

*Action EXMO**Échanges de connaissance structurée médiatisés par ordinateur**Rhône-Alpes*

THÈME 3A



*R*apport
d'Activité

2001

Table des matières

1	Composition de l'équipe	2
2	Présentation et objectifs généraux	2
3	Fondements scientifiques	3
3.1	Sémantique de la representation de connaissance	3
3.2	Transformations et propriétés	4
3.3	Interprétation des flux d'information	5
4	Domaines d'applications	6
4.1	Panorama	6
4.2	Ingénierie des systèmes de transformations	6
4.3	Technologies pour un « web sémantique »	6
5	Logiciels	7
5.1	Panorama	7
5.2	TRANSMORPHER : spécification et exécution de flux de transformations XML	7
5.3	DLML : Description Logic Markup Language	8
6	Résultats nouveaux	9
6.1	Interopérabilité sémantique	9
6.1.1	Familles de langages	9
6.1.2	Transformations porteuses de preuves	10
6.1.3	Composition de propriétés et vérification	11
6.2	Préservation de scénarii dans les documents multimédia	11
6.3	Treillis et information	12
7	Contrats industriels (nationaux, européens et internationaux)	13
7.1	Fluxmedia	13
8	Actions régionales, nationales et internationales	13
8.1	Actions nationales	13
8.1.1	Action de recherche coopérative ESCRIRE	13
8.2	Actions européennes	14
8.2.1	Réseau thématique OntoWeb : ontology-based information exchange for knowledge management and electronic commerce	14
9	Diffusion de résultats	14
9.1	Animation de la communauté scientifique	14
9.2	Enseignement	14
9.3	Participation à des colloques, séminaires, invitations	14
10	Bibliographie	15

1 Composition de l'équipe

Responsable scientifique

Jérôme Euzenat [CR Inria]

Assistante de projet

Françoise de Coninck [jusqu'au 1er octobre 2001]

Marie-Anne Dauphin [depuis le 1er octobre 2001]

Chercheurs doctorants

Olivier Brunet [allocataire Ministère]

Pierre-Antoine Champin [CIFRE (Dassault-Systèmes) en co-direction avec Alain Mille (LISI)]

Raphaël Troncy [CIFRE (Institut National de l'Audiovisuel) depuis le 1er décembre 2000]

Ingénieur

Laurent Tardif [mis à disposition par FLUXMEDIA du 1er février au 30 mars 2001]

Chercheur invité

Heiner Stuckenschmidt [du 20 août au 20 octobre 2001]

2 Présentation et objectifs généraux

Mots clés : représentation des connaissances, sémantique des représentations, transformations, ontologies, treillis, préservation de propriétés, web sémantique, représentation du contenu, sémiologie, Transmorpher, WWW, DLML, XML, XSLT.

EXMO étudie l'échange de connaissance structurée et formalisée. La connaissance est représentée dans des langages formellement définis. Ils peuvent aller de XML – métalangage structuré mais sans sémantique – aux langages de représentation de connaissance – structurés, sémantiquement définis mais spécialisés.

Le but de l'action EXMO est le développement d'outils théoriques et logiciels pour aider à l'organisation, la manipulation, la composition et la présentation d'éléments de connaissance structurés lors de la communication entre humains. Dans le processus de communication, l'ordinateur peut introduire une plus-value à son rôle de médium et de mémoire en accomplissant des tâches comme le formatage, le filtrage, la catégorisation, le test de consistance ou la généralisation.

Assurer l'adéquation et l'intelligibilité de la connaissance pour les interlocuteurs nécessite le développement d'une compréhension abstraite des représentations et des transformations qui leur sont appliquées. Les travaux de l'action EXMO sont focalisés sur deux aspects. L'aspect transformation rend compte des modifications de la connaissance pendant la communication alors que l'aspect communication concerne la préservation de l'intelligibilité de la connaissance transformée.

Axes de recherche

L'aspect transformation a pour but d'élaborer une « théorie générale des transformations » fondée sur les propriétés satisfaites par les transformations plutôt que sur les transformations

elles-mêmes. Parmi ces propriétés, on trouve la préservation du contenu ou de la structure, la traçabilité des sources ou, au contraire, la confidentialité. Dans un premier temps, on étudie certaines familles de propriétés particulières afin de mettre en évidence l'intérêt de l'approche :

- La préservation du sens (les conséquences d'une représentation transformée sont-elles les transformées de celles de la représentation initiale ?) est l'un de nos premiers sujets d'intérêt. Pour cela, nous étudions la façon dont elle se manifeste dans les familles homogènes de langages. Nous cherchons à exploiter conjointement les transformations, les propriétés qu'elles satisfont ainsi que les preuves de ces propriétés de manière à les vérifier ou les utiliser dans d'autres preuves.
- La préservation, ou non-préservation, de l'information est étudiée dans le cadre d'une modélisation par des treillis. Cette modélisation à l'intérêt de s'interpréter dans une logique épistémique intuitionniste qui devrait permettre de décider les propriétés d'une transformation vis-à-vis de l'information.
- Nous explorons aussi les transformations de la représentation temporelle des documents multimédia qui permettent d'adapter un document multimédia à un appareil précis. Le but ici est d'assurer que tous les scénarii d'exécution du document sont préservés en dépit des transformations (cela peut consister à garantir qu'un élément du document peut toujours être vu).

La partie communication, à plus long terme, cherche à contribuer à l'intelligibilité de la connaissance échangée entre les interlocuteurs. Pour cela on étudie des propriétés appelées « sémiologiques », c'est-à-dire qui concernent l'interprétation par un utilisateur humain de la représentation communiquée et de sa préservation.

Applications

Les applications anticipées sont l'ingénierie des systèmes de transformation (où le système d'information est compris comme un flux de transformations) et le « web sémantique » (serveurs de connaissance, représentation du contenu, transformation de représentations).

3 Fondements scientifiques

3.1 Sémantique de la représentation de connaissance

Depuis plusieurs années, la sémantique des langages de représentation de connaissance (logiques de descriptions, graphes conceptuels et représentations de connaissance par objets) a été étudiée ^[Neb90]. Elle est définie en général à l'aide de la théorie des modèles.

On définit inductivement sur la structure du langage une fonction d'interprétation (I) vers un ensemble nommé domaine d'interprétation (D). Cette fonction reflète la construction du sens d'une expression en fonction de ses composants. Les assertions de ce langage sont satisfaites par une interprétation si elles remplissent une condition (en général être interprétées dans un sous-ensemble particulier du domaine). Dans ce cadre un modèle est une interprétation dans

[Neb90] B. NEBEL, *Reasoning and revision in hybrid representation systems, Lecture Notes in Artificial Intelligence 422*, Springer Verlag, Berlin (DE), 1990.

laquelle tous les axiomes sont satisfaits. Une expression (δ) est alors une conséquence d'un ensemble d'axiomes (Δ) si elle est satisfaite par tous leurs modèles (ce qui est noté $\Delta \models \delta$).

Pour un ordinateur, le but consiste à déterminer si une expression particulière (par exemple une requête) est conséquence des axiomes (par exemple, la base de connaissances considérée). Pour cela on développe des systèmes exécutables, appelés démonstrateurs, qui peuvent être fondés sur l'interprétation de règles d'inférence ou sur des programmes plus classiques. Ils permettent de déduire des théorèmes (ce qui est noté $\Delta \vdash \delta$). Ces démonstrateurs vérifient la propriété de correction s'ils ne répondent positivement qu'en cas de conséquence et la propriété de complétude s'ils répondent positivement pour toutes les conséquences. Cependant, suivant le langage et sa sémantique, la décidabilité – c'est-à-dire la possibilité de créer de tels démonstrateurs – n'est pas garantie. Même dans le cas de langages décidables, leur complexité algorithmique peut être prohibitive.

Pour cela on est amené à réaliser un compromis entre l'expressivité d'un langage et la complexité ou la complétude des démonstrateurs associés. Ce choix s'est traduit par la définition de langages à l'expressivité bornée comme les graphes conceptuels ou les représentations par objets ou de familles de langages modulaires dont les démonstrateurs peuvent être étendus, comme les logiques de descriptions.

L'action EXMO s'appuie d'abord sur des langages dont la sémantique est ainsi définie afin d'établir les propriétés des manipulations informatiques appliquées aux représentations.

3.2 Transformations et propriétés

Les traitements appliqués aux représentations sont qualifiés de transformations. On se concentrera sur des transformations obtenues par assemblage de transformations élémentaires dont on ne se préoccupe pas de la description. On connaît par contre leurs entrées et sorties ainsi que des propriétés liant les entrées aux sorties.

On définit un système de transformation par l'ensemble de ses transformations élémentaires et l'ensemble de ses assembleurs de transformations. Un flux de transformation est un ensemble d'instances de transformations élémentaires assemblées et reliées par des canaux. Un flux de transformations est lui-même une transformation.

Plus concrètement, nos travaux concernent des transformations syntaxiques de documents XML (« Extensible Markup Language ») encodant des langages de représentation de connaissance. Nous cherchons à exploiter le langage de transformations XSLT (« XML Stylesheet Language Transformations »^[Cla99]) recommandé par le W3C, pour lequel nous avons proposé un langage d'assemblage de transformations.

Le but d'EXMO est d'étudier les propriétés des transformations et leur combinaison. Une propriété est simplement un prédicat booléen portant sur une transformation (par exemple, « préserver l'information » est un tel prédicat – il est vrai ou faux d'une transformation – et se vérifie s'il existe un procédé algorithmique permettant, pour toute représentation Δ , de retrouver Δ à partir de son transformé $\tau(\Delta)$). Les questions intéressantes sont alors celles de la décidabilité d'une propriété pour un système de transformations particulier et celle de vérifier, pour un flux de transformations donné qu'il vérifie bien une propriété.

[Cla99] J. CLARK, (ÉD.), « XSL Transformations (XSLT) », *Recommandation*, W3C, Cambridge (MA US), 1999, <http://www.w3.org/TR/xslt>.

Nous identifions deux types de propriétés : des propriétés de préservation et des propriétés à motifs qui consistent à tester si la présence d'un motif dans la représentation de départ entraîne la présence de ce motif dans la représentation d'arrivée.

Les propriétés de préservation nous intéressent plus particulièrement. Elles peuvent permettre de garantir la préservation d'un ordre entre la représentation d'entrée (Δ) et la représentation de sortie ($\tau(\Delta)$) comme son anti-préservation. Pour cela on peut identifier des propriétés :

- syntaxiques : comme la structure ou l'ordre des éléments ;
- sémantiques : comme la préservation des conséquences ($\tau(\Delta) \Rightarrow \Delta$, c'est-à-dire $\forall \delta, \Delta \models \delta \Rightarrow \tau(\Delta) \models \tau(\delta)$) ;
- sémiologiques : comme la préservation des interprétations (soit Σ les règles d'interprétation des expressions et \models^i l'interprétation de l'individu i , $\forall \delta, \forall i, j, \Delta, \Sigma \models^i \delta \Rightarrow \tau(\Delta), \tau(\Sigma) \models^j \tau(\delta)$).

3.3 Interprétation des flux d'information

L'analyse des flux d'information est centrée autour d'une formalisation à base de treillis (un ensemble E muni d'un ordre partiel \leq et dans lequel tout couple d'éléments a un plus petit majorant pour l'ordre). Les treillis permettent d'avoir une notion d'imprécision dans la description de l'information. L'information disponible en un point sur un objet est un élément de ce treillis. L'ordre permet de préciser qu'un élément du treillis contient plus d'information qu'un autre. On peut ainsi définir l'absence totale d'information (l'élément maximal de l'ordre) et un élément plus grand qu'un autre est une approximation de celui-ci (car il contient moins d'informations, mais ne le contredit pas). Ainsi, considérant deux systèmes échangeant de l'information, on peut écrire de façon rigoureuse qu'un système connaît précisément telle information d'un autre système, ou qu'il ne connaît cette information que partiellement, ou encore pas du tout.

Une transformation τ d'un tel treillis dans un autre permet de modéliser la modification subie par l'information lors de sa circulation (la transformation) d'un espace de représentation à un autre. En se basant sur cette approche, on étudie la façon dont se composent et s'assemblent les transformations. Cela se traduit pour les propriétés sur ces systèmes par d'autres transformations que l'on dérive des précédentes.

Cette étude se base sur la comparaison de ce formalisme avec d'autres travaux existant concernant la représentation de connaissance et les échanges d'information. Les principaux travaux considérés sont la théorie des flux d'information (*Information Flow Theory* [BS98]) et l'analyse de concepts [GW99].

[BS98] J. BARWISE, J. SELIGMAN, *Information Flow, the Logic of Distributed Systems*, Cambridge university press, Cambridge (UK), 1998.

[GW99] B. GANTER, R. WILLE, *Formal Concept Analysis*, Springer Verlag, Berlin (DE), 1999.

4 Domaines d'applications

4.1 Panorama

EXMO applique son savoir-faire à plusieurs domaines d'applications tels que la mémoire technique (voir [projet EXMO, année 2000, section contrats, module genie]) ou la réutilisation en conception (voir [projet EXMO, année 2000, section contrats, module prosper]). Mais les efforts du projet sont maintenant dirigés vers deux types d'applications particuliers : l'ingénierie des systèmes de transformations (§4.2) et les technologies pour un « web sémantique » (§4.3)

4.2 Ingénierie des systèmes de transformations

Résumé : *La diffusion et la transformation de documents structurés induisent une ingénierie des systèmes de transformations qu'il faudra pourvoir en logiciels et en méthodes formelles qu'EXMO cherche à construire.*

De plus en plus, l'informatisation et la mise en réseau des organisations les conduisent à échanger de l'information sous forme électronique. Le commerce électronique engendre un échange permanent de tels documents.

Comme les informations diffusées ne sont pas destinées ni adaptées à l'ensemble des membres d'une organisation, il est nécessaire de transformer leur structure et leur contenu. De manière similaire, les sites du Web sont de plus en plus souvent engendrés à partir de bases de données ou de fonds premiers et les documents du commerce électronique subissent de nombreuses transformations jusqu'à ce que le produit soit envoyé.

Les besoins d'interopérabilité ont conduit à la définition du langage d'expression de documents structurés XML. Des langages, tels qu'XSLT ou Omnimark, permettent d'implémenter des transformations autonomes, mais il semble inévitable que, dans l'avenir, il faudra gérer des systèmes de transformations complexes (voir §3.2).

L'ingénierie de ces systèmes de transformations nécessitera des outils, des méthodologies et des méthodes formelles. En effet, il sera nécessaire de vérifier qu'un système particulier ne laisse pas échapper plus d'information qu'autorisé ou que le flux dans sa globalité termine. Pour cela, il faudra disposer d'une description analysable de ces flux de transformations et de caractérisations formelles des propriétés escomptées. EXMO est concerné par les outils et les méthodes formelles et cherche à associer les deux en proposant des solutions pour la création de véritables ateliers de conception de systèmes de transformations (voir §5.2).

4.3 Technologies pour un « web sémantique »

Résumé : *Dans un univers où les documents du web seront annotés formellement, il sera nécessaire d'importer et de manipuler ces annotations en accord avec leur sémantique et en accord avec leurs utilisations. Nous travaillons aux technologies permettant cela.*

L'utilisation des technologies d'Internet est l'opportunité pour les entreprises d'accéder à et de partager la connaissance bien souvent difficilement accessible sous forme documentaire. Cependant, les limites de cette approche apparaissent rapidement : l'organisation des sites

se révèle une tâche coûteuse et la recherche en texte intégral peu efficace. La recherche et l'interrogation d'un site en s'appuyant sur le contenu des documents sont une nécessité et les formalismes de représentation de connaissance sont de bons candidats pour représenter ce contenu. La représentation du contenu permettra de le manipuler pour faire de la recherche par analogie, par spécialisation, par similitude, etc.

L'idée d'un « web sémantique » [BLHL01] consiste à adjoindre au web actuel (informel) des annotations (rédigées dans des langages exploitables par une machine) liées entre elles comme au web informel. Exploiter ce web sémantique demandera donc d'appréhender ces représentations formelles diversifiées. La problématique de l'action EXMO se trouve donc au cœur de la mise en œuvre du web sémantique.

L'ensemble des travaux d'EXMO ont généralement pour but de concourir à une meilleure appréhension des contenus. Cela concerne, bien entendu, les travaux sur l'intelligibilité des connaissances communiquées, mais aussi les travaux sur la transformation de représentations formelles. Les travaux sur le web sémantique se basent actuellement sur la notion d'ontologie (que l'on peut rapidement décrire comme une axiomatisation d'un domaine). Même s'il existe un jour un langage commun de représentation, il faudra importer les ontologies décrites dans d'autres langages d'une façon qui respecte la sémantique des langages manipulés. Apporter des solutions à ce problème fait partie des ambitions d'EXMO (voir §6.1).

5 Logiciels

5.1 Panorama

Les travaux de l'action EXMO sont susceptibles de donner lieu au développement de logiciels. Nous avons ainsi conçu et développé un système de spécification et d'exécution de flux de transformations (voir §5.2).

Dans le cadre d'expérimentations nous avons été amenés à développer des couches logicielles relativement légères. C'est le cas de DLML (voir §5.3) décrit ci-dessous.

5.2 TRANSMORPHER : spécification et exécution de flux de transformations XML

Participants : Jérôme Euzenat [Correspondant], Laurent Tardif.

Pour établir ou vérifier des propriétés sur des transformations, il faut en disposer d'une représentation. Le langage XSLT, en exprimant les transformation en XML, permet de faire cela à peu de frais, mais reste verbeux et délicat à analyser. Afin de résoudre ce problème, nous avons conçu et développons en collaboration avec la société FLUXMEDIA le logiciel TRANSMORPHER qui se présente comme une sur-couche à XSLT permettant d'exprimer des flux de transformations.

TRANSMORPHER [8, 14] est un environnement permettant de définir et d'exécuter des transformations génériques sur des documents XML. Il a pour but de proposer des compléments

[BLHL01] T. BERNERS-LEE, J. HENDLER, O. LASSILA, « The semantic web », *Scientific american* 279, 5, 2001, p. 35-43, <http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>.

du langage XSLT afin de :

- décrire simplement des transformations simples (suppressions d'éléments, remplacements de noms d'attributs, assemblage de documents...);
- assembler des transformations en connectant leurs entrées et sorties (multiples);
- appliquer des transformations jusqu'à leur inapplicabilité;
- appliquer des substitutions d'expressions rationnelles sur le contenu;
- intégrer des moteurs de transformation externe.

TRANSMORPHER permet de décrire en XML des flux de transformations, c'est-à-dire des ensembles de transformations connectées par des canaux d'entrée-sortie. Les canaux véhiculent l'information à transformer en XML. Les transformations peuvent être d'autres flux de transformations ou des transformations élémentaires. TRANSMORPHER définit un ensemble d'abstractions de transformations élémentaires dotées d'un modèle d'exécution et des instantiation de ces transformations. On peut décliner ainsi les transformations élémentaires : appel externe (incluant XSLT), dispatcheur, sérialiseur, moteurs de requêtes, traducteurs, itérateurs et systèmes de règles.

Les systèmes de règles permettent de décrire des transformations dans un langage plus simple qu'XSLT. Actuellement, ce langage est exécuté en le transformant directement en une transformation XSLT (sauf pour les substitutions d'expressions rationnelles).

TRANSMORPHER est principalement un ensemble de classes Java documentées (qui peuvent être spécialisées et intégrées dans un autre logiciel) et un moteur d'exécution du système de transformations. La spécification d'un système de transformations peut se faire en programmant en Java par l'instanciation des classes ou en fournissant une spécification du système en XML.

Une extension de ce logiciel consiste à attacher des assertions à chaque transformation (élémentaire, externe ou composée) afin de signifier si une propriété est prouvée, supposée, à vérifier ou à démontrer. Ceci permettra de montrer des exemples concrets de détermination de propriétés de transformations composées.

Le système peut être obtenu depuis <http://transmorpher.inrialpes.fr>.

5.3 DLML : Description Logic Markup Language

Participant : Jérôme Euzenat [Correspondant].

DLML [11] n'est pas un langage mais un système modulaire de description de types de documents (DTD) permettant de décrire la syntaxe et la sémantique de nombreuses logiques de descriptions (voir §3.1). Une importante motivation de DLML est de pouvoir incorporer des représentations en langage formel (ici les logiques de descriptions) dans des documents XML en particulier pour l'action *ESCRIRE* (voir §8.1).

DLML tire parti de la conception modulaire des logiques de descriptions pour définir les logiques par assemblage de constructeurs élémentaires. Le système contient la spécification de la syntaxe (DTD) et de la sémantique (DSD) de plus de 40 opérateurs élémentaires. Ainsi, la définition d'une logique est donnée par la spécification, en XML, des opérateurs qu'elle autorise. À partir de cette déclaration, DLML est capable d'assembler les spécifications élémentaires pour engendrer les DTD et DSD de la logique elle-même. Plus de 25 logiques répertoriées sont décrites

sur le site.

Le langage DLML est associé à un ensemble de transformations XSLT permettant de transformer des descriptions DLML d'une logique vers une autre (soit en normalisant, soit en appauvrissant l'information), d'un langage (par exemple, de syllogismes) vers une logique de descriptions, d'afficher une description à l'aide de LaTeX ou d'importer des descriptions d'un autre système (FaCT de l'université de Manchester).

Le système peut être utilisé depuis <http://co4.inrialpes.fr/xml/dlml/>.

6 Résultats nouveaux

6.1 Interopérabilité sémantique

Participants : Jérôme Euzenat [Correspondant], Heiner Stuckenschmidt [TZI Bremen].

Dans le cadre de la communication de représentations formalisées, on cherche à garantir la préservation du sens entre deux représentations. Plus précisément, entre deux langages L et L' , une représentation r de L va être transformée en une représentation $\tau(r)$ dans L' (où $\tau : L \rightarrow L'$). On adopte donc une approche consistant à étudier l'interopérabilité sémantique via une transformation.

Assurer l'intelligibilité sémantique peut être décrit à l'aide des deux équations complémentaires :

$$\forall r \subseteq L, \forall \delta \in L, r \models_L \delta \Rightarrow \tau(r) \models_{L'} \tau(\delta) \text{ et } \forall r \subseteq L, \forall \delta \in L, \tau(r) \models_{L'} \tau(\delta) \Rightarrow r \models_L \delta$$

Garantir l'interopérabilité sémantique dans toute sa généralité est, bien entendu, hors de portée. Par conséquent, nous nous intéressons à des cas particuliers de ces équations [12].

6.1.1 Familles de langages

Dans le cas d'une famille de langages, il existe un langage $L \vee L'$ tel que toute formule de L ou de L' soit une formule de $L \vee L'$. De plus, l'interprétation des opérateurs de L sont les mêmes dans $L \vee L'$ (et par conséquent, les opérateurs présents dans les deux langages ont la même interprétation). Un cas de famille de langages bien connu est celui des logiques de descriptions où toute une hiérarchie de langages a été définie ^[Neb90].

Avec DLML (voir §5.3), nous avons décrit en XML la famille des logiques de description pour pouvoir explorer les possibilités des familles de langages. Cela permet d'envisager différents cas de figure :

- L' contient tous les constructeurs de L , dans ce cas, la traduction τ est l'identité. L'un des avantages de DLML c'est qu'il n'est effectivement nul besoin d'appliquer de transformations car les DTD sont compatibles ;
- L contient strictement plus de constructeurs que L' (il est plus expressif). Dans ce cas, la seule solution n'est pas comme on pourrait penser l'appauvrissement syntaxique. Il

[Neb90] B. NEBEL, *Reasoning and revision in hybrid representation systems, Lecture Notes in Artificial Intelligence 422*, Springer Verlag, Berlin (DE), 1990.

faut, en toute théorie, mettre en œuvre un démonstrateur pour déduire de r tout ce qui est exprimable en L' .

- Nous avons exploré le cas où l'on disposait de preuves d'équivalence entre L et L' d'un type assez particulier : pour chaque constructeur d'un langage, on montre que toute expression l'utilisant est exprimable avec les constructeurs de l'autre. On a alors la définition constructive d'une transformation que l'on peut tenter de développer en XSLT. Nous avons appliqué ce cas à l'équivalence entre les langages \mathcal{ACC} et \mathcal{ALUE} .

L'approche par famille de langages présente un cas intéressant car il permet de réaliser des transformations préservant certaines des propriétés ci-dessus aisément [13]. Mais pour aller plus loin, il faut lier transformations et preuves de leurs propriétés. La famille sera alors utilisable en toute sécurité.

6.1.2 Transformations porteuses de preuves

On a vu ci-dessus que les transformations pouvaient être bâties sur des preuves. Il est utile de pouvoir disposer de ces preuves. Ainsi, pour l'équivalence entre deux logiques de descriptions, on montre que les expressions de l'une sont exprimables en fonction des expressions des autres et que la transformation préserve les modèles (au sens de la théorie des modèles). Disposer de la description de la sémantique des langages et de la preuve d'une transformation permet d'appliquer à la transformation de représentations le principe du code porteur de preuve ("proof-carrying code" [Nec98]) et par conséquent d'être certain de disposer de la transcription de la représentation initiale dotée des propriétés annoncées.

Afin de parvenir à vérifier la preuve des propriétés sémantiques du type de (1) et (2), il est nécessaire de disposer (a) de la représentation de la transformation, (b) de la représentation de la sémantique des langages (ou spécification de patrons) et (c) de la représentation de la preuve. Ces trois éléments seront à développer afin de pouvoir communiquer et vérifier des preuves :

- Pour représenter les transformations, on s'appuie sur le langage de TRANSMORPHER décrit au §5.2. L'un des objectifs de ce langage est d'encapsuler le langage XSLT utilisé dans les transformations de manière à présenter un langage plus aisément analysable et à décomposer hiérarchiquement les transformations de telle sorte que l'on puisse réaliser des preuves à partir de « lemmes » qui seront des contraintes pour les sous-transformations.
- DSD est un langage en cours de développement de description de la sémantique (en théorie des modèles) fondée sur XSLT, XPATH et MathML. Il sera expérimenté sur plusieurs formalismes de représentation de connaissance et par des transformations entre formalismes.
- La représentation des preuves (de telle sorte que l'on puisse vérifier facilement des preuves par induction sur la structure de termes dans un premier temps) reste à développer. Une telle représentation devra tirer parti des travaux sur MathML et OpenMath mais aussi d'initiatives plus adaptées telles que OMDoc [Koh00].

[Nec98] G. NECULA, *Compiling with proofs*, thèse de doctorat, Carnegie-Mellon university, Pittsburg (PA US), 1998.

[Koh00] M. KOHLHASE, « OMDoc: an open markup format for mathematical documents », *rapport de recherche n° SR-00-02*, Universität des Saarlandes, Saarebrücken (DE), 2000, <http://www.mathweb>.

Actuellement, nous avons fait des expériences avec le langage de description de la sémantique. Celui-ci a été utilisé dans la description de DLML (où nous proposons une DSD de chaque logique construite automatiquement à partir de la sémantique des constructeurs de termes).

Nous disposons de quelques exemples de preuves (principalement d'équivalence) dans les logiques de descriptions qui seraient un très bon premier exemple de l'application de ces concepts. Nous avons aussi des exemples de transformations entre représentations hétérogènes : syllogistique et logiques de descriptions [9].

6.1.3 Composition de propriétés et vérification

L'un des buts d'EXMO est d'envisager des flux de transformations qui résultent de l'assemblage de transformations plus élémentaires (qui peuvent être vues comme des fonctions, l'assemblage étant le plus souvent la composition). Si chacune des transformations plus élémentaires est dotée d'assertions spécifiant les propriétés qu'elle satisfait, il faut construire la propriété concernant l'assemblage. Un exemple simple est la propriété de terminaison sur des entrées finies qui est préservée par la composition mais pas forcément par l'itération saturée (jusqu'à ce que la transformation ne soit plus applicable). La préservation du sens, par contre, est préservée par la composition et l'itération.

Notre but est d'inventorier à la fois les constructeurs de transformations (composition, itération, séparation des données) et les propriétés que l'on peut vouloir associer à une transformation (classement, privilèges, pertinence, niveau de granularité, traçabilité de l'information). On développe aussi une typologie des transformations (analysables par une machine) qui permette de prouver aisément certaines propriétés (principalement syntaxiques). Alors on pourra élaborer la preuve qu'un flux de transformations satisfait une propriété par assemblage. Il est aussi intéressant d'étudier l'interférence entre propriétés de différents types (par exemple, comment préserver la traçabilité face aux exigences de classement de l'information ?).

Ce travail est en cours. Il a donné lieu à la définition du langage de TRANSMORPHER (voir §5.2).

6.2 Préservation de scenarii dans les documents multimédia

Participants : Victor Dias, Jérôme Euzenat [Correspondant], Nabil Layaïda [Opéra].

La structure temporelle des scenarii multimédia est exprimée en SMIL 2.0 à l'aide d'opérateurs qui peuvent se transcrire dans les relations temporelles qualitatives entre les éléments apparaissant dans la présentation. Un modèle (au sens de la théorie des modèles) de cette description est l'ensemble, pour toutes les exécutions valides, des positions respectives des occurrences de ces éléments dans le temps. Ces modèles peuvent encore être rassemblés en classes modulo que leur description qualitative saturée (dans laquelle il n'existe qu'une seule relation qualitative entre chaque couple d'éléments) soit la même.

Lorsqu'un document multimédia doit être exécuté sur des plates-formes dotées de possibilités diverses (un téléphone ne pouvant afficher qu'une image à la fois, une borne interactive sans

clavier), il est nécessaire de le transformer afin de l'adapter au dispositif cible. Cette adaptation va, en première approximation, réduire l'ensemble des modèles du scénario.

On s'intéresse à décrire des classes de modèles que l'on voudrait préserver lors de la transformation et à établir la préservation de ces classes (qu'il y ait au moins un modèle préservé par classe).

Au-delà de la préservation de cette structure temporelle, il est souvent nécessaire de préserver la structure de parcours. Les documents multimédia possèdent une structure de navigation dite « non déterministe » (par exemple, un bouton qui, lorsqu'il est pressé, interrompt totalement la présentation en cours et passe à une autre section). Cette structure permet de naviguer dans le document. Nous nous intéressons à la classe de modèles correspondant à un parcours de cette structure et à sa préservation. En effet, si la structure navigationnelle est transformée, il est possible de se trouver avec des éléments de documents non atteignables ou des ordres de présentations non exécutables (ce qui peut être très important dans le cadre d'un jeu par exemple).

Par ailleurs, nous nous sommes rendu compte que l'évaluation des transformations en fonction de la préservation du maximum de contraintes temporelles n'était pas forcément la meilleure dans tous les cas. Il est donc nécessaire de décrire les scénarii temporels avec des primitives de plus haut niveaux (comme des structures rhétoriques).

6.3 Treillis et information

Participant : Olivier Brunet [Correspondant].

Si l'on envisage les langages que nous utilisons comme des treillis ($\langle L, \leq \rangle$), une transformation τ est une application d'un treillis dans un autre. Une transformation perdra de l'information si sa transformation adjointe (définie par le plus petit majorant des éléments qui ont pour image un élément de L') peut retourner un élément plus grand que l'élément de départ (elle n'en retournera jamais de plus petit).

On peut définir le cas particulier où les transformations sont des applications dans le même langage ($\tau : L \rightarrow L$), comme le domaine d'interprétation d'une logique modale. Dans ce cas, l'interprétation d'une formule ϕ est l'ensemble des représentations (les nœuds du treillis) dans lesquels elle est vraie et l'interprétation de la modalité K_i (connu de i , liée à une transformation τ_i) comme l'ensemble des représentations qui, une fois transformées par τ_i , satisfont la formule à connaître ($[K_i \phi] = x \in L; \tau_i(x) \in [\phi]$). Comme l'information croît de manière monotone en descendant le treillis, l'interprétation d'une formule est un ensemble clos pour \leq .

L'axiomatisation de cette structure révèle qu'il s'agit de la logique modale IS4 (qui ne satisfait ni le tiers exclus $\neg \phi \vee \neg \neg \phi$, ni l'axiome 5 $\neg K_i \phi \Rightarrow K_i \neg K_i \phi$ – du fait de la spécificité de la négation sur des sous-ensembles clos pour \leq) [7].

L'intérêt de ce travail est de permettre de poser des questions liées à la connaissance après transformation (par exemple, $\exists \phi; K_{ij} \phi \wedge \neg K_i \phi \wedge \neg K_j \phi$, est-il possible en rassemblant les connaissances de i et j de déduire des informations dont aucun des deux ne dispose, avec $\forall \phi; (K_i \phi \Rightarrow K_{ij} \phi) \wedge (K_j \phi \Rightarrow K_{ij} \phi)$).

7 Contrats industriels (nationaux, européens et internationaux)

7.1 Fluxmedia

Participants : Jérôme Euzenat, Laurent Tardif.

Le logiciel TRANSMORPHER (voir §5.2) est développé en partenariat avec la société FLUX-MEDIA incubée par l'INRIA. Le partenariat définit la contribution des deux parties au développement du système ainsi que le mode de diffusion sous forme ouverte (GNU General Public License).

8 Actions régionales, nationales et internationales

8.1 Actions nationales

8.1.1 Action de recherche coopérative ESCRIRE

Participants : Jérôme Euzenat [correspondant], Raphaël Troncy.

Il existe différents formalismes de représentation de connaissance et nul ne connaît exactement leurs qualités respectives. Le but de l'ARC ESCRIRE consiste donc à comparer trois types de représentations de connaissance (graphes conceptuels, représentations de connaissance par objets et logiques de descriptions) du point de vue de la représentation du contenu de documents et de sa manipulation. Pour cela, l'action s'appuie sur les compétences dans chacune des représentations des projets ACACIA, EXMO et ORPAILLEUR respectivement. L'objectif de l'action consiste à comparer les apports de chacun des types de représentation pour la représentation du contenu dans les serveurs de connaissances.

La mise à l'épreuve de ces différents formalismes pour le traitement d'un jeu de documents a nécessité une réflexion méthodologique sur le passage des textes à leur représentation formelle (de façon suffisamment indépendante des formalismes employés) en lien avec le type d'accès que l'on veut avoir sur ces documents [10].

On a donc défini en XML un ensemble de langages pivôts (c'est-à-dire indépendants des représentations de connaissance utilisées) pour décrire des ontologies et des documents, des requêtes et des réponses à ces requêtes. On a par ailleurs développé une interface permettant de poser graphiquement la même requête à plusieurs systèmes et de collecter les réponses à ces requêtes.

Un ensemble de requêtes défini de manière coordonnée sera évalué dans chacun des contextes. À l'issue de ce travail, les différents formalismes seront comparés entre eux (mais aussi à la recherche en texte intégral) selon un protocole prédéfini. Celui-ci devra apprécier des critères tant qualitatifs (expressivité des requêtes, accessibilité/lisibilité des informations, etc.) que quantitatifs (temps de réponse à une requête, taux de précision/rappel des réponses, etc.). Cette évaluation proposera une grille d'analyse des avantages et inconvénients d'un langage de représentation formel vis-à-vis de la recherche d'informations sur le Web.

On trouvera plus d'informations sur ESCRIRE à <http://www.inrialpes.fr/exmo/cooperation/escuire/>.

8.2 Actions européennes

8.2.1 Réseau thématique OntoWeb : ontology-based information exchange for knowledge management and electronic commerce

Participants : Jérôme Euzenat [correspondant], Heiner Stuckenschmidt.

EXMO est impliqué dans le réseau thématique ONTOWEB (“Ontology-based information exchange for knowledge management and electronic commerce”) financé par l’union européenne. Le nœud INRIA d’ONTOWEB est constitué du projet ACACIA et des actions AID, EXMO et ORPAILLEUR. Il a en charge le module dévolu à promouvoir la collaboration internationale sur les thématiques d’ONTOWEB. EXMO représente l’INRIA au sein du bureau du projet.

On trouvera plus d’informations sur ONTOWEB à <http://www.ontoweb.org>.

9 Diffusion de résultats

9.1 Animation de la communauté scientifique

- Jérôme Euzenat est membre élu du bureau de l’Association Française d’Intelligence Artificielle (AFIA).
- Comité de rédaction de la revue « L’objet » (Jérôme Euzenat).
- Comité de programme des éditions 2001 des colloques « Langages et modèles à objets », « Journées nationales sur les modèles de raisonnement », du séminaire « Ontology and information sharing » de l’IJCAI 2001 et du séminaire « Knowledge management and organizational memory » de l’IJCAI 2001 (Jérôme Euzenat).
- Membre du bureau et coordinateur du package « Promoting world-wide collaboration » du réseau thématique européen OntoWeb (voir §8.2) impliquant 70 équipes (2001-2003).
- Organisation (matérielle et scientifique) avec Isabel Cruz, Stefan Decker et Deborah McGuinness, du « 1st international semantic web working symposium (SWWS-1) », Stanford (CA US), 30 juillet-1er août 2001. 250 personnes étaient présentes sans autre annonce que l’appel à communication.
- Organisation scientifique du workshop stratégique NSF-UE sur le « web sémantique » piloté par ERCIM, Sophia-Antipolis, 3-5 octobre 2001 (Jérôme Euzenat).

9.2 Enseignement

- Coordination du séminaire du profil « Inférence et connaissance » du DEA « Informatique, systèmes et communication » (Jérôme Euzenat).

9.3 Participation à des colloques, séminaires, invitations

- Intégration sémantique d’annotations formelles : Séminaire INRIA « Web sémantique », Nancy (FR), 25-26 janvier 2001, exposé oral
- Approche sémantique et sémiologique de l’interopérabilité : Séminaire « Inférence Connaissance », Université Joseph-Fourier-INP, Grenoble (FR), 1er mars 2001, exposé oral

- What annotation scheme for textual resources? : IMPG workshop on « ontologies in molecular biology », INRIA Rhône-Alpes, Grenoble (FR), 27 avril 2001, exposé oral
- Approche sémantique et sémiologique de l'interopérabilité : Séminaire ERIC, université Lyon 2, Bron (FR), 14 mai 2001, exposé oral
- Une infrastructure pour garantir formellement l'interopérabilité dans un web sémantique hétérogène : Séminaire « Informatique fondamentale » Leibniz, INPG, Grenoble (FR), 7 juin 2001, exposé oral
- Participation au séminaire « Semantic web » (Sophia-Antipolis, 3-5 octobre 2001), exposé sur « Interoperability in an open semantic web » et coordination du groupe de travail « infrastructure » (Jérôme Euzenat).
- Séminaire ONTOWEB, Amsterdam (6-8 décembre).

10 Bibliographie

Ouvrages et articles de référence de l'équipe

- [1] R. DUCOURNAU, J. EUZENAT, G. MASINI, A. NAPOLI (éditeurs), *Langages et modèles à objets : état et perspectives de la recherche, collection "Didactique", 19*, INRIA, INRIA Rocquencourt (FR), 1998, <http://www.inria.fr/rrrt/d-019.html>.
- [2] J. EUZENAT, *Représentations de connaissance : de l'approximation à la confrontation*, Habilitation à diriger des recherches, Université Joseph Fourier, Grenoble (FR), janvier 1999.

Livres et monographies

- [3] I. CRUZ, S. DECKER, J. EUZENAT, D. MCGUINNESS (éditeurs), *Semantic web working symposium*, 2001, <http://www.semanticweb.org/SWWS/program/full/SWWSProceedings.pdf>.

Articles et chapitres de livre

- [4] F. CERBAH, J. EUZENAT, « Traceability between models and texts through terminology », *Data and knowledge engineering* 38, 1, 2001, p. 31–43.
- [5] J. EUZENAT, « Construction collaborative de bases de connaissance et de documents pour la capitalisation », in : *Ingénierie et capitalisation des connaissances*, M. Zacklad et M. Grundstein (éditeurs), Hermès Science publisher, Paris (FR), 2001, ch. 2, p. 25–48.
- [6] J. EUZENAT, « Granularity in relational formalisms with application to time and space representation », *Computational intelligence* 17, 4, 2001, p. 703–737.

Communications à des congrès, colloques, etc.

- [7] O. BRUNET, « A Model for Knowledge Representation in Distributed Systems », in : *Proc. KI workshop on Modal Logic in AI, Wien (OS)*, p. 3–12, 2001.
- [8] J. EUZENAT, L. TARDIF, « XML transformation flow processing », in : *Proc. 2nd conference on extreme markup languages, Montréal (CA)*, p. 61–72, 2001, <http://transmorpher.inrialpes.fr/paper/>.
- [9] J. EUZENAT, « An infrastructure for formally ensuring interoperability in a heterogeneous semantic web », in : *Proc. 1st international on semantic web working symposium (SWWS), Stanford (CA US)*, p. 345–360, 2001, <http://www.semanticweb.org/SWWS/program/full/paper16.pdf>.

- [10] J. EUZENAT, « L'annotation formelle de documents en (8) questions », *in : Actes 6e journées sur ingénierie des connaissances (IC), Grenoble (FR)*, J. Charlet (éditeur), Presses universitaires de Grenoble, p. 95–110, Grenoble (FR), 2001.
- [11] J. EUZENAT, « Preserving modularity in XML encoding of description logics », *in : Proc. 14th workshop on description logics (DL), Stanford (CA US)*, D. McGuinness, P. Patel-Schneider, C. Goble, R. Möller (éditeurs), p. 20–29, 2001, <http://ceur-ws.org/Vol-49/Euzenat-20start.ps>.
- [12] J. EUZENAT, « Towards a principled approach to semantic interoperability », *in : Proc. IJCAI 2001 workshop on ontology and information sharing, Seattle (WA US)*, A. Gomez Perez, M. Gruninger, H. Stuckenschmidt, M. Uschold (éditeurs), p. 19–25, 2001, <http://ceur-ws.org/Vol-47/euzenat.pdf>.
- [13] H. STUCKENSCHMIDT, J. EUZENAT, « Ontology Language Integration : A Constructive Approach », *in : Proc. IJCAI 2001 workshop on Applications of Description Logics, Wien (AT)*, 2001, <http://ceur-ws.org/Vol-44/StuckenschmidtEuzenat.ps.gz>.

Rapports de recherche et publications internes

- [14] ACTION EXMO, FLUXMEDIA, *Transmorpher 1.0*, Grenoble (FR), 2001, <http://transmorpher.inrialpes.fr/refman>.
- [15] J. EUZENAT, (COORD.), « 1st international semantic web working symposium (SWWS-1) », *Deliverable n° 7.6*, INRIA Rhône-Alpes, Montbonnot (FR), 2001.