur co-occurrence. Un cas d'utilisation peut donc toujours s'expliquer par le modèle d'utilisation, et, quand il existe, par le modèle de la tâche courante. L'utilisateur peut également considérer que dans un contexte précis (non répertorié par une tâche particulière), deux éléments du cas d'utilisation se font référence. Il a alors la possibilité de mettre en relation explicite ces éléments du cas d'utilisation, et leur co-occurrence s'explique par cette relation *contextuelle*. Toutes ces relations supplémentaires entre éléments du cas d'utilisation sont autant d'explications supplémentaires du cas, liées à la tâche : on dit qu'elles rationalisent le cas. Remarquons que comme les explications doivent pouvoir être rendues explicites à l'utilisateur, la notion d'explication fait également de droit partie du modèle d'utilisation.

**Capitalisation de connaissances** La capitalisation de connaissances, appliquée à une organisation, consiste à mettre en place les mécanismes qui permettent ou/et garantissent que les connaissances d'une entreprise soient capitalisées, c'est à dire rendues disponibles pour résoudre de nouveaux problèmes. L'outil informatique est bien entendu souvent la cheville ouvrière de ce type d'organisation. Dans le cadre de cet article, nous nous intéressons à la capitalisation de connaissances qui peut être faite par un système informatique *per se* dans le cadre d'un outil particulier. En d'autres termes, il s'agit des connaissances que l'outil peut capitaliser pour faciliter les tâches de son ou ses utilisateurs. La capitalisation de connaissances est un processus intégré à l'outil, en tant que mécanisme d'enrichissement de ses modèles de tâche.

Les définitions que nous venons de présenter ont un cadre plus large que l'étude des Strates-IA, et nous avons utilisé la même démarche dans le cadre d'autres projets de recherche ([6], [11] par exemple), mais c'est dans le cadre de cette étude que nous l'avons formalisée comme objectif en soi.

---

2. Il serait facile de montrer que le modèle d'utilisation d'un outil informatique correspond en fait à un modèle de tâche générique partagé par toutes les tâches s'appuyant sur l'outil. La sémantique de base de ces deux types de modèle est donc fondamentalement la même mais les connaissances explicatives d'un artefact réalisé avec cet outil sont très différentes selon que l'on dispose ou non d'un modèle de tâche bien précis.

3. La notion de requête est très générale à ce niveau, il s'agit de toute information fournie par l'utilisateur pour obtenir un service de l'outil informatique.

La figure 1 illustre schématiquement les relations existantes entre modèle d'utilisation et modèles de tâche.

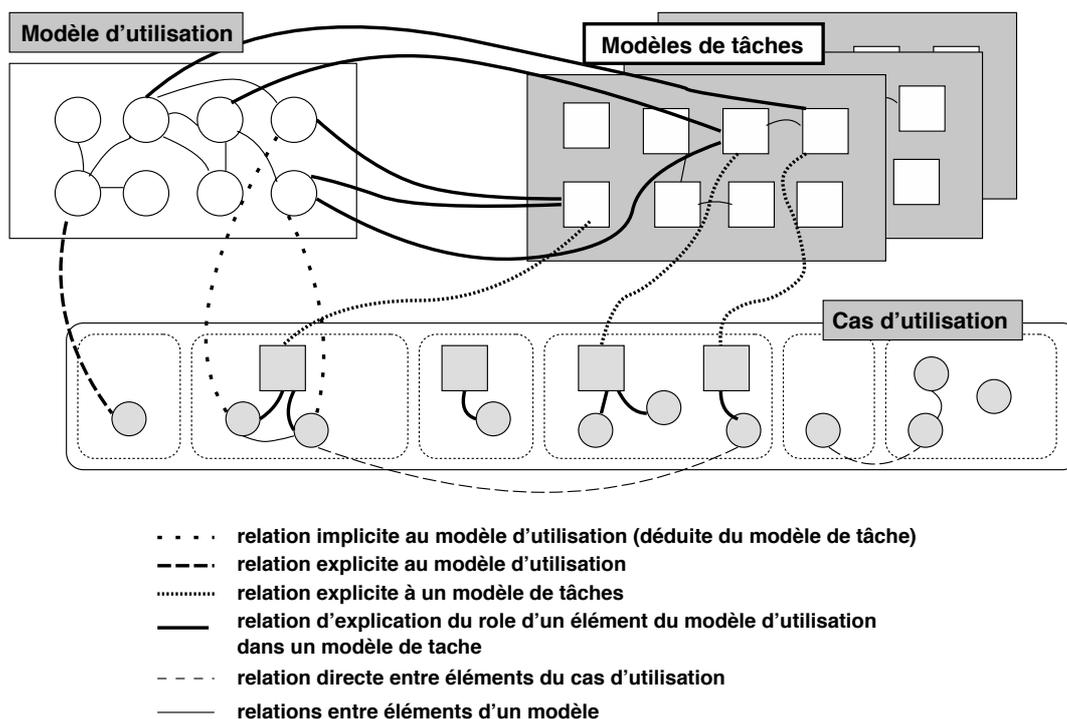


FIG. 1 – *Modèle d'utilisation, modèles de tâches, cas d'utilisation et explications*

Les éléments d'un modèle d'utilisation s'expliquent par les relations qu'ils entretiennent les uns avec les autres. Les modèles de tâches s'expliquent par les éléments du modèle d'utilisation qu'ils utilisent et par les relations qu'ils entretiennent les uns avec les autres. Les éléments d'un cas d'utilisation s'expliquent par les éléments du modèle d'utilisation qu'ils utilisent et par les éléments du modèle de tâche qu'ils respectent. Les relations directes entre éléments du cas d'utilisation n'ont *a priori* de sens que dans ce cas d'utilisation.

## 2.2 Un exemple dans le domaine de la bureautique

Les définitions ci-dessus peuvent être illustrées par l'exemple d'un logiciel de traitement de texte qui est l'outil informatique sans doute le plus courant :

- **Modèle d'utilisation** : les objets de base principaux sont le caractère, le mot, la ligne, le paragraphe, le document, le modèle de document, les éléments du langage de *programmation*, *etc.*
- **Modèle de tâche** : un nombre croissant de modèles correspondant à une tâche particulière sont proposés comme (structurer une) lettre, (organiser un) rapport, *etc.* L'utilisateur peut créer ses propres modèles de tâche en composant différents objets du modèle d'utilisation et en ajoutant des méthodes écrites selon la syntaxe disponible dans le modèle d'utilisation.
- **Assistants** : un assistant générique (permettant de comprendre le modèle d'utilisation) permet d'accéder à des assistants spécifiques (fournissant souvent une aide interactive pour exploiter au mieux un modèle de tâche particulier). Ces assistants exploitent à l'heure actuelle exclusivement les modèles génériques des tâches, à l'exclusion des modèles qui pourraient être créés par l'utilisateur.

Les notions de *capitalisation d'expérience*, *cas d'utilisation*, *explications* n'ont pas d'équivalent dans les outils bureautiques actuels, où les modèles n'ont malheureusement pas (encore) le statut de connaissances.

### 3 Les Strates-IA : modèle d'utilisation

Le modèle des Strates-IA a été mis en place dans le cadre du projet SESAME<sup>4</sup> afin d'autoriser les descriptions les plus riches possibles de documents audiovisuels en vue de leur utilisation dans le cadre d'un système d'information audiovisuel. Cette approche s'inscrit dans une pratique de réécriture des documents audiovisuels visant à rendre possible leur accès, la navigation, l'édition, *etc. cf.* [14, 3]. Dans cette partie, nous en présentons rapidement le modèle d'utilisation. Les éléments de base du modèle — illustrés figure 2 — sont les *unités audiovisuelles* (UAV), qui sont des représentants de *strates*<sup>5</sup>; les *éléments d'annotations* (EA), qui sont des objets caractérisés par un *terme* (le nom de l'EA) et des attributs/valeurs. On considère qu'un EA correspond à l'expression d'une caractéristique audiovisuelle liée à un *objet d'intérêt* repéré en visualisant le flux (par exemple un personnage, une sensation, un taux de couleur ou une scène), automatiquement ou manuellement. Un EA est l'*inscription dans le flux* d'un *élément d'annotation abstrait* (EAA) lequel possède également des attributs/valeurs. Un élément d'annotation annote une unité audiovisuelle (relation  $R_a$ ), et est en relation de décontextualisation  $R_d$  avec l'élément d'annotation abstrait dont il est issu. Ces derniers sont organisés dans une *base de connaissances* BC, en un arbre de spécialisation/abstraction (relation  $R_s$ ), mais d'autres relations telles que celles qu'on peut trouver dans un thésaurus sont possibles (par exemple «voir aussi»  $R_{va}$ ). Enfin, on crée autant de *dimensions d'analyse* que nécessaire afin d'exprimer les possibilités d'analyse d'un flux audiovisuel. Une dimension d'analyse est — en simplifiant — un ensemble d'éléments d'annotation abstraits dans lesquels on pourra puiser pour annoter un flux, par exemple une dimension d'analyse *Hommes Politiques* pourrait regrouper les EAA  $\langle EAA : Chirac \rangle$ ,  $\langle EAA : Mandela \rangle$  et  $\langle EAA : Jospin \rangle$ .

Deux éléments d'annotation quelconques peuvent être mis en relation à l'aide de la *relation élémentaire*  $R_e$ , qui n'exprime rien d'autre que le fait qu'elle mette en relation ces deux EA. Dans le cadre d'une tâche d'annotation «contrôlée», pour laquelle toute mise en relation de deux EA doit avoir un sens, on utilise deux relations élémentaires ainsi qu'un EA (appelé élément d'annotation de relation), lequel exprime le sens de la relation, par exemple on pourra exprimer le fait «qu'au cours du plan, la caméra suit Nelson Mandela» par  $\langle Plan \rangle R_e \langle FocusVideo \rangle R_e \langle Mandela \rangle$ . Les attributs des EA peuvent être de tous types, par exemple un texte illustrant un discours d'un homme politique, ou une image considérée comme représentative d'un plan). Pour plus de détails sur les tenants et les aboutissants de l'approche<sup>6</sup>, le lecteur est invité à se référer à [14].

Les éléments du modèle d'utilisation que nous présentons dans la suite mettent en jeu les éléments de base que nous venons de définir, et s'appuient sur la constatation que l'ensemble d'un système Strates-IA (*i.e.* les UAV, les EA, les EAA, leurs attributs et leurs relations) représente un unique graphe orienté étiqueté  $G_g$ . Les étiquettes des sommets sont composées de couples attributs-valeurs, les deux attributs obligatoires étant *Type* (qui prend ses valeurs dans  $\{EA, EAA, UAV\}$ ), et *Terme* le terme caractérisant l'élément (un numéro unique pour une UAV, l'expression d'une caractéristique audiovisuelle pour les EA et EAA). Les étiquettes des arcs ont un seul attribut *Type* qui prend ses valeurs dans l'ensemble des relations disponibles ( $R_a, R_e, etc.$ ), aux contraintes de cohérence près. Un tel graphe est dit *totalelement étiqueté*.

4. Système d'exploration de Séquences Audiovisuelles enrichies par L'Expérience.

5. Morceaux de documents audiovisuels caractérisés par un instant de début et de fin.

6. Elle permet de prendre en compte la dynamique du média, autorise une finesse d'annotation variable, gère les annotations structurelles de la même manière que les autres, permet une mise en relations homogène de tous les documents de la base, *etc.*

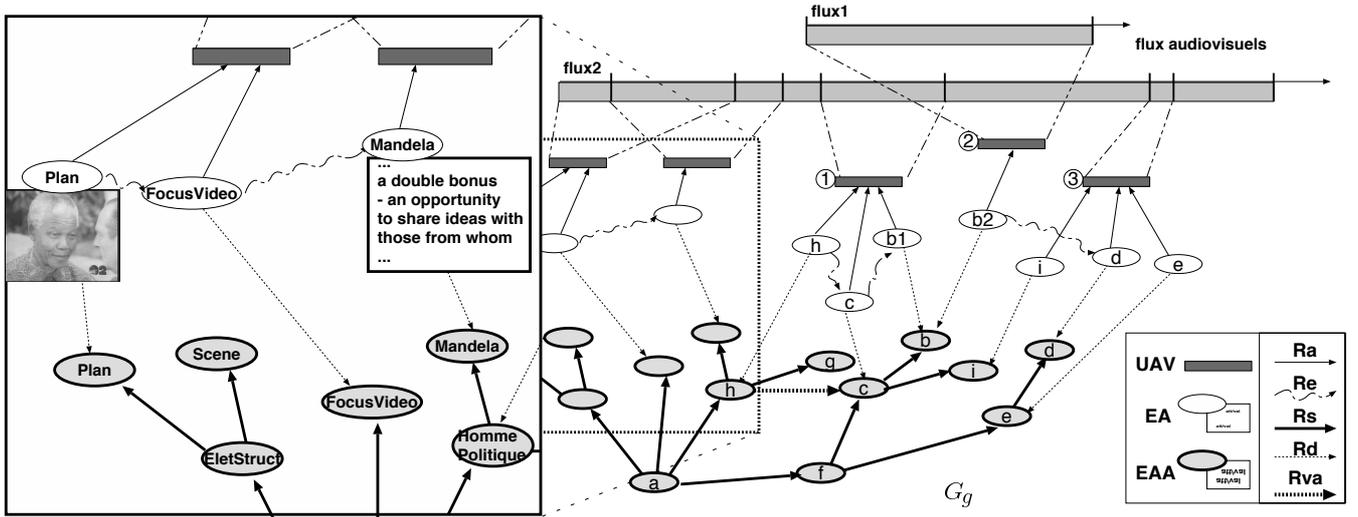


FIG. 2 – *Eléments de base des Strates-IA*

Comme ce graphe est connexe, on considère que pour tout couple d'éléments  $(x, y) \in G_g$  il existe au moins un chemin dans le graphe permettant de mettre  $x$  en relation avec  $y$ . On dit alors que  $y$  fait partie du *contexte* de  $x$  (par exemple figure 2, l'EA d fait partie du contexte de l'UAV 2 du fait d'un chemin passant par b2). Nous définissons la notion de contexte d'un élément  $x$  de  $G_g$  comme l'ensemble des éléments de  $G_g$  qui peuvent être mis en relation avec lui. Afin de contrôler le contexte, c'est à dire les chemins de mise en relation dans le graphe, nous mettons en place la notion de *graphe potentiel* permettant de décrire un chemin abstraitement. Un graphe potentiel  $g_p$  est alors un graphe étiqueté répondant aux mêmes contraintes que  $G_g$ , l'attribut terme pouvant prendre une valeur supplémentaire  $*$ . Un tel graphe est dit *partiellement étiqueté*.

On dit qu'un sous-graphe  $g$  de  $G_g$  est une *instance* d'un graphe potentiel  $g_p$  dans  $G_g$  si il existe une fonction bi-univoque  $f$  de l'ensemble  $S_p$  des sommets de  $g_p$  dans l'ensemble  $S_g$  des sommets de  $g$  avec:  $(x, y)$  arc de  $g_p \Leftrightarrow (f(x), f(y))$  arc de  $g$  (critère de cohérence syntaxique concernant la structure du graphe);  $\forall (x, y)$  arc de  $g_p$ ,  $Type((x, y)) = Type((f(x), f(y)))$ ; et  $\forall x$  sommet de  $g_p$ ,  $Type(x) = Type(f(x))$  et  $Terme(x) = Terme(y)$  si  $Terme(x) \neq *$  (critères de cohérence sémantique). La figure 3 illustre cette notion: le sous-graphe  $g_{i1}$  est instance de  $g_p$  dans le graphe  $G_g$  de la figure 2. L'*instanciation*<sup>7</sup> d'un graphe potentiel dans le graphe général du système donne un certain nombre d'instances (par exemple figure 3,  $g_p$  a trois instances  $g_{i1}$ ,  $g_{i2}$  et  $g_{i3}$  dans  $G_g$ ).

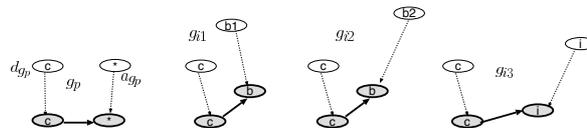


FIG. 3 – *Un graphe potentiel et ses instances dans  $G_g$*

Parmi l'ensemble des graphes potentiels possibles<sup>8</sup>, nous ne nous intéressons ici qu'à des graphes potentiels contextuels exprimant des possibilités de chemins contextuels dans le graphe, caractérisés par deux sommets particuliers, un EA  $d_{g_p}$  point de départ du chemin, et un autre EA dit point d'arrivée  $a_{g_p}$ . Le graphe  $g_p$  figure 3 est un graphe de cette sorte. L'utilisation d'un tel graphe pour

7. Nous n'évoquons pas ici les mécanismes d'instanciation qui se basent sur une heuristique raisonnable, selon laquelle quel que soit le graphe potentiel  $g_p$  considéré, on dispose d'au moins une correspondance connue et unique entre deux sommets de  $g_p$  et  $G_g$  à partir duquel on peut rechercher l'isomorphisme de sous-graphe.

8. Relations contextuelles potentielles, filtres de désignation, requêtes étoilées, possibilités de manipulation, de jonction de ces graphes, *etc.* non présentés ici.

mettre en évidence le contexte d'un EA  $e$  du graphe  $G_g$  consiste à identifier l'EA de départ de  $g_p$  avec l'EA de  $G_g$  considéré, et à instancier  $g_p$ . L'ensemble des points d'arrivée des instances de  $g_p$  dans  $G_g$  représente le contexte de  $e$  suivant  $g_p$ . Ainsi, en utilisant  $g_p$  de la figure 3 dans  $G_g$  de la figure 2, on peut repérer trois EA (b1, b2, i) dans le contexte de l'EA  $c$ .

Terminons la présentation du modèle d'utilisation (partiel) des Strates-IA avec la notion de *valence* d'un EA, qui est stockée comme attribut de l'élément d'annotation abstrait dont il est inscription dans le flux. La sémantique d'une valence peut se comprendre comme *une possibilité explicite, pour l'EA concerné, de relation avec d'autres EA atteignables par le graphe potentiel contextuel associé à la valence*. Une valence est donc une connaissance constituée d'un EAA  $eaar_r$  et d'un graphe potentiel contextuel  $g_p$  à point de départ EA, et exprime le fait que dans le contexte d'un EA  $e$  caractérisé par  $g_p$  instancié dans  $G_g$  à partir de  $e$ , on trouve généralement un EA  $e'$  en relation avec  $e$  par la médiation d'un EA  $e_r$  inscription dans le flux de  $eaar_r$  et de deux relations élémentaires  $R_e$ . Par exemple, l'EAA  $\langle EAA : Conduire \rangle$  pourrait avoir un attribut **valence** constitué de l'EAA  $\langle EAA : OBJ \rangle$  et d'un graphe potentiel contextuel permettant de rechercher dans un contexte de co-occurrence temporelle stricte (même UAV) un EA extrait d'un EAA qui soit une spécialisation de l'EAA  $\langle EAA : Vehicule \rangle$ , par exemple  $\langle Formule1 \rangle$ , et de le mettre en relation avec  $\langle Conduire \rangle$  par  $\langle Conduire \rangle R_e \langle OBJ \rangle R_e \langle Formule1 \rangle$  (cf. figure 4).

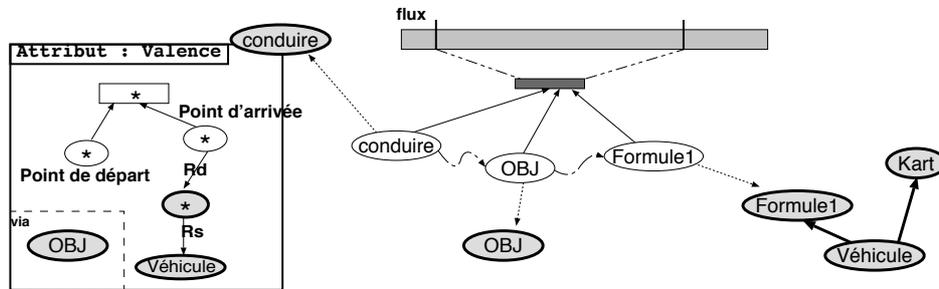


FIG. 4 – Illustration d'un attribut valence de  $\langle EAA : Conduire \rangle$

Synthétisons en quelques lignes les éléments présentés du modèle d'utilisation : les *éléments d'annotation* sont extraits d'*éléments d'annotation abstraits* et annotent les *unités audiovisuelles*. Les EAA sont organisés dans une *base de connaissances*, et peuvent être regroupés en *dimensions d'analyse*. Des *graphes potentiels*, en s'instanciant, permettent de définir des *contextes* d'éléments du *graphe général*. Les *valences* expriment des possibilités de relation entre EA.

## 4 Modèle de la tâche de description

C'est dans le cadre de l'ingénierie des connaissances que la notion de tâche a été proposée (voir [16]) et a même fait l'objet de tentative d'ontologie de tâches génériques, comme par exemple dans [5]. Cette approche était alors totalement liée au développement de systèmes à base de connaissances. Nous considérons les assistants à l'exploitation des outils informatiques comme des systèmes à base de connaissance, devant s'appuyer sur les modèles de connaissances appropriés aux tâches de l'utilisateur. L'outil, même s'il ne prend pas le statut de système à base de connaissance pour autant, doit permettre la manipulation explicite des connaissances qui régissent son fonctionnement, de façon à rendre la notion d'« assistant intelligent » possible. A l'instar de [9], nous considérons qu'il s'agit de la seule façon de rendre de véritables services « personnalisés » aux utilisateurs.

Pour illustrer comment le modèle d'utilisation des Strates-IA permet de servir de support aux tâches d'un utilisateur de SESAME (pour une description fonctionnelle de SESAME voir [13]), nous choisissons la tâche générique d'*annotation de document audiovisuel* qui est à la base d'autres tâches comme l'indexation, la recherche d'une séquence, l'analyse d'un document, *etc.* comme montré figure 5.

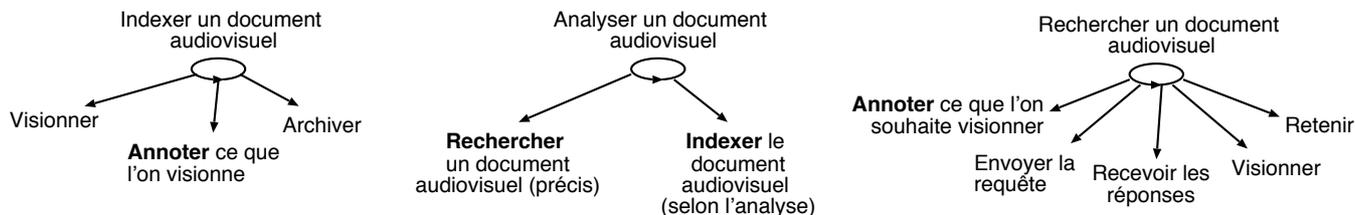


FIG. 5 – *Indexer*: annoter les UAV d'un document — *Rechercher*: annoter pour élaborer les requêtes à la base — *Analyser*: annoter selon son propre point de vue (sur-annoter)

Pour présenter les tâches, nous adoptons un formalisme de diagramme de décomposition permettant de montrer comment le modèle d'utilisation est exploité par les modèles de tâches. Présenter les diagrammes de description de ces tâches serait également intéressant pour mettre en évidence les connaissances mobilisées par chacune des sous-tâches, mais cette description sortirait du cadre que nous nous sommes fixés dans cette présentation (pour une description complète des possibilités de ce formalisme, voir par exemple [10]).

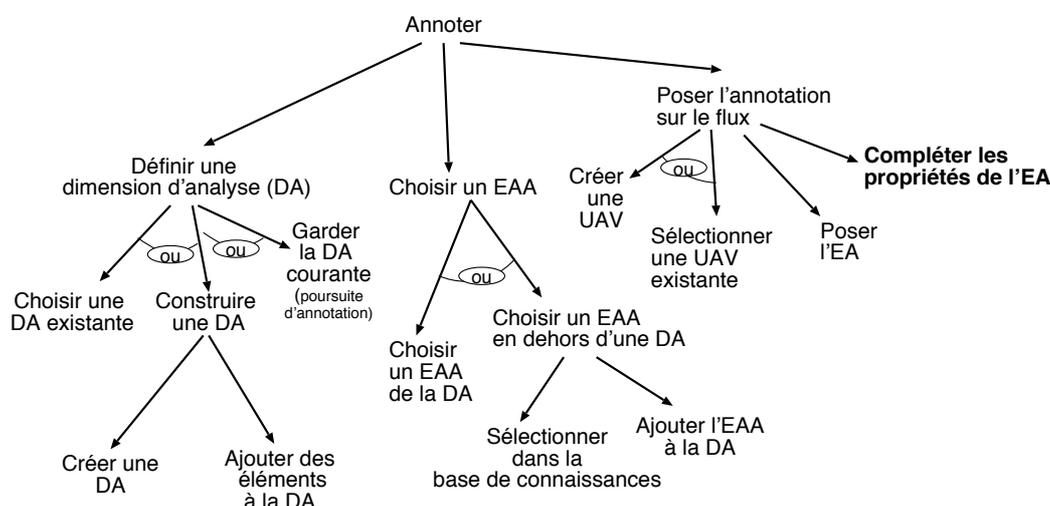


FIG. 6 – *Annoter*: la tâche de base dans le cadre de *SESAME*

Les modèles de tâches de *SESAME* font tous appel à la tâche générique **Annoter** (décrite figure 6). Il s'agit d'un choix délibéré favorisant le même processus de façon à favoriser au maximum la réutilisation des cas d'utilisation qui respecteraient ces modèles de tâches. La tâche **Annoter** se fonde sur le choix d'un *objet d'intérêt* par l'utilisateur. Cet objet d'intérêt n'a pas d'existence formelle dans le modèle d'utilisation, mais il sera caractérisé comme EAA dans une dimension d'analyse que l'utilisateur exploite pour guider son activité d'annotation. D'une manière plus ou moins directe, cette dimension d'analyse oriente le choix d'un élément d'annotation à poser sur le flux audiovisuel. Un cas d'utilisation pourrait donc avoir comme point de départ un EAA choisi dans une DA particulière, et comme trace de résolution l'ensemble des valeurs d'attributs et de relations élémentaires que l'utilisateur a choisies à partir de l'objectif initial. Il est facile de voir qu'un cas d'utilisation complexe peut être composé de sous-cas d'utilisation correspondant à des sous-buts, c'est à dire à des EAA posés dans le cadre d'un objectif plus général. Les tâches **compléter les propriétés d'un EA** et **explorer contextuellement un EA pour trouver un EA-cible** (de mise en relation) sont décrites figures 7 et 8.

La sous-tâche **Compléter les propriétés d'un EA** consiste à compléter l'élément d'annotation posé sur le flux par des attributs le précisant. Les attributs locaux sont intrinsèques à l'annotation elle-même, il s'agit typiquement d'informations numériques (issues d'un traitement d'image par exemple)

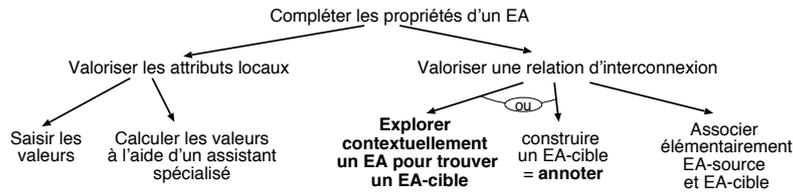


FIG. 7 – Compléter les propriétés d'un EA : une sous-tâche spécialisée par des assistants spécifiques

ou symboliques (issues d'une interprétation contextuelle, un qualificatif d'importance par exemple). Ces attributs seront par exemple exploités par une mesure de similarité entre UAV lors d'une exploration. Plus importantes et originales sont les relations d'interconnexion : il s'agit de connecter d'autres éléments d'annotation avec l'élément d'annotation qui vient d'être posé. Trois possibilités sont ouvertes : associer directement un EA existant (cette association se fait à *vue*, c'est à dire dans l'environnement visible de l'utilisateur et ne nécessite pas d'explorer la base pour le trouver ; créer un élément d'annotation nouveau, ce qui ramène à la tâche d'annotation elle-même dans un contexte très précis (l'utilisateur est guidé par l'élément d'annotation qu'il souhaite mettre en relation) et enfin ; l'exploration de la base à la recherche d'éléments d'annotations (donc d'autres unités audiovisuelles) intéressantes à mettre en relation. Cette dernière possibilité est une tâche plus complexe qui mobilise les possibilités de l'outil comme l'illustre le diagramme de la figure 8 correspondant à Explorer contextuellement un EA pour trouver un EA-cible.

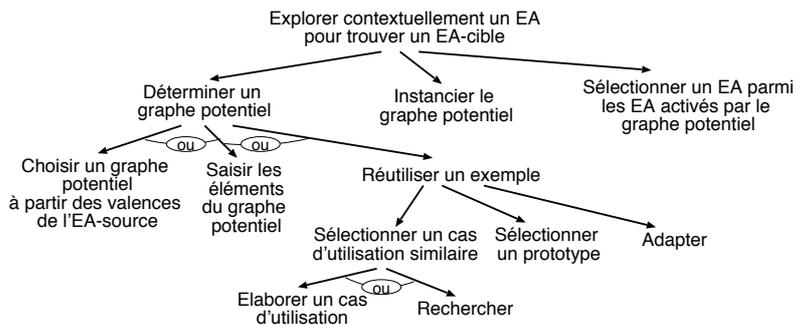


FIG. 8 – Explorer un contexte d'EA : un processus récurrent dans SESAME

Cette tâche est complètement fondée sur l'exploitation de la notion de graphe potentiel décrit dans le modèle d'utilisation<sup>9</sup>. L'exploration se déroule en trois étapes successives : **déterminer un graphe potentiel**, caractéristique de la portée du contexte considéré par l'utilisateur ; **instancier le graphe potentiel** qui consiste à trouver tous les sous-graphes de la base satisfaisant le graphe potentiel, ce qui permet d'atteindre un ensemble d'éléments d'annotations candidats à la mise en relation contextuelle directe ; enfin, **sélectionner un EA** (cette dernière opération se fait à nouveau à *vue*). La première sous-tâche **déterminer un graphe potentiel** est à la fois la plus puissante pour exprimer l'objectif de l'utilisateur, mais aussi la plus difficile à réaliser efficacement. Deux sous-tâches sont nécessaires : sélectionner un exemple ; adapter l'exemple. Plusieurs alternatives sont possibles pour la tâche de sélection : utiliser les valences de l'EA source, c'est à dire, directement utiliser un graphe proposé comme possible ; fabriquer directement un graphe potentiel, ce qui suppose de bien connaître la base et devrait être réservé aux *experts*, et enfin ; réutiliser un exemple de graphe à adapter au contexte courant. Cette dernière alternative correspond à un assistant à la tâche qui peut s'imaginer selon deux alternatives : sélectionner un graphe sur la base de son utilisation dans le cadre d'un cas d'utilisation similaire (des constructions de graphes potentiels caractérisés par des EA source différents), ou ; sélectionner un graphe dans un ensemble de *prototypes*, c'est à dire des

9. A noter également que les possibilités de *navigation* sont décrites également par des graphes potentiels.

graphes potentiels types caractérisés par un EAA et non pas un EA. La différence essentielle entre la première et la seconde alternative réside dans l'exploitation de l'expérience concrète contextualisée, par opposition à l'exploitation d'une expertise non contextualisée. Dans le cas de la réutilisation de cas d'utilisation similaires, l'adaptation consiste à substituer des éléments du graphe potentiel exemple par de nouveaux éléments corrigeant ainsi les écarts de contexte. Cette adaptation n'est possible que parce que les cas d'utilisation sont rationalisés par des modèles de tâche (par exemple une dimension d'analyse particulière dans SESAME). Adapter un prototype est en fait plus difficile car il n'est pas contextualisé, ce qui ne permet pas de guider l'utilisateur sur les valeurs d'instanciation qu'il convient de mettre dans tel ou tel noeud du graphe potentiel type.

## 5 Discussion

### 5.1 Travaux en rapport

L'approche de structuration des connaissances que nous mettons en œuvre avait été pressentie dans le domaine de l'assistance à l'opérateur [4] sans fournir de cadre d'étude précis. Les mêmes idées fondamentales ont suscité des travaux plus récents comme [17] pour les appliquer à la mémoire d'entreprise ou encore comme [12] dans le domaine de la conception. D'autres travaux, spécialisés dans la recherche d'information [1] ou limités au domaine financier [7] ont implicitement la même approche.

Le modèle de tâche d'annotation de SESAME fournit une assistance considérable à la mise en relation des annotations en s'appuyant sur la mise en contexte. D'autres études se sont interrogées sur la notion d'annotation en lien avec leur utilisation, par exemple [8] [18]. Parmi les travaux d'actualité, citons le développement de la norme de description de documents multimédias MPEG7<sup>10</sup>, et la proposition de l'INA [2].

### 5.2 Comment apprendre de l'expérience concrète : premières pistes

L'expérience concrète est représentée par des cas d'utilisation, eux-mêmes rationalisés par les modèles disponibles. Plusieurs directions de recherche sont ouvertes pour exploiter ces cas d'utilisation :

- Les cas d'utilisation sont des illustrations de l'exploitation des connaissances pour réaliser une tâche. L'apprentissage peut se faire par renforcement des explications utilisées pour assembler les éléments d'un cas d'utilisation, tandis qu'une différenciation d'explications peut être mise en évidence par les explications que l'utilisateur est amené à modifier ou ajouter. Le processus de validation d'une «mémoire d'entreprise» pourrait s'appuyer sur ce type d'approches. Dans SESAME, l'utilisation d'un EAA ou d'une valence s'illustre et s'explique directement par un exemple dans la base.
- Les valences peuvent être facilement acquises par induction à partir des graphes potentiels utilisés. Ce processus d'apprentissage est en boucle ouverte sur l'utilisateur expert qui doit valider les candidatures de valences avant de les ajouter. Le lecteur aura compris que si la notion de valence fait bien partie du modèle d'utilisation, les valeurs des valences sont clairement associées à une tâche. En conséquence, le processus de valorisation des valences est un processus de description de tâche et les valences d'un EAA dépendent alors de la tâche courante.
- Les relations mises en place par l'utilisateur entre éléments d'un cas d'utilisation peuvent être, par répétition, autant d'indications statistiques de relations plus générales intégrables dans le modèle d'utilisation et/ou de tâches. Dans le cadre de SESAME, des relations élémentaires (épisodiques par définition) répétées dans la base entre EA issus d'EAA identiques poussent à induire des relations entre EAA. Il s'agit d'une sorte de *data mining* validé à l'avance par les différents épisodes d'annotation eux-mêmes (un épisode d'annotation est un cas d'utilisation dans le cadre de SESAME).

---

10. Voir <http://drogo.csel.t.stet.it/mpeg/standards/mpeg-7/mpeg-7.htm>.

- Enfin les cas d'utilisation peuvent servir directement à l'assistance à la tâche. Un assistant générique pourrait se spécialiser en fonction d'une description explicite de tâche et de base de cas d'utilisation qu'il pourrait ainsi exploiter pour couvrir plus facilement l'espace des problèmes correspondant à chacune des tâches. Cette approche se décline facilement au niveau des sous-tâches. Cette façon de voir les choses pourrait faciliter considérablement la réalisation d'assistants pour des tâches mal définies au départ, mais s'appuyant sur un modèle d'utilisation unique. Ainsi, dans SESAME on pourrait se baser sur les contextes (représentés par des graphes contextuels) pour essayer de distinguer des tâches non explicites dans l'activité de l'utilisateur.

## 6 Conclusion

Dans cet article nous avons présenté une approche originale de modélisation de l'exploitation d'un outil informatique par un *modèle d'utilisation* (tous les concepts de l'outil) et des *modèles de tâche* (les concepts liés aux tâches réalisées à l'aide de l'outil). Les cas d'utilisation décrivent des épisodes de l'activité concrète de l'utilisateur. Ils s'expliquent par les relations de ses éléments aux modèles de l'outil et par les relations directes ajoutées par l'utilisateur. De plus, en tant que *traces* de l'activité, ils ouvrent la voie à un apprentissage et à une aide à l'utilisateur basés sur l'expérience. La deuxième partie de l'article illustre les définitions de la première, et présente de façon simplifiée le modèle d'utilisation (Strates-IA) de SESAME, un système d'information audiovisuelle, en introduisant les notions nouvelles de graphes potentiels, de graphes potentiels contextuels, d'instanciations, et de valences. Le modèle de la tâche *annoter* est ensuite décrit sous forme de diagrammes de décomposition de tâches, décrits à partir du modèle d'utilisation, ainsi que la notion de cas d'utilisation. Une discussion sur les possibilités d'apprentissage à partir de l'expérience concrète des cas d'utilisation, illustrée par des exemples liés à SESAME, termine cet article.

Le cadre mis en place dans cet article, de modélisation d'un outil informatique en vue d'un apprentissage et d'une aide à l'utilisateur basés sur des cas d'utilisation concrets nous apparaît comme extrêmement important. En effet, la notion de description de tâche spécialisant un modèle d'utilisation et celle de cas d'utilisation illustrant les modèles et rationalisés par eux sont centrales pour pouvoir réutiliser efficacement l'expérience concrète pour assister l'utilisateur dans ses différentes tâches. Le modèle des Strates-IA illustre parfaitement une telle démarche : tout en ouvrant des voies prometteuses au niveau de la description et de l'utilisation de documents audiovisuels, il autorise naturellement capitalisation et partage de connaissances.

## Références

- [1] R. Atterman and D. Griffin. Improving case retrieval by remembering questions. In *Proc. of the Thirteenth National Conference on Artificial Int. and the Eighth Innovative Applications of Artificial Int. Conf.*, pages 678–83. MIT Press Cambridge, MA, USA, 1996.
- [2] G. Auffret, J. Carrive, O. Chevet, T. Dechilly, R. Ronfard, and B. Bachimont. Audiovisual-based hypermedia authoring: using structured representations for efficient access to av documents. In *ACM Hypertext'99*, Darmstadt, Germany, Feb. 1999.
- [3] Bruno Bachimont. Bibliothèques numériques audiovisuelles : des enjeux scientifiques et techniques. *Document Numérique, Numéro Spécial "Les Bibliothèques Numériques"*, 2(3–4):219–242, Jan. 1999.
- [4] Guy Boy. *Assistance à l'opérateur : une approche de l'intelligence artificielle*. Edition TEKNEA, Toulouse, 1988.

- [5] B. Chandrasekaran, J.R. Josephson, and V. Richard Benjamins. The ontology of tasks and methods. In <http://cpsc.ucalgary.ca/KAW/KAW98>, editor, *Proceedings of the 11th Workshop on Knowledge Acquisition, Modelling and Management*, 1998.
- [6] F. Corvaisier, A. Mille, and J.-M. Pinon. Radix 2, assistance à la recherche d'information documentaire sur le web. In *IC'98, Ingénierie des Connaissances, Pont-à-Mousson, France*, pages 153–163. INRIA-LORIA, Nancy, 13-15 Mai 1998.
- [7] O. Curet, J. Elliot, and M. Jackson. Designing knowledge discovery based systems in business, finance and accounting with a case-based approach: two case studies. In *IEEE Colloquium on Knowledge Discovery and Data Mining, London, UK*, pages 2/1–4, oct. 1996.
- [8] M. Davis. Media streams: An iconic visual language for video annotation. In *Proceedings of the 1993 IEEE Symposium on Visual Languages*, pages 196–203, Bergen, Norway, August 1993. IEEE Computer Society Press.
- [9] C. Faron, I. Akoulchina, and J.G. Ganascia. Integrating knowledge in hypermedia: the satellite system. In *Symposium on Robotics and Cybernetics. CESA '96 IMACS Multiconference. Computational Engineering in Systems Applications*, pages 94–102, Lille, 9-12 July 1996.
- [10] B. Fuchs. *Raisonnement à partir de cas: Représentation des connaissances et du raisonnement basé sur l'expérience. Application à la supervision industrielle*. Thèse en informatique, CPE-Lyon et Université Jean Monnet, Saint-Etienne, France, 1997.
- [11] A. Mille, B. Fuchs, and B. Chiron. Le raisonnement fondé sur l'expérience : un nouveau paradigme en supervision industrielle? à paraître dans *la Revue d'Intelligence Artificielle*, 1999.
- [12] D. Pi-Sheng. Using case-based reasoning approach to the support of ill-structured decisions. *European Journal of Operational Research*, 93(3-4):357–364, 1996.
- [13] Prié, J.-M. Jolion, and A. Mille. Sesame: modèle conceptuel de description de documents audiovisuels et assistants d'annotation de contenu. In *4èmes journées d'études et d'échanges compression et représentation des signaux audiovisuels: Coresa 98*, pages 227–234. France Telecom, Cnet, Lannion, 1998.
- [14] Y. Prié, A. Mille, and J.-M. Pinon. Ai-strata: A user-centered model for content-based description and retrieval of audiovisual sequences. In *First Int. Advanced Multimedia Content Processing Conf.*, pages 143–152, Osaka, Nov. 1998. To appear in LNCS, Springer-Verlag.
- [15] Y. Prié, A. Mille, and J.-M. Pinon. Une approche de modélisation de documents audiovisuels en strates interconnectées par les annotations. In *IC'98, Ingénierie des connaissances, Pont-à-Mousson, France*, pages 143–152. Inria-Loria, Nancy, 13-15 Mai 1998.
- [16] A.Th. Schreiber, B.J. Wielinga, and J.A. Breuker, editors. *Kads: A Principled Approach To Knowledge-Based System Development*, volume 11 of *Knowledge-Based Systems*. Academic Press, London, 1993.
- [17] G. Simon. Aide à la conception et à la validation de mémoire d'entreprise. In *IC'98, Ingénierie des Connaissances, Pont-à-Mousson, France*, pages 267–275. INRIA-LORIA, Nancy, 13-15 Mai 1998.
- [18] K. Zettsu, K. Uehara, K. Tanaka, and N. Kimura. A time-stamped authoring graph for video databases. In *Databases and Expert Systems Applications, LNCS 1308*, pages 192–201. Springer-Verlag, 1997.