



HAL
open science

Annotation formelle graphique de documents multimédia

Nicolas Moreau, Michel Leclère, Michel Chein, Alain Gutierrez

► **To cite this version:**

Nicolas Moreau, Michel Leclère, Michel Chein, Alain Gutierrez. Annotation formelle graphique de documents multimédia. IC 2007 - 18es Journées Francophones d'Ingénierie des Connaissances, Jul 2007, Grenoble, France. pp.313-324. lirmm-00194647

HAL Id: lirmm-00194647

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00194647>

Submitted on 7 Dec 2007

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

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

Annotation formelle graphique de documents multimédia

Nicolas Moreau, Michel Leclère, Michel Chein, Alain Gutierrez

LIRMM, CNRS - Université Montpellier 2
161 rue Ada, F-34392 Montpellier cedex 5 - France
{moreau,leclere,chein,gutierrez}@lirmm.fr

Résumé : Cet article présente une méthodologie et un atelier de construction graphique de descriptions formelles de documents. Cet atelier est destiné à être intégré dans des applications de gestion de corpus de documents nécessitant l'utilisation d'un processus d'annotation basé sur l'interprétation du contenu des documents (en particulier l'indexation de documents audiovisuels). L'approche proposée repose sur l'élaboration d'un modèle de description multidimensionnel des documents permettant de contextualiser les différentes descriptions. Ce modèle est exprimée dans le formalisme des graphes conceptuels. L'atelier exploite l'API CoGITaNT de raisonnement sur ce formalisme et notre nouvelle API CoGUI de manipulation graphique des éléments de ce formalisme qu'elle décline en 3 outils : un éditeur de modèle de description basé sur l'élaboration d'une ontologie et d'une charte de description, un éditeur de description, et un outil de requêtage sur la base de descriptions construite. Dans cet article, nous présentons le modèle de description et le scénario de construction des descriptions.

1 Introduction

Annoter consiste à associer des données supplémentaires à un document ou une partie d'un document existant. Cette définition très générale masque une grande variété de processus d'annotations qui se différencient par :

- leur mise en œuvre automatique ou manuelle ;
- le langage utilisé, de la langue naturelle à un langage formel en passant par des couleurs ou des signes ;
- l'objectif de l'annotation : commentaire, information, qualification, filtrage, indexation, etc ;
- la nature de l'annotation : objective (ajout de méta-données classiques : auteur, date...) ou subjective (expression d'un point de vue, etc).

Dans ce travail, nous nous intéressons à une annotation formelle (cf. Euzenat (2005)) de document, basée sur l'utilisation d'un langage de représentation de connaissances et l'exploitation d'ontologies. Cette approche peut s'appliquer indifféremment sur tout type de documents, mais est particulièrement pertinente lorsqu'il s'agit d'exploiter un corpus de documents audiovisuels où l'annotateur est amené à choisir une interprétation

d'un document et à la décrire pour permettre la réutilisation de ce document (cf. Isaac *et al.* (2005)).

Ces dernières décennies de nombreuses propositions ont été faites pour des formalismes de représentation de connaissances, qu'ils soient orientés ontologies formelles comme les logiques de descriptions, orientés utilisateurs tels les topic maps ou les graphes conceptuels, ou orientés interopérabilité avec les langages du web sémantique tels que RDF/S, OWL... Le principal inconvénient – et frein à leur usage – est l'important investissement humain qu'ils nécessitent, tant dans l'élaboration du vocabulaire conceptuel de description que dans le travail proprement dit d'écriture de la description. Peu de travaux s'intéresse aux méthodes et outils permettant de rendre efficace et rigoureux le travail de construction d'une description par un annotateur (dans cet article, le terme annotateur réfère aux personnes spécialistes d'au moins un aspect particulier des documents manipulés et ayant vocation à exprimer sur ces aspects une information). Quand un éditeur dédié est proposé pour un langage formel de représentation de connaissances, il se contente souvent de proposer des fonctionnalités d'assemblage des primitives de base (cf. par exemple WebKB¹, ou TooCom²). Les plus évolués sont basés sur l'identification de patron de description permettant de générer des formulaires à compléter (cf. par exemple Protégé³) mais ces formulaires ne laissent pas la liberté de compléter librement les instances de ces patrons.

Notre participation à différents projets de valorisation et diffusion de corpus multimédia nous a amené à proposer une méthodologie de construction de description basée sur l'élaboration d'un modèle de description multidimensionnel. Ce modèle de description s'appuie sur une ontologie de domaine afin de s'assurer que les descriptions produites par plusieurs annotateurs avec différents points de vue soient exploitables conjointement, qu'il étend par une *charte de description* qui guide l'annotateur dans son travail d'élaboration de la description formelle associée à un document. L'un des enjeux principaux de cette charte est d'inciter l'annotateur à élaborer une description plus riche (i.e. un réseau sémantique de concepts) des documents annotés alors que son travail traditionnel se limite à l'identification de concepts (i.e. des mots-clés). L'objectif est d'assister au maximum l'annotateur dans sa tâche afin de limiter le temps passé à la construction de ses descriptions. Pour cela, le modèle de description repose sur l'identification de "schémas conceptuels" que l'annotateur va sélectionner, assembler et préciser. Ces schémas sont de plus contextualisés à certaines parties de l'ontologie de domaine afin de permettre à l'outil de s'adapter à la dimension de l'annotation en cours. Ainsi en quelques clics, l'annotateur peut construire une description riche et originale.

L'atelier proposé répond aux besoins spécifiques que requiert l'indexation des documents audiovisuels tels que définis dans Isaac *et al.* (2005) ; il reprend les solutions proposées qu'il étend en les intégrant dans le formalisme des graphes conceptuels (Sowa, 1984). Ce formalisme dispose de nombreux atouts qui, nous semble-t-il, le rend très adapté à son utilisation comme langage de description dans un processus d'annotation formelle de documents :

- il est graphique dans la lignée des réseaux sémantiques qui permettent une repré-

¹<http://www.webkb.org/interface/CGtextualEditor.html>.

²<http://sourceforge.net/projects/toocom/>

³<http://protege.stanford.edu/>

- sentation et interprétation intuitive de la connaissance ;
- il permet une structuration et une contextualisation des connaissances grâce aux emboîtements de graphes ;
- il permet l’expression de connaissances variées : des descriptions, des patrons, des règles, des contraintes (cf. la famille SG (Baget & Mugnier, 2002)) ;
- il est logiquement fondé permettant ainsi de définir les notions de consistance de descriptions, de bases de descriptions et de déduction de connaissances à partir de ces descriptions ;
- ses mécanismes de raisonnement sont basés sur la théorie des graphes permettant ainsi de visualiser graphiquement les inconsistances et déductions (et de bénéficier de l’efficacité des algorithmes développés pour cette théorie) ;
- il dispose de nombreuses proximités avec RDF/S le langage standard d’annotation du web sémantique (cf. les travaux de Baget (2005)) et les Topics Maps la norme ISO de description de documents (cf. Carloni *et al.* (2006)) permettant ainsi l’export des descriptions construites vers d’autres “outils sémantiques”.

Notons enfin que ce formalisme est implémentée dans COGITANT, l’API de développement d’applications à base de graphes conceptuels, utilisable librement (CoGITANT, 1997).

L’identification et la structuration des modèles de description sur lesquels repose notre proposition est issue des retours d’expérience des différents projets dans lesquels nous sommes impliqués (Opales, Saphir, Logos). Tous les exemples de cet article sont issus du projet Saphir. Saphir est un projet ANR-RIAM en cours mené en partenariat avec l’INA⁴, la MSH⁵, le LIRMM, le LERIA⁶ et NETIA⁷, qui vise à l’élaboration d’un Système d’Assistance à la Publication Hypermédia par spécification Intentionnelle et modélisation Rhétorique. Dans le cadre du projet Saphir notre outil est intégré à un atelier de segmentation et d’indexation de vidéos développé par l’INA.

La suite de l’article est organisée de la manière suivante. La section 2 présente brièvement le modèle GC que nous considérons dans cette proposition. La section 3 présente les différents composants des modèles de description que notre méthodologie met en œuvre et leur formalisation à l’aide des graphes conceptuels. Enfin la section 4 présente l’atelier logiciel qui outille la méthodologie proposée.

2 Le formalisme des graphes conceptuels

Notre proposition s’appuie sur le formalisme de graphes conceptuels que nous développons depuis 15 ans au LIRMM (Chein & Mugnier, 1992) et qui se distingue de la proposition initiale de Sowa (1984) par le parti pris de conserver un formalisme à base de graphes que ce soit au niveau des représentations ou au niveau du raisonnement. Il se base sur un support qui fait office d’ontologie simple et que l’on peut compléter par des règles et contraintes de graphes (cf. la famille SG (Baget & Mugnier, 2002)). Dans

⁴L’institut national de l’audiovisuel <http://www.ina.fr>.

⁵La maison des sciences de l’homme <http://semioweb.msh-paris.fr/AAR/FR/>.

⁶Le laboratoire d’informatique de l’université d’Angers.

⁷Un éditeur de logiciels de gestion et diffusion des contenus audio et vidéo <http://www.netia.net>.

Chein & Mugnier (2004), l'utilisation de types conjonctifs est introduite : on ajoute alors dans l'ontologie des ensembles de types incompatibles limitant les conjonctions possibles.

2.1 Les graphes simples

Un *graphe conceptuel simple (SG)* est un graphe biparti étiqueté : l'une des classes de sommets, dite de sommets *concepts*, représente des entités, et l'autre, dite de sommets *relations*, représente des relations entre ces entités ou des propriétés de ces entités.

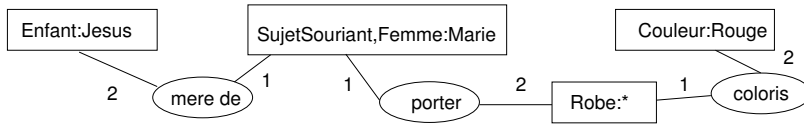


FIG. 1 – SG description de *La vierge à l'enfant* de Léonard de Vinci

Un sommet concept d'un SG est étiqueté par un couple $t : m$ où t est un type de concept (possiblement conjonctif) et m est un marqueur. Si le sommet représente une entité précise alors m est un identifiant représentant cette entité (le sommet est dit *individuel*), sinon m est noté $*$ (le sommet est dit *générique*). Le sommet [*Sujet-Souriant,Femme:Marie*] du graphe de la figure 1 désigne Marie, tandis que le sommet [*Robe:**] désigne "une" robe. Un sommet relation est étiqueté par une relation r et, si n est l'arité de r , n arêtes sont incidentes à ce sommet; ces arêtes sont totalement ordonnées. On note $G = (C_G, R_G, E_G, l_G)$ un SG où C_G est l'ensemble des sommets concepts, R_G l'ensemble des sommets relations, E_G l'ensemble des arêtes et l_G la fonction d'étiquetage des sommets et des arêtes.

Les étiquettes des sommets sont prises dans un vocabulaire appelé **support**. Nous considérerons ici un support (i.e. une ontologie simple) comme une structure $O = (T_C, T_R, B, \sigma)$ où :

- T_C et T_R sont des ensembles ordonnés par spécialisation de types de concept et de relations d'arité quelconque respectivement ($t' \leq t$ s'interprète par " t' est une spécialisation de t ");
- B est un ensemble de types bannis. Un type banni est un type conjonctif, par exemple $\{Humain, Objet\}$ qui ne peut pas désigner une même entité (les sous-types d'un type banni sont eux-même des types bannis).
- σ associe à chaque relation n -aire, sa signature, un n -uplet de types de concept qui définit le type maximal de chacun de ses arguments.

Un graphe G est **cohérent** par rapport à une ontologie $O = (T_C, T_R, B, \sigma)$ si :

- Les étiquettes des sommets relations et types des sommets concepts de G appartiennent resp. à T_R et T_C ;
- Le type conjonctif de chaque individu – le type des sommets concepts génériques, et la conjonction des types des sommets concepts ayant un même identifiant pour les individuels – n'est pas sous-type d'un type banni dans B ;
- Les sommets relations respectent les signatures définies dans σ ;

La notion fondamentale pour comparer des SG est une application d'un SG dans un autre appelée **projection**. Intuitivement, l'existence d'une projection de G dans H montre que la connaissance représentée dans G est contenue dans H . Une projection π de G dans H est plus précisément une application de C_G dans C_H et de R_G dans R_H qui conserve les arêtes (si on a une arête entre r et c étiquetée i dans G alors on a une arête entre $\pi(r)$ et $\pi(c)$ étiquetée i dans H) et peut "spécialiser" les étiquettes des sommets. Le résultat fondamental sur lequel le formalisme SG s'appuie est l'équivalence entre l'existence d'une projection et la déduction logique sur les formules associées par une transformation ϕ en logique des prédicats aux SG (cf. (Sowa, 1984) et (Chein & Mugnier, 1992)).

2.2 Les graphes emboîtés typés

Le modèle des NCG (nested conceptual graphs) typés (TNG) a été introduit pour contextualiser des connaissances (cf. (Chein & Mugnier, 2007; Chein *et al.*, 1998)). Cette classe est récursivement définie par :

1. Un graphe emboîté typé est obtenu à partir d'un SG en ajoutant à chaque sommet concept c un troisième champ, égal à $**$.
2. Soient G un graphe emboîté typé, c_1, c_2, \dots, c_k des sommets concepts de G , n_1, n_2, \dots, n_k des types d'emboîtement, et G_1, G_2, \dots, G_k des graphes emboîtés typés. Le graphe obtenu en substituant l'ensemble $\{(n_1, G_1), (n_2, G_2), \dots, (n_k, G_k)\}$ au troisième argument $**$ de c_i est un graphe emboîté typé.

On ajoute alors à la structure ontologique une arborescence de types d'emboîtements T_E . Les types d'emboîtement permettent de préciser la nature du lien qui lie un concept à son graphe emboîté (description, association, composition, historique...), l'ordre permet d'introduire une spécialisation entre eux (par exemple $Composition \leq Association$).

Graphiquement, les couples (n_i, G_i) associés au sommet c sont représentés par des boîtes à l'intérieur du sommet c , étiquetées par n_i et contenant le graphe G_i . L'exemple de la figure 2 montre un sommet concept contenant deux emboîtements (de type *Interactions* et *Relations*).

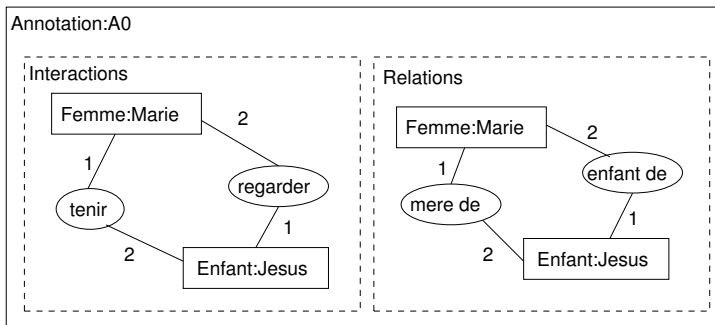


FIG. 2 – Un graphe emboîté typé

2.3 Les règles et contraintes

Au formalisme “noyau” des graphes conceptuels simples, la famille \mathcal{SG} ajoute deux types de connaissance : les *règles* et les *contraintes* (Baget & Mugnier, 2002).

Une règle exprime une connaissance de la forme “si hypothèse alors conclusion”, où hypothèse et conclusion sont deux SG. La figure 3 présente un exemple de règles. Les pointillés lient certains sommets de l’hypothèse et de la conclusion ; La règle exprime que si une femme “est mère” d’un enfant, alors cet enfant “est l’enfant” de cette femme. En appliquant la règle au SG de la figure 1 on obtient la relation supplémentaire “*enfant de*” entre [*SujetSouriant,Femme:Marie*] et [*Enfant:Jesus*]

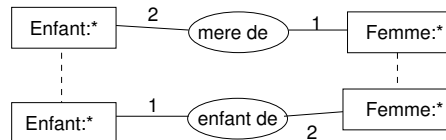


FIG. 3 – Exemple de règle de SG

Les contraintes permettent d’étendre la notion de cohérence d’un graphe. On distingue les contraintes positives des contraintes négatives :

- une contrainte positive a la même forme syntaxique qu’une règle mais exprime une obligation conditionnelle d’apparition d’un motif dans un graphe : “si une certaine connaissance est présente alors une autre doit l’être aussi”. Le graphe de la figure 3 déclaré comme une contrainte imposerait qu’un graphe qui a la relation “mère de” entre un enfant et une femme doit également avoir la relation “enfant de” entre eux ;
- une contrainte négative a la même forme syntaxique qu’un SG, mais exprime qu’un certain motif “ne doit pas être trouvé”.

3 Méthodologie d’annotation

Les premières expérimentations avec nos partenaires-annotateurs nous ont montré que le travail d’annotation ne se fait pas “en une passe” sur un document en entrée, mais est plutôt dirigé par une analyse descendante suivant une certaine idée répondant à un objectif précis d’annotation. On repère très grossièrement les parties du document qui relèvent de cette idée en décrivant très superficiellement ces parties, puis ces parties sont redécoupées en extraits de plus en plus petits auxquels on attache des descriptions de plus en plus précises.

Cette approche, dite de stratification dans la communauté du document audiovisuel (cf. Prié (1999)), réifie l’acte d’annotation en un objet structuré identifiant la source annotée, la description qui lui est associée par cette annotation et un certain nombre de méta-données d’annotation : auteur, date... Rappelons que les descriptions qui nous intéressent ici sont celles dites interprétées (celles qui sont issues d’une activité humaine de représentation), tandis que de nombreux travaux visent à identifier d’autres caractéristiques (sons, mouvements de caméras...) sur une source annotée.

Dans notre atelier nous ne manipulerons que des identifiants de ces annotations, en permettant d'associer une description formelle à chaque annotation (deux annotations pouvant repérer la même source, on peut associer de manière indépendante deux descriptions à un même extrait).

3.1 Les dimensions d'annotation

Le modèle doit permettre de manipuler des descriptions multi-dimensionnelle au sens où il doit permettre d'associer dans une même annotation des caractéristiques issues de différents objectifs d'annotation. Ces différentes dimensions ne sont par ailleurs pas indépendantes les unes des autres. Bien au contraire, elles se complètent et un objectif d'annotation nécessite souvent la combinaison de différentes dimensions.

Dans le cadre du projet Saphir, un groupe d'annotateurs s'intéresse aux journaux télévisés qu'ils décrivent en analysant la bande son, la bande vidéo et les incrustations d'images et textes. Un autre groupe s'intéresse à des interviews d'expert et veulent représenter le discours de l'orateur : leur méthodologie consiste à d'abord décrire une des thématiques de l'interview, puis à analyser la rhétorique sur ce thème en indiquant les différents actes et relations narratives utilisés ; chaque acte pouvant à son tour être décrit en précisant plus finement la thématique abordée et la rhétorique utilisée...

Les différentes dimensions sont amenées à partager une partie de leur vocabulaire conceptuel ; dans le cas de l'analyse des journaux télévisés par exemple, les actants (présentateur, invité...) se retrouvent à la fois dans l'analyse audio et l'analyse vidéo. L'idée n'est donc pas de permettre de regrouper dans une même description des graphes réalisés sur des ontologies distinctes mais plutôt de permettre d'identifier sur une unique ontologie les parties mobilisées par chaque dimension et d'adapter l'environnement d'annotation à la dimension courante.

Les emboîtements du modèle des graphes conceptuels fournissent cette contextualisation des connaissances par dimension. Les types d'emboîtement permettent d'identifier la dimension utilisée. Afin de permettre la construction d'une charte de description au plus près de la démarche de l'annotateur, nous introduisons la notion d'*ontologie modulaire*. Une ontologie modulaire est une ontologie sur laquelle on a défini des *modules* identifiant le vocabulaire mobilisé par les différentes dimensions d'annotation.

Définition 1

Un module d'une ontologie $O = (T_C, T_R, T_E, B, \sigma)$ est un triplet $m = (T'_C, T'_R, T'_E)$ tel que :

- $T'_C \subseteq T_C, T'_R \subseteq T_R, T'_E \subseteq T_E$, où T'_C, T'_R, T'_E sont ordonnés par la restriction des ordres issus de T_C, T_R, T_E ;
- $\sigma[T'_R] \subseteq T'_C$: les types de concepts des signatures des relations sélectionnées dans le module doivent appartenir au module (l'idée étant que si la relation est sélectionnée pour participer à la dimension alors les arguments de cette relation doivent y participer aussi).

Chaque dimension d'annotation est donc représentée par un type d'emboîtement auquel on associe un module (un même module peut être associé à plusieurs types d'emboîtement).

Définition 2

Une ontologie modulaire est un triplet (O, M, D) où $O = (T_C, T_R, T_E, B, \sigma)$ est une ontologie, M un ensemble de modules définis sur O et D une application de T_E dans M représentant les dimensions.

À chaque module $m = (T'_C, T'_R, T'_E)$ d'une ontologie $O = (T_C, T_R, T_E, B, \sigma)$, on peut associer la sous-ontologie $O_m = (T'_C, T'_R, T'_E, B', \sigma')$ où :

- B' est la restriction de B aux types de T'_C ;
- σ' est l'application σ réduite aux éléments de T'_R .

Une sous-ontologie est donc une ontologie.

Nous appelons MNG (Modulary Nested Graph) le modèle de graphes construits sur une ontologie modulaire où chaque graphe contenu dans un emboîtement doit respecter la sous-ontologie de sa dimension.

Définition 3

Un MNG G est cohérent par rapport à une ontologie modulaire (O, M, D) si :

- G est un cohérent par rapport à O ;
- pour tout emboîtement e de G , le graphe G_e formé des sommets qui sont immédiatement contenus dans e et dont on ne conserve des emboîtements inclus que les boîtes racines vides est cohérent par rapport à la sous-ontologie O_m induite par le module de sa dimension soit $D(\text{type}(e))$.

3.2 La charte de description

Les dimensions d'annotation formalisées par la notion d'ontologie modulaire assurent un premier niveau de charte d'indexation puisqu'elles fixent une structuration préalable des graphes, en limitant notamment le vocabulaire conceptuel utilisable dans chaque dimension. Un deuxième niveau consiste à prédéfinir des "schémas conceptuels" que l'annotateur pourra sélectionner, assembler et modifier. L'identification et l'utilisation de schémas présente plusieurs avantages :

- inciter l'annotateur à fournir une description riche plutôt qu'un unique concept. L'annotateur travaille à partir d'un graphe générique sélectionné en fonction de son objectif d'annotation et n'a plus qu'à le modifier et/ou le compléter ;
- éviter le travail répétitif de reconstruction de graphes très proches d'une annotation à l'autre. Ce qui permet d'accélérer le processus d'annotation et évite la lassitude des annotateurs ;
- assurer une certaine unité à la base de descriptions en dirigeant les annotateurs vers un "genre de description" commun ;
- faciliter l'apprentissage du travail d'annotation par un nouvel annotateur en lui permettant de s'approprier rapidement le "genre de description" recherché.

Par le biais de ces schémas, les annotateurs peuvent modéliser une démarche d'annotation : ils identifient pour chaque dimension, un ou plusieurs schémas initiaux de description, puis des points d'extensions de ces schémas à partir desquels la description peut se poursuivre. Deux sortes de schémas ont été définis dans notre approche : les graphes patrons associés aux types d'emboîtement et les graphes prototypiques associés aux types de concept ou relations.

Les **graphes patrons** sont associés aux dimensions pour lesquelles ils représentent des manières différentes de commencer une description dans cette dimension. Ils sont donc cohérents à la sous-ontologie de la dimension à laquelle ils sont attachés (cf. Fig.4).

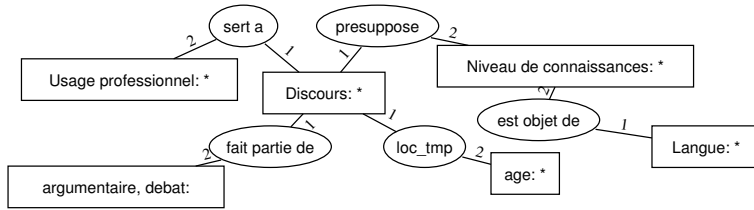


FIG. 4 – Graphe patron associé à la dimension *Rhétorique*

Les **graphes prototypes**, ou schémas de type, expriment des “mises en contexte” classiques d’un sommet de ce type. Ce sont des TNG à emboîtements vides qui contiennent un sommet spécial appelé *tête* ayant pour type le type dont ils sont le prototype (cf. Fig.5).

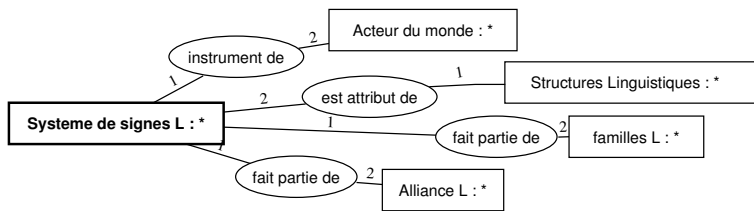


FIG. 5 – Graphe prototypique associé au type de concept *Système de signes linguistiques*

Les graphes prototypes étant associés aux types, ils sont potentiellement transversaux à toutes les dimensions. Aussi, lors de l’édition d’un graphe suivant une dimension, l’utilisateur n’a accès qu’aux prototypes dont tous les sommets sont étiquetés par des éléments du vocabulaire associé à la dimension.

3.3 Les axiomes

Une ontologie modulaire peut être complétée par des graphes "ontologiques" exprimant des connaissances générales sur une dimension (des axiomes). Ces graphes doivent donc être cohérents par rapport aux sous-ontologies des dimensions auxquelles ils sont rattachés. Trois types de représentation d’axiomes sont possibles.

Les **règles** permettent de simplifier la tâche des annotateurs en spécifiant, une fois pour toutes, des connaissances implicites. Elles peuvent par exemple servir à exprimer la symétrie ou transitivité d’une relation.

Les **contraintes** permettent d’apporter un caractère contraignant à la charte d’annotation : on peut par exemple restreindre la signature d’une relation à des paramètres

plus précis dans une dimension donnée ou imposer un environnement obligatoire à un concept d'un certain type.

Les **graphes d'individus** sont des graphes de faits ontologiques. Chacun de ces graphes contient au moins un sommet concept individuel. Ils représentent des connaissances encyclopédiques qui n'ont pas lieu d'être répétées à chaque annotation. Ils sont automatiquement attachés aux descriptions dès lors qu'elles réfèrent à l'un des individus qu'ils décrivent. L'ensemble des individus "ontologiques" est calculé à partir de ces graphes : tout individu qui y apparaît est considéré comme "ontologique" par opposition aux individus créés par les utilisateurs.

4 L'atelier logiciel

La méthodologie d'annotation proposée dans la section précédente a été outillée à l'aide de CoGUI⁸ notre interface graphique générique de construction et de manipulation de bases de connaissances en graphes conceptuels. CoGUI permet l'édition d'ontologies, des différents modèles de graphes, et dispose de fonctionnalités de vérification et validation des objets construits : certaines des propriétés à vérifier sont directement obtenues "par construction" ; pour les autres, une liste de messages d'erreurs et d'avertissements couplée à une visualisation graphique sur le graphe de la zone concernée facilite la correction des problèmes. De plus, CoGUI permet la mise en œuvre des différents mécanismes de raisonnements associés au modèle par un accès au serveur de raisonnement CoGITaNT.

CoGUI dispose d'un mécanisme de profil d'utilisation permettant de définir des interfaces adaptées à une tâche particulière. Trois profils ont été définis pour appuyer notre méthodologie : l'éditeur de modèle de description, l'outil d'élaboration des descriptions qui exploite ce modèle pour en permettre une construction aisée, et l'outil de requêtage qui permet l'interrogation de la base de descriptions selon une ou plusieurs dimensions. Les requêtes peuvent soit viser à retrouver les documents dont une annotation contient une certaine connaissance, soit viser à extraire des connaissances particulières des descriptions associées aux annotations d'un ou plusieurs documents de la base.

La première étape de définition du modèle est la création de l'ontologie modulaire. Les trois hiérarchies sont construites de manière graphique. Une définition informelle peut être associée à chaque primitive ontologique. L'ontologie permet le multilinguisme : elle est basée sur une langue principale et peut être déclinée sous différentes langues suivant les besoins des utilisateurs. Les modules sont construits en sélectionnant les types pertinents sur l'ontologie globale qui peut être construite précédemment ou pendant le processus de modularisation.

L'étape suivante consiste à définir la charte d'annotation : (1) saisie des graphes patrons, prototypiques et ontologiques, (2) association aux dimensions (types d'emboîtement), puis (3) vérification de la cohérence de ces graphes par rapport à la sous-ontologie à laquelle ils se rapportent.

Une fois le modèle d'annotation défini, vient la phase d'annotation proprement dite. Dans le projet SAPHIR, notre outil est intégré à l'outil de segmentation Audio Visual

⁸<http://www.lirmm.fr/coGUI/>

Studio développé par l'INA. Une description s'obtient après plusieurs étapes qui suivent généralement le scénario suivant :

1. L'utilisateur sélectionne une annotation (i.e. une structure d'Audio Visual Studio identifiant une partie d'un document). Un graphe conceptuel composé d'un unique sommet, appelé la tête de la description, typé `annotation` et ayant pour marqueur individuel l'identifiant de l'annotation est généré. Dans le projet Saphir, les annotations sont typées et ce type est repris comme type de la tête de la description : `strate`, `segment`, `image`, `détail`, `texte` ... L'utilisateur peut dès lors construire sa description en insérant différents emboîtements dans ce sommet. Le graphe de la figure 2 peut être vu comme un exemple de description.
2. L'utilisateur choisit une dimension d'annotation. L'outil créé automatiquement l'emboîtement correspondant et "plonge" dans la dimension, masquant la tête de la description pour permettre à l'utilisateur de se concentrer sur la dimension. L'outil calcule également la sous-ontologie associée à la dimension, pour n'offrir à l'utilisateur que les éléments pertinents de l'ontologie. L'utilisateur peut à tout moment revenir à la description dans sa globalité, pour voir les dimensions d'annotation qu'elle comporte.
3. L'utilisateur peut sélectionner un graphe patron pour démarrer un parcours d'annotation. À la création d'une description ou lors de l'activation d'une dimension d'annotation, l'outil propose la liste des graphes patrons correspondant à la dimension choisie. Le parcours d'annotation n'est aucunement contraignant, il sert juste de guide. L'utilisateur peut spécifier, généraliser ou même supprimer les sommets du graphe patron inséré.
4. L'utilisateur peut sélectionner un graphe prototypique. Lors de son annotation, l'utilisateur peut demander à compléter la description d'un sommet par l'un de ces contextes courants d'utilisation. Le système ajoute alors automatiquement les informations du prototype dans la description en assimilant le sommet courant au sommet tête du prototype. Comme pour les graphes patrons, ces graphes prototypes une fois ajoutés ne sont plus contrôlés et peuvent être modifiés.

En fin de saisie, l'utilisateur doit valider sa description d'annotation. Le test de cohérence de la description à l'ontologie modulaire est effectué et, en cas d'erreur, un ensemble de messages est affiché. Le mécanisme de visualisation graphique des erreurs permet alors de corriger sa description. Dans un deuxième temps, on vérifie le respect des contraintes via CoGITaNT.

5 Conclusion

Ce travail jette les bases d'une méthodologie d'annotation formelle basée sur la définition d'un modèle de description multi-dimensionnel permettant de guider, assister puis valider un travail d'annotation manuelle. Cette méthodologie est appuyée par Co-GUI notre interface graphique d'édition, de vérification et d'exploitation de graphes de connaissances.

Cette proposition demande maintenant à être validée par une expérimentation complète intégrant une phase d'exploitation réelle des descriptions construites. Le projet SAPHIR qui a pour objectif d'obtenir une chaîne éditoriale permettant à un expert du contenu, mais novice en « hypermédia et vidéo » de créer « facilement » une publication utilitaire basée sur une exploitation et une valorisation d'un corpus audiovisuel nous offre un premier cas d'expérimentation.

Enfin, la question de la validité des marqueurs individuels se pose. Il faut en effet s'assurer que les annotateurs n'utilisent pas deux marqueurs individuels différents pour désigner un même individu sinon cela entraînera du silence lors des requêtes. À l'opposé, ils ne doivent pas utiliser un même marqueur pour désigner deux entités différentes sinon cela amènera du bruit dans les réponses. L'outil actuel se contente de vérifier que l'ensemble des types caractérisant un marqueur individuel n'est pas sous-type d'un type conjonctif banni.

Références

- BAGET J.-F. (2005). Rdf entailment as a graph homomorphism. In *Proc. of ISWC'05*.
- BAGET J.-F. & MUGNIER M.-L. (2002). Extensions of Simple Conceptual Graphs : The Complexity of Rules and Constraints. *JAIR*, **16**, 425–465.
- CARLONI O., LECLÈRE M. & MUGNIER M.-L. (2006). Introduction de raisonnement dans un outil de gestion de connaissances basé sur les topic maps. In *17èmes journées francophones d'Ingénierie des Connaissances (IC'2006)*.
- CHEIN M. & MUGNIER M.-L. (1992). Conceptual Graphs : Fundamental Notions. *Revue d'Intelligence Artificielle*, **6**(4), 365–406.
- CHEIN M. & MUGNIER M.-L. (2004). Concept Types and Coreference in Simple Conceptual Graphs. In *Proc. ICCS'04*, LNAI : Springer.
- CHEIN M. & MUGNIER M.-L. (2007). Positive Nested Conceptual Graphs. In *Proc. ICCS'97*, LNAI, p. 95–109 : Springer.
- CHEIN M., MUGNIER M.-L. & SIMONET G. (1998). Nested graphs : A graph-based knowledge representation model with FOL semantics. In *Principles of Knowledge Representation and Reasoning*, p. 524–535.
- COGITANT (1997). Conceptual graphs integrated tools allowing nested typed graphs. <http://cogitant.sourceforge.net>.
- EUZENAT J. (2005). L'annotation formelle de documents en (8) questions. In *Ingénierie des connaissances*. L'Harmattan.
- ISAAC A., BACHIMONT B. & LAUBLET P. (2005). Indexation de documents av : ontologies, patrons de conception et d'utilisation. In *16èmes journées francophones d'Ingénierie des Connaissances (IC'2005)*.
- PRIÉ Y. (1999). *Représentation de documents audiovisuels en Strates-Interconnectées par les Annotations pour l'exploitation contextuelle*. PhD thesis, INSA de Lyon.
- SOWA J. F. (1984). *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley.