Validation et enrichissement d'annotations : application à la veille médiatique

Olivier Carloni[*,**] , Michel Leclère[*] , Marie-laure Mugnier[*]

[*]LIRMM 161 rue Ada, 34392 Montpellier
{carloni, leclere, mugnier}@lirmm.fr
http://www.lirmm.fr
[**]Société Mondeca, 3 Cité Nollez 75018 Paris
olivier.carloni@mondeca.com,
http://www.mondeca.com

Résumé. Cet article présente un service de validation et d'enrichissement d'annotations conçu pour un outil industriel de gestion des connaissances basé sur le langage des Topic Maps (TM). Un tel service nécessitant la mise en œuvre de raisonnements sur les connaissances, nous avons été amené à doter le langage des TM d'une sémantique formelle. Ceci a été réalisé par l'intermédiaire d'une transformation des TM vers le formalisme logique des graphes conceptuels qui comme les TM dispose d'une représentation graphique des connaissances. La solution a été mise en oeuvre dans une application conçue pour la veille médiatique. Des annotations sont extraites automatiquement de dépêches sur l'actualité économique puis ajoutées à la base de connaissances. Elles sont ensuite fournies au service de validation qui décide de leur validité, les enrichit et évalue leur pertinence afin de faciliter le travail du veilleur.

1 Introduction

Depuis une dizaine d'années, la connaissance a fait une entrée massive dans les organisations industrielles et administratives, répondant aux besoins (1) de capitalisation du patrimoine immatériel des organisations que constituent les savoirs et savoirs-faire de leurs membres et (2) d'exploitation à un niveau "sémantique" des systèmes d'information des organisations, c'est-à-dire s'appuyant sur la sémantique des informations échangées. Mondeca est un éditeur de logiciels qui développe ITM (Intelligent Topic Manager) un logiciel de gestion des connaissances structurant les connaissances représentées en 3 niveaux :

1. Le niveau méta contient l'ontologie de représentation. Il comporte toutes les primitives du langage d'ITM permettant de créer les ontologies de chaque "client" de l'outil ITM.
2. Le niveau modèle contient les ontologies créées pour chaque client, c'est-à-dire les types et des propriétés générales qui seront utilisées pour décrire les instances. Plusieurs sortes d'ontologies sont généralement définies : une ontologie de domaine spécifique au client, des ontologies "de services" (primitives de spécification de thesaurus, primitives de des-

cription d'une organisation logique des documents...) permettant la mise en œuvre des différents services offerts par ITM.

3. Le niveau des instances contient divers espaces de travail dont un thesaurus et une base d'annotations sémantiques, ces instances étant construites en utilisant le vocabulaire des ontologies correspondantes.

ITM propose différents services d'acquisition et d'exploitation des connaissances de sa base :

- ajout/modification/suppression d'instances par l'intermédiaire d'interfaces à base de formulaires "dynamiquement contrôlés" par l'ontologie correspondante ;
- construction automatique d'annotations à partir de documents peu ou pas structurés à partir de techniques de traitement automatique du langage naturel (Amardeilh et al. (2005)) ;
- import/export de connaissances dans différents formats XML permettant d'intégrer des connaissances issues d'autres outils ;
- navigation par des interfaces graphiques dans le réseau de connaissances ;
- recherche d'un topic ou d'une association spécifique.

Le Lirmm et Mondeca collaborent à un projet de mise en œuvre de raisonnements sur les connaissances des bases ITM afin de permettre une validation et un enrichissement sémantique des connaissances ajoutées à la base d'annotations et d'étendre les capacités d'exploitation de la base de connaissances.

Le langage de représentation utilisé dans ITM est basé sur les "Topic Maps" (TM). Les TM sont un langage standardisé ayant pour objectif de permettre la structuration d'un ensemble de ressources (documents adressables) à l'aide de *topics* qui rassemblent des ressources partageant des propriétés communes et d'*associations* indiquant des liens entre topics (ISO/IEC :13250 (2000)). On peut voir une TM comme un graphe (ou un hypergraphe) étiqueté dont les sommets sont les topics et les associations, et les arêtes indiquent la participation d'un topic à une association (cf. Auillans et al. (2002), voir aussi section 3). Le langage des TM dispose d'une syntaxe XML (TopicMaps.Org (2001)) qui en fait un des langages candidats à une utilisation dans le cadre du web sémantique. Contrairement à ses concurrents directs, RDF/S et OWL, les TM n'ont pas été munies d'une sémantique formelle et aucun outil de raisonnement n'a été proposé pour ce langage.

Les TM ont une proximité évidente avec les représentations à base de graphes conceptuels (Sowa (1984)), une TM pouvant être naturellement traduite en un graphe conceptuel simple (SG). Nous avons donc défini des transformations des TM dans le formalisme des graphes conceptuels, et plus spécifiquement dans celui de la famille $\mathcal{SG}$ (Baget et Mugnier (2002)).Une transformation est dédiée aux TM de niveau modèle et l'autre aux TM de niveau instance.

Ces transformations dotent directement ITM des mécanismes de raisonnement de la famille $\mathcal{SG}$, qui sont basés sur des opérations de graphes tout en étant corrects et complets vis-à-vis de la déduction en logique des prédicats. De plus, elles permettent de bénéficier des algorithmes efficaces implémentés dans la plate-forme CoGITaNT de développement d'applications à base de graphes conceptuels (Genest (1997)). Enfin, ces transformations permettent d'envisager l'intégration de connaissances plus riches dans le logiciel ITM puisque la famille $\mathcal{SG}$ dispose de connaissances telles que règles, contraintes et définitions, qui n'existent pas dans ITM.

Dans cet article, nous présentons les transformations qui permettent de représenter dans le formalisme de la famille $\mathcal{SG}$ les connaissances d'ITM décrites en TM et le service de validation et d'enrichissement mis en œuvre. La figure 1 présente l'architecture générale du système

conçue pour ce projet. Le service de validation et d'enrichissement maintient une copie, enri-

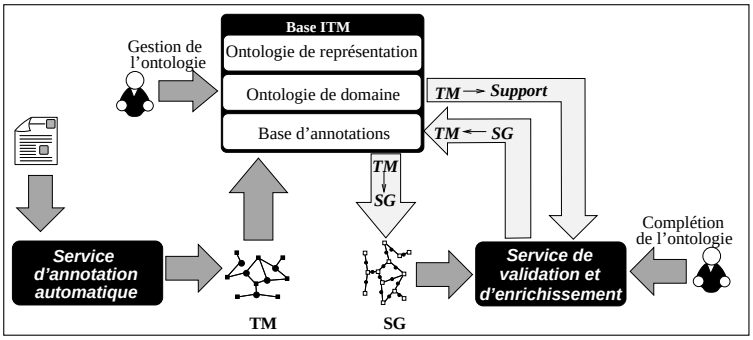

FIG. 1 – *L'architecture du service de validation et d'enrichissement.*

chie par application de règles d'inférence, d'une portion "cohérente" de la base d'annotations ITM. Le choix de la portion "cohérente" est déterminé par l'ordre d'ajout des annotations à la base ; toute nouvelle annotation rendant la base incohérente est rejetée. Ceci permet d'assurer la monotonie des raisonnements effectués sur cette base. Le système $\mathcal{SG}$ proposé repose sur un processus itératif en quatre étapes : vérification, intégration, enrichissement et validation de l'annotation. Ce système et les transformations ont été exploités dans le projet PressIndex, une application dédiée à la veille médiatique.

La veille médiatique consiste à synthétiser et discriminer, selon la pertinence, les sources et le contenu, des informations provenant de divers médias dans un secteur d'actualité précis, facilitant ainsi la prise de décision concernant ce secteur. Dans le cas de PressIndex, ces informations sont classées selon leur *nature*, c'est-à-dire selon qu'elles relèvent de la *rumeur*, de *l'annonce* officielle ou du *fait* avéré. Le cœur de l'application PressIndex de veille médiatique est une base ITM alimentée par le service d'annotation automatique (Amardeilh (2006)) et couplée à deux services $\mathcal{SG}$: le service de validation et enrichissement et un service d'évaluation de la pertinence des annotations construites. On fournit au service d'annotation des dépêches portant sur l'actualité économique. Ce dernier en extrait les principaux évènements dont il détermine la nature avant de les modéliser en TM sous forme d'annotation.

Dès qu'une de ces annotations est ajoutée à la base ITM, elle est transmise au premier service $\mathcal{SG}$ pour être validée et enrichie. En retour ce service de validation qualifie l'annotation ajoutée de "non valide" ou "valide". Dans le premier cas, le service renvoie un rapport d'erreurs explicitant le problème rencontré et permettant ainsi sa correction (suppression ou modification de l'annotation). Dans le deuxième cas, le service renvoie un ensemble de connaissances relevant du domaine veillé qui ont été déduites à partir de l'annotation.

L'annotation complétée de ces nouvelles connaissances est ensuite transmise au service d'évaluation de la pertinence qui en déduit des relations entre faits, annonces ou rumeurs susceptibles de déclencher des alertes de veille. Il s'agit plus précisément de détecter des contradictions, confirmations ou infirmations de rumeurs et d'annonces.

La suite de cet article se structure en 4 parties. La section 2 donne un aperçu de la famille $\mathcal{SG}$, langage cible des transformations des TM d'ITM. La section 3 décrit le langage des TM dans lequel sont modélisées les bases d'ITM. La section 4 décrit les transformations des TM

vers les composants $\mathcal{SG}$, ainsi que le service de validation et d'enrichissement. Enfin la section 5 présente sa mise en œuvre dans l'application de veille concurrentielle et quelques résultats expérimentaux. En conclusion sont évoqués d'autres travaux similaires à celui présenté dans cet article.

2 La famille $\mathcal{SG}$

Les graphes conceptuels, et en particulier la famille $\mathcal{SG}$ (Baget et Mugnier (2002)), nous ont paru être un "bon" formalisme de représentation de connaissances sur lequel bâtir les mécanismes de raisonnement envisagés. Différents arguments ont motivé ce choix :

- il existe une proximité évidente entre les topic maps et les graphes conceptuels simples ;
- les graphes conceptuels possèdent une sémantique formelle en logique des prédicats ;
- la famille $\mathcal{SG}$ fournit différents types de connaissances : les faits représentés par des graphes conceptuels simples, les règles d'inférence, ainsi que les contraintes ;
- les raisonnements de la famille $\mathcal{SG}$ sont basés sur des opérations de graphe. Ceci fournit potentiellement des possibilités d'explication des raisonnements à l'utilisateur puisque les raisonnements peuvent être visualisés dans le formalisme que celui-ci connaît et directement sur les connaissances qu'il a définies. Ces raisonnements sont corrects et complets par rapport à la sémantique formelle ;
- les algorithmes développés pour les graphes conceptuels, et en particulier pour la famille $\mathcal{SG}$, utilisent des techniques de combinatoire (notamment de théorie des graphes et de satisfaction de contraintes) qui les rendent particulièrement efficaces ;
- la famille $\mathcal{SG}$ est implémentée dans CoGITaNT, une API de développement d'applications à base de graphes conceptuels, utilisable librement (Genest (1997)), et bénéficie d'outils graphiques associés (TooCOM et CoGUI [1]).

Cette section présente de façon relativement informelle le formalisme utilisé. Pour plus de précisions, voir (Chein et Mugnier (1992)) pour une définition du modèle de base ou (Baget et Mugnier (2002)) pour une présentation de toute la famille $\mathcal{SG}$.

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. Le graphe G_1 de la figure 2 peut être vu comme représentant la connaissance suivante : "un grand groupe de l'agroalimentaire acquiert la société $S1$ qui contrôle la société $S2$, elle aussi spécialisée dans l'agroalimentaire". Les étiquettes des sommets sont prises dans un vocabulaire appelé *support* qui peut être plus ou moins riche. Nous considèrerons ici un support comme une structure $S = (T_C,\ T_R,\ I,\ \sigma)$. T_C est un ensemble de types de concepts et T_R est un ensemble de relations d'arité (nombre d'arguments) quelconque. T_C et T_R sont partiellement ordonnés, l'ordre partiel traduisant une relation de spécialisation ($t' \leq t$ s'interprète par "t' est une spécialisation de t"). I est un ensemble de marqueurs dits individuels. σ associe à toute relation une signature qui définit le type maximal de chacun de ses arguments.

[1] `http://sourceforge.net/projects/toocom` et `http://www.lirmm.fr/cogui`

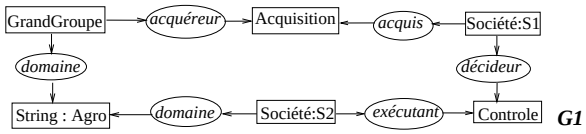

FIG. 2 – *Un exemple de SG.*

Le support peut être vu comme une ontologie rudimentaire et les SG encodent des connaissances assertionnelles (que nous appellerons des *faits*) : ils assertent l'existence d'entités et de relations entre ces entités. Un sommet concept d'un SG est étiqueté par un couple $t : m$ où t est un type de concept et m est un marqueur appartenant à I. Si le sommet représente une entité non identifiée le sommet est dit *générique*, et aucun marqueur n'est spécifié (un marqueur générique noté * est parfois représenté). Dans le graphe G_1 de la figure 2 par exemple, le sommet `[Société :S1]` désigne "la" société S1, tandis que le sommet `[GrandGroupe]` désigne "un" grand groupe. 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. De façon classique, dans les dessins, les sommets concepts sont représentés par des rectangles et les sommets relations par des ovales, et l'ordre sur les arêtes incidentes à un sommet relation n-aire par une numérotation de ces arêtes de 1 à n (ou par des flèches lorsque les relations sont binaires). 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.

2.2 Sémantique et raisonnement

La déduction sur les SG se calcule par une sorte d'appariement de graphes (un homomorphisme de graphes), appelé *projection*. Intuitivement, l'existence d'une projection de G dans H montre que la connaissance représentée dans G est contenue dans H, autrement dit que G se déduit de H, ou encore que H est une spécialisation de G. 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.Dans la figure 3 par exemple, il existe une projection de Q dans G_2 (sachant que `GrandGroupe < Société`). Les sommets colorés donnent l'image de la projection (sur cet exemple la projection est injective). Ce formalisme de base est muni d'une sémantique en logique du premier ordre par le biais d'une application notée Φ (Sowa (1984)). Un SG se traduit par une formule $\Phi(G)$ existentielle,

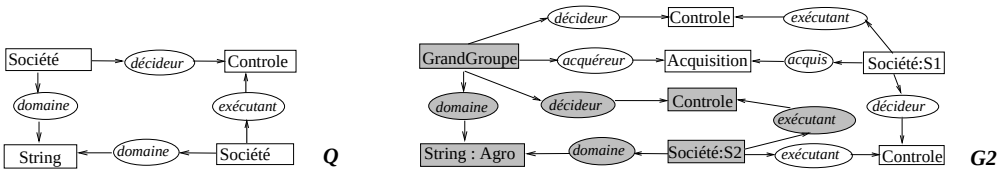

FIG. 3 – *Image de la projection de Q dans G_2*

positive et conjonctive. La formule associée au graphe G_1 de la figure 2 est par exemple : $\Phi(G_1) = \exists x \exists y \exists z GrandGroupe(x) \wedge Acquisition(y) \wedge Controle(z) \wedge Societe(S1) \wedge Societe(S2) \wedge String(Agro) \wedge acquereur(x, y) \wedge acquis(S1, y) \wedge domaine(x, Agro) \wedge domaine(S2, Agro) \wedge decideur(S1, z) \wedge executant(S2, z)$. Le résultat de correction (Sowa (1984)) et de complétude (Chein et Mugnier (1992)) de la projection établit l'équivalence entre l'existence d'une projection et la déduction logique sur les formules associées aux SG. Précisons que pour obtenir la complétude, il est nécessaire que le SG cible H soit sous une forme dite normale : tout marqueur individuel apparaît au plus une fois dans H (autrement dit, il n'existe pas deux sommets qui représentent la même entité identifiée). Considérons maintenant une base de connaissances composée d'un support et d'une base F d'assertions (ou faits). Cette base peut-être interrogée par le mécanisme de projection. Sous sa forme la plus simple, une requête, soit Q, est elle-même un SG (Figure 3 : vu comme une requête, le SG Q demande à trouver les occurrences du motif suivant "deux sociétés d'un même domaine tels que l'une contrôle l'autre"). La base répond à la requête s'il existe une projection de Q dans F, ce qui se traduit par le fait que Q se déduit de la base de connaissances.

2.3 La notion de "différence"

Deux sommets concepts différents d'un SG ne désignent pas forcément des entités différentes. Pour exprimer la différence, on peut ajouter aux SG une relation binaire particulière sur l'ensemble des sommets concepts, antiréflexive et symétrique, dont la sémantique logique est celle de $\neq$. Cette relation, notée dif, est visualisée par des liens dits de "différence" (voir par exemple la figure 5, où il existe un lien de différence entre les sommets d'étiquette `Entier` de la contrainte C^-_{MAX}). Nous faisons l'hypothèse classique du nom unique (***unique name assumption***), toute paire de sommets concepts ayant des marqueurs individuels différents appartient donc à la relation dif. La projection (soit π de G dans H) doit alors prendre en compte la différence : si $\{c_1, c_2\} \in dif_G$ alors $\{\pi(c_1), \pi(c_2)\} \in dif_H$. Dans le cas général, la projection, même ainsi étendue, n'est plus complète par rapport à la déduction en logique classique[2]. Dans le cas d'ITM, la base de faits est composée d'entités dont on sait qu'elles sont toutes différentes. Toute paire de sommets concepts du SG résultant de la transformation appartient donc à la relation dif. Ceci assure la complétude de la projection.

2.4 Les règles de SG

Au formalisme "noyau" des graphes conceptuels simples, la famille $\mathcal{SG}$ ajoute deux types de connaissance : les *règles* et les *contraintes*. Une règle d'inférence exprime une connaissance de la forme "si hypothèse alors conclusion", où hypothèse et conclusion sont deux SG. La figure 4 présente deux exemples de règles. Les pointillés lient certains sommets de l'hypothèse et de la conclusion ; ces sommets sont appelés sommets *frontières* ; les sommets de la conclusion autres que les sommets frontières apparaissent en grisé. Les règles $R1$ et $R2$ expriment des propriétés du type de concept "contrôle" : l'acquisition d'une société entraîne le contrôle de cette société ($R1$) et le contrôle s'exerce de façon "transitive" ($R2$). La sémantique logique Φ s'étend de façon naturelle aux règles. Pour la règle R_1 de la figure 4, on obtient par

[2] Elle l'est par contre si l'on considère une logique qui n'adopte pas le principe du tiers-exclu, comme la logique intuitionniste (Leclère et Mugnier (2006)).

exemple $\Phi(R_1) = \forall x \forall y \forall z\ (Societe(x) \wedge Societe(y) \wedge Acquisition(z) \wedge acquereur(x,z) \wedge acquis(y,z) \rightarrow \exists t(Controle(t) \wedge decideur(x,t) \wedge executant(t,y)))$. Une règle R s'applique

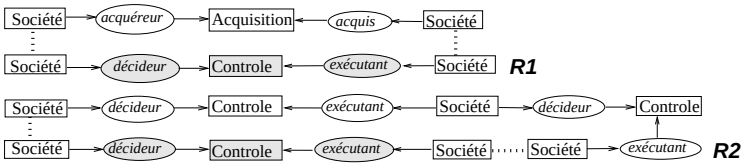

FIG. 4 – *Règles*

à un SG F s'il existe une projection de l'hypothèse de R dans F. L'application de R à F selon une telle projection π consiste à "attacher" à F la conclusion de R, en fusionnant chaque sommet frontière de la conclusion avec l'image par π du sommet frontière lui correspondant dans l'hypothèse. Exemple : en appliquant $R1$ au SG G_1 de la figure 2 (avec `GrandGroupe` $<$ `Société`) puis $R2$ au graphe obtenu, on obtient le SG G_2 de la figure 3. Etant donné un fait F et un ensemble de règles $\mathcal{R}$, on dit qu'un SG F' se dérive de F par $\mathcal{R}$ s'il existe une séquence d'applications de règles de $\mathcal{R}$ menant de F à F'. Une base de connaissances K est maintenant composée du support, d'un ensemble de règles $\mathcal{R}$ et d'un ensemble de faits F. Le mécanisme d'interrogation doit donc prendre en compte la connaissance implicite encodée dans les règles. La base de connaissances répond à une requête Q s'il existe un SG F' dérivé de F (par les règles de $\mathcal{R}$) tel que Q se projette dans F'. Considérons par exemple les figures 2, 3 et 4 : la base formée du SG G_1 et des règles $R1$ et $R2$ répond à la requête Q ; en effet, Q se projette dans G_2 dérivé de G_1 par les règles. On dispose de mécanismes de chaînage avant et de chaînage arrière adéquats et complets par rapport à la déduction logique (Salvat et Mugnier (1996)). Précisons que le problème d'interrogation n'est plus décidable (autrement dit il n'existe pas d'algorithme qui résolve ce problème en un temps fini quelles que soient les données) si l'on ne restreint pas la forme des règles. On peut notamment se restreindre à des règles permettant de "saturer"[3] la base de faits (c'est-à-dire d'appliquer les règles de toutes les façons possibles en un temps fini) ; répondre à une requête consiste alors à rechercher les projections de la requête dans la base saturée.

2.5 Les contraintes de SG

Une contrainte de SG peut être soit négative soit positive. Une contrainte *négative*, représente une interdiction, ou incohérence et a la même forme syntaxique qu'un SG, mais une sémantique intuitive différente. Elle exprime qu'un certain motif "ne doit pas être trouvé". La contrainte C^{-}_{MAX} de la figure 5 par exemple exprime l'interdiction pour une société d'avoir "deux capitaux" ; on peut la voir comme une contrainte fonctionnelle ("une société a au plus un capital"). Un SG G satisfait une contrainte négative C s'il n'existe pas de projection de C dans G. En d'autres termes, G satisfait C si C ne se déduit pas de G.

Les contraintes *positives* permettent de représenter des connaissances obligatoires. Ces dernières ont la même forme syntaxique que les règles, à ceci près que la partie hypothèse est nommée *condition* et la partie conclusion *obligation*. Elles expriment que le motif *obligation*

[3] Baget et Salvat (2006) propose une condition suffisante pour tester qu'un ensemble de règles est à saturation finie

"doit être trouvé" si le motif *condition* l'a été au préalable. La contrainte C^+_{MIN} de la figure 5 par exemple exprime l'obligation pour une société d'avoir "au moins un capital". Un SG G satisfait une contrainte positive C si pour toute projection π dans G de la partie *condition* de C il existe une projection π' de la partie *obligation* telle que pour tout sommet frontière s $\pi(s) = \pi'(s)$.

G est dit cohérent relativement à un ensemble de contraintes $\mathcal{C}$ s'il satisfait toutes les contraintes de $\mathcal{C}$. Lorsque la base de connaissances comporte non seulement des faits mais aussi des règles, ces règles doivent être prises en compte dans la notion de cohérence. Une base K composée d'un support, d'un ensemble de faits F, d'un ensemble de règles $\mathcal{R}$ et d'un ensemble de contraintes $\mathcal{C}$ est dite cohérente si tout SG dérivable de F par $\mathcal{R}$ est cohérent relativement à $\mathcal{C}$. Le problème d'interrogation n'est défini que pour une base cohérente. Notons que pour vérifier la cohérence d'un SG relativement à un ensemble de contraintes positives on est confronté à certaines subtilités qui ne sont pas abordées dans cet article (voir Baget et Mugnier (2002) pour plus d'informations).

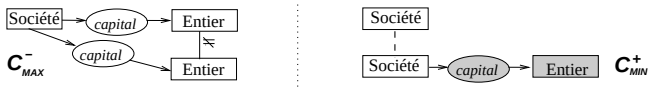

FIG. 5 – *Exemple de contraintes négative et positive.*

3 Les bases ITM

Les connaissances des bases ITM sont représentées en Topic Maps. Une TM est composée de *topics*, représentant des entités du domaine modélisé, et d'*associations* qui relient les topics entre eux. Les topics sont instances de classes. Ils peuvent être identifiés par des *noms* et caractérisés par des *occurrences* qui sont utilisées dans ITM comme des couples attribut-valeur décrivant le topic. Les associations sont typées et identifient le *rôle* joué par chaque topic dans l'association. Deux associations spécifiques sont normalisées pour toute TM : l'association *superclasse-sousclasse* et l'association *classe-instance* (représentée comme étiquette de topic pour faciliter la lecture dans cet article).

Une TM est donc un graphe biparti étiqueté $tm = (T, A, E, type, nom)$ composé de sommets topics T et de sommets associations A. Les arêtes $E \subseteq A \times T$ indiquent la participation des topics aux associations. $type$ est une fonction d'étiquetage de $T \cup A \cup E$ dans L_T où L_T est un ensemble de libellés identifiant des topics. La fonction $type$ associe à chaque sommet et arête un libellé de L_T spécifiant selon le cas la classe (d'un topic), le type (d'une association) et le rôle (d'un lien de participation). La fonction partielle nom de T dans L_T permet d'identifier certains topics par un nom. La figure 6 présente une TM décrivant un topic de classe `Société` et de nom `Oracle` qui joue le rôle `acquéreur` dans une association de type `acquisition` la reliant au topic de nom `Innobase`.

On peut compléter la description d'un topic par un ensemble d'occurrences qui indiquent une information pertinente pour le topic. Chaque occurrence est composée d'une valeur v, d'un type de données d (appelé son *type physique* dans ITM), et du nom l du topic qui type

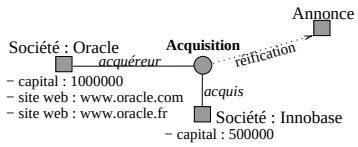

FIG. 6 – *Une TM.*

l'occurrence (appelé *type logique* dans ITM). On dispose donc d'une fonction occ de T dans $2^{V \times D \times L_T}$ qui associe à un topic l'ensemble de ses occurrences, où D est l'ensemble des types de données, et V l'ensemble des valeurs des types de D.

ITM offre la possibilité d'attribuer des occurrences aux associations, ce qui n'est pas permis dans la norme TM (ISO/IEC :13250 (2000)). Afin de rester conforme aux représentations autorisées par ITM, le domaine de occ est étendu à l'ensemble A des associations.

La TM de la figure 6 associe au topic `Oracle` trois occurrences : la valeur `1000000` de type `entier` comme `capital`, et les valeurs `www.oracle.com` et `www.oracle.fr` de type `URL` comme `site web`. Ces occurrences peuvent s'interpréter comme le capital de la société Oracle et les adresses web permettant de la contacter. Les occurrences de type physique `pointeur` ont la particularité de permettre la création de relations "directes" entre topics (i.e. sans créer d'association) ou entre une association et un topic. On peut ainsi relier des topics d'espaces différents (par exemple, associer un terme du thesaurus à une ressource de l'espace d'annotation), exprimer une "modalité"...

Une base de connaissances ITM forme une TM structurée en trois niveaux par l'association *classe-instance* (un exemple est donné en figure 7) :

- le niveau méta décrit le méta-modèle réflexif des connaissances d'ITM ;
- le niveau modèle comprend différents modèles utilisés dans une base ITM, en particulier l'ontologie de domaine spécifiant la sémantique du vocabulaire utilisé pour décrire (au niveau sémantique) la base d'annotations ;
- le niveau instance contient différents espaces de travail dont la base d'annotations "contrôlées" par l'ontologie de domaine du niveau supérieur.

Les connaissances de la TM de niveau méta sont interprétées pour doter les bases de connaissances ITM de la sémantique opérationnelle suivante[4] :

- une association a de type `type d'association permis` reliant trois topics t_c, t_a et t_r spécifie qu'il est permis au niveau inférieur à un topic de classe t_c de jouer le rôle t_r dans l'association de type t_a. Si on spécifie une occurrence ayant comme valeur un entier n de type logique `cardinalité maximum` (resp. `cardinalité minimum`) sur l'association a, cela signifie qu'une association de type t_a doit admettre au maximum (resp. minimum) n rôle(s) de type t_r.
- une association a de type `type d'occurrence permis` reliant t_c et t_o permet à un topic de classe t_c de posséder une occurrence de type logique t_o. On peut spécifier sur a une occurrence de type `cardinalité maximum` (resp. `cardinalité minimum`) et de valeur n qui signifie qu'un topic de classe t_c doit comporter au maximum (resp. minimum) n occurrence(s) de type t_o.

[4]Toutes les opérations effectuées dans ITM s'appuient sur cette sémantique opérationnelle.

– une association de type `a pour type physique` reliant t_o et t_p spécifie que toute occurrence de type logique t_o a pour type physique t_p ce qui permet de contraindre la valeur de l'occurrence (donnée numérique, textuelle, format date, pointeur) et de l'interpréter correctement ;
– l'association de type `classe-sousclasse` définit un ordre partiel sur les topics et permet l'héritage des `types d'association permis`, des `types d'occurrence permis` et de leurs cardinalités respectives.

Par ailleurs, en utilisant un type particulier d'associations nommé `type d'association orientée`, ITM permet d'exprimer au niveau ontologique que certaines associations sont hiérarchiques, c'est-à-dire doivent être interprétées comme transitives et antisymétriques. Un topic t_a de type `type d'association orientée` est un type d'association conventionnel relié par `type d'association orientée permis` à deux topics t_{r_p} et t_{r_s} jouant un rôle de type soit `prédécesseur`, soit `successeur`, ainsi qu'à un topic c_t représentant la classe du topic t à qui il est permis de jouer les rôles de type t_{r_p} (interprété comme prédecesseur) ou t_{r_s} (interprété comme successeur) dans une association de type t_a au niveau instance.

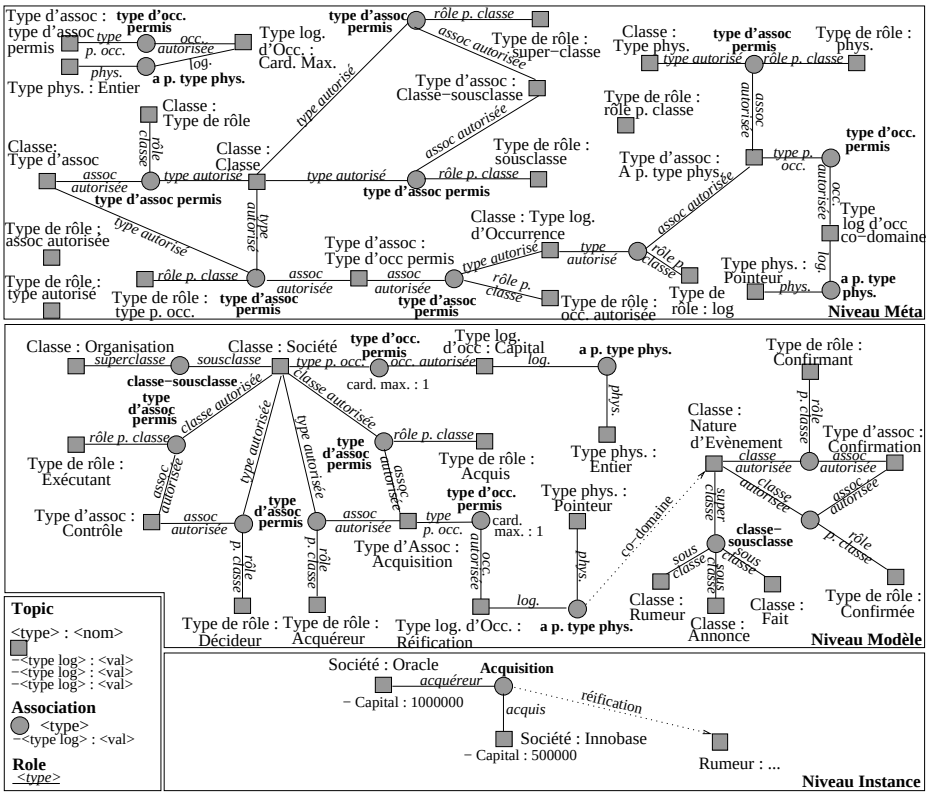

FIG. 7 – *Un extrait de base ITM avec son niveau méta, modèle et instance.*

4 Le service de validation et d'enrichissement

Le service de validation et d'enrichissement repose sur une base de connaissances $\mathcal{SG}$ représentant une portion "cohérente" des bases ITM étendue à chaque ajout "valide" de connaissances dans ITM et complétée par applications de règles d'inférence traduisant la sémantique du domaine modélisé. Cette base de connaissances $\mathcal{SG}$ est obtenue à partir des bases ITM par l'utilisation de transformations du modèle des Topic Maps vers le formalisme de la famille $\mathcal{SG}$. Plusieurs transformations ont été définies afin de prendre en compte les différents niveaux de connaissance représentés dans les bases ITM. D'autre part, outre ces connaissances en provenance d'ITM, la base $\mathcal{SG}$ peut être enrichie par des règles et contraintes utiles au domaine modélisé mais non prises en charge par ITM de manière standard.

La base de connaissances $\mathcal{SG}$ contient plus précisément un *support* issu du niveau modèle d'ITM ; un ensemble de *règles d'inférence* et un ensemble de *contraintes*, issues d'une part du niveau modèle d'ITM et d'autre part spécifiées par l'expert du domaine et enfin, une base d'*assertions* issue du niveau instance d'ITM et enrichie par les règles d'inférence.

Cette section décrit l'obtention du support, des annotations, puis des différentes catégories de règles et contraintes formalisant la sémantique opérationnelle des bases ITM. Elle présente ensuite le mécanisme d'ajout contrôlé mis en œuvre par le service de validation et d'enrichissement.

4.1 Obtention d'un support, de règles et de contraintes à partir d'une TM modèle

La fonction f_{TM2S} permet d'obtenir un support à partir du niveau modèle d'ITM (les exemples sont relatifs à la figure 7) ; tandis que les fonctions f_{MAX2C^-} et f_{MIN2C^+} interprètent les cardinalités maximum et minimum de l'ontologie, respectivement en contraintes négatives et positives ; et la fonction f_{H2RC^-} la nature hiérarchique d'un type d'association de l'ontologie en une règle et une contrainte négative.

Trois types de concept C, TA et TPO sont créés comme sous-type d'un type universel $\top$. Ces sous-types représentant respectivement le supertype des classes de topic, types d'association et types physiques d'occurrence. Tous les topics t dont le $type$ est `classe`, `type d'association` ou `type physique` deviennent respectivement par f_{TM2S} des types de concepts sous-types de C, TA et TPO et sont identifiés par $nom(t)$. L'ordre partiel sur T_C est induit par celui de l'association `classe-sousclasse`. Par exemple, les topics `Société` et `Organisation` (du niveau modèle) deviennent par f_{TM2S} des éléments de T_C ordonnés de la manière suivante *Société* $\leq$ *Organisation* $\leq C$.

Trois relations TR, TLO, et P ayant respectivement pour signature $\sigma(TR) = (C, TA)$, $\sigma(TLO) = (\top, TPO)$ et $\sigma(P) = (\top, C)$ sont créées pour représenter respectivement les supertypes des rôles, des types logiques d'occurrence et des types de pointeurs.

Tous les topics tr dont le $type$ est `type de rôle` qui participent à une association tap ternaire de $type$ `type d'association permis` avec des topics c de $type$ `classe` et ta de $type$ `type d'association` deviennent par f_{TM2S} des relations identifiées par $nom(tr)$ sous-relations de TR et ont pour signature $\sigma(nom(tr)) = (f_{TM2S}(c), f_{TM2S}(ta))$.

Par exemple, le topic (du niveau modèle) `acquis` relié par `type d'association permis` à `Société` et `Acquisition` devient par f_{TM2S} une relation *acquis* mise sous TR et ayant pour signature *σ(acquis)=(Société, Acquisition)*.

Si tap possède une occurrence (m, tpo, tlo) de type tlo égal à `cardinalité maximum` et de valeur m on lui associe par f_{MAX2C^-} une contrainte négative SG G représentant $m+1$ concepts génériques de type $f_{TM2S}(c)$ chacun d'entre eux reliés aux autres par un lien de "différence" et par une relation de type $f_{TM2S}(tr)$ à un unique sommet concept générique de type $f_{TM2S}(ta)$. De la même façon, si tap possède une `cardinalité minimum` m, $f_{MIN2C^+}(tap)$ produit une contrainte positive SG dont la partie *obligation* est identique à G, à ceci près qu'elle comporte m sommets concepts génériques de type $f_{TM2S}(c)$ et le même nombre de sommets relations de type $f_{TM2S}(tr)$. La partie *condition* ne contient que le sommet de type $f_{TM2S}(ta)$ en tant que sommet frontière.

Tous les topics tlo dont le $type$ est `type logique d'occurrence` sont reliés par une association ptp de $type$ `a pour type physique` à un topic tpo et par une association tp de $type$ `type d'occurrence permis` à un topic x ; ceux pour lesquels tpo est différent de `pointeur` deviennent par f_{TM2S} des relations identifiées par $nom(tlo)$ sous-relations de TLO et ont pour signature $\sigma(nom(tlo)) = (f_{TM2S}(x), f_{TM2S}(tpo))$. Pour ceux dont tpo est égal à `pointeur`, tp possède une occurrence de valeur nom_{co} et de type logique `co-domaine` ; ils engendrent par f_{TM2S} des relations $nom(tlo)$ sous-relations de P et de signature $\sigma(nom(tlo)) = (f_{TM2S}(x), f_{TM2S}(t_{codom}))$, avec t_{codom} le topic de nom nom_{co}.

Si ptp possède une cardinalité maximum (resp. cardinalité minimum) de valeur m, on produit par f_{MAX2C^-} (resp. par f_{MIN2C^+}) une contrainte négative SG (resp. une contrainte positive SG) selon laquelle un sommet concept de type $f_{TM2S}(x)$ ne doit pas être (resp. doit être) relié à $m+1$ (resp. m) sommets concepts génériques différents de type $f_{TM2S}(tpo)$ par $m+1$ (resp. m) relations de type $f_{TM2S}(tlo)$.

La figure 5 montre les contraintes négative et positive obtenues par les transformations f_{MAX2C^-} et f_{MIN2C^+} sur une cardinalité maximum et minimum signifiant dans ITM que toute instance de la classe `Société` a une et une seule occurrence de valeur entière et de type logique `Capital`.

La figure 8 représente une partie de la hiérarchie des types de concept et des types de relation fournie par la transformation f_{TM2S} appliquée sur la TM du niveau modèle.

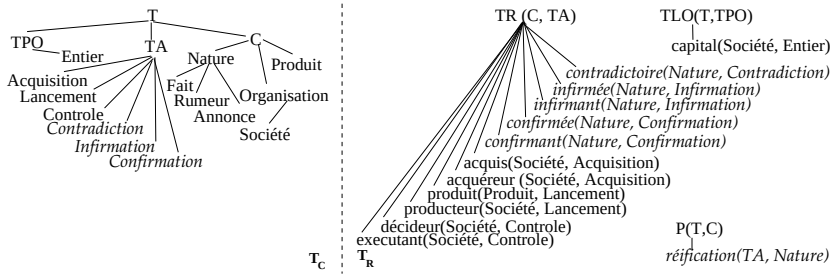

FIG. 8 – *Un extrait du support provenant de la transformation f_{TM2S} appliquée sur l'ontologie d'ITM.*

La transformation f_{H2RC^-} associe à un topic t_a de type `type d'association orientée` une contrainte SG interdisant l'apparition de circuits de longueur un pour l'association

de type t_a et une règle SG exprimant la propriété de transitivité de t_a. La règle jumelée à la contrainte évite à de telles associations de former un cycle.

La figure 9 présente une règle R et une contrainte négative C^- formalisant la nature hiérarchique d'une association de type `Contrôle` et de nature `Fait`.

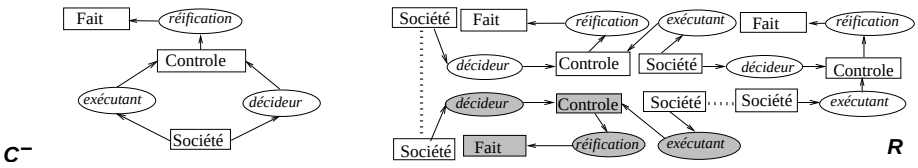

FIG. 9 – *Une règle et une contrainte négative obtenues à partir de* f_{H2RC^-}.

4.2 Obtention d'un SG à partir d'une TM annotation

La fonction f_{TM2SG} permet à partir d'une TM annotation issue de la base ITM d'obtenir un SG à ajouter à la base de connaissances $\mathcal{SG}$ (les exemples sont relatifs à la TM du niveau instance de la figure 6 transformée en le SG de la figure 10).

1. Tous les topics t dont le nom est spécifié sont transformés par f_{TM2SG} en un sommet concept individuel étiqueté $type(t) : nom(t)$; de plus $nom(t)$ est inséré dans l'ensemble des marqueurs individuels I. Par exemple, le topic `Oracle` de $type$ `Société` est transformé par f_{TM2SG} en le sommet concept individuel `Société :Oracle`.

2. Tous les topics t n'ayant pas de nom spécifié (resp. les associations a) sont transformés par f_{TM2SG} en un sommet concept générique étiqueté $type(t)$ (resp. $type(a)$). Par exemple, l'association de $type$ `acquisition` est transformée par f_{TM2SG} en le sommet concept générique `Acquisition`.

3. Chaque arête e reliant une association a à un topic t est transformée par f_{TM2SG} en un sommet relation d'étiquette $type(e)$ dont les deux arêtes incidentes étiquetées resp. 1 et 2 le relient resp. à $f_{TM2SG}(t)$ et $f_{TM2SG}(a)$. Par exemple, l'arête portant le rôle `acquéreur` est transformée par f_{TM2SG} en le sommet relation `acquéreur` le reliant aux deux sommets concepts précédents.

4. Chaque occurrence $o = (v, tpo, tlo)$ d'un topic ou d'une association x engendre par f_{TM2SG} un sommet relation d'étiquette tlo dont les deux arêtes le relient resp. au sommet concept $f_{TM2SG}(x)$ et à un nouveau sommet concept individuel étiqueté $tpo : v$. Dans le cas où tpo est égal à `pointeur`, la seconde arête est reliée au sommet concept $f_{TM2SG}(y)$, avec y le topic dont $nom(y) = v$. Par exemple, l'occurrence `1000000` du topic `Oracle` de type logique `capital` et de type physique `Entier` devient par f_{TM2SG} un sommet concept d'étiquette `Entier :1000000` relié au sommet concept `Société :Oracle` par un sommet relation `capital`.

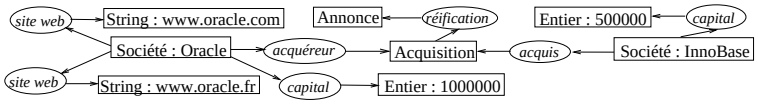

FIG. 10 – *Le SG obtenu à partir de la TM de la figure 6.*

4.3 Le mécanisme d'ajout contrôlé

La base de connaissances $\mathcal{SG}$ contient : un support S issu de la transformation f_{TM2S} a partir duquel sont définis un graphe G, un ensemble R de règles et deux ensembles C^- et C^+ de contraintes négatives et positives tels que pour chaque SG X définissant l'une de ces règles ou contraintes, il soit possible de contruire une TM dont X est l'image par f_{TM2SG}. Le graphe G représente la partie valide et enrichie d'une portion "cohérente" de la base d'annotations ITM.

À l'initialisation, le graphe G est vide. Il est modifié par ajouts successifs d'annotations issues d'ITM par le mécanisme d'ajout contrôlé. Après chaque ajout d'annotations ce graphe G conserve les propriétés suivantes :

- les types qu'il utilise sont définis dans le support ;
- les relations respectent leur signature ;
- il est sous forme normale ;
- il est saturé par rapport à l'ensemble de toutes les règles ;
- il ne viole aucune contrainte négative.

Chaque annotation ajoutée dans ITM est transformée par la fonction f_{TM2SG} en un SG A. Le graphe A obtenu est sous forme normale (les marqueurs individuels sont les identifiants de topics d'ITM) et il ne contient que des sommets concept et relation dont les types sont définis dans le support issu de la TM modèle de la même base ITM.

Etant donnés une base de connaissances $B = (S, R, C^+, C^- , G)$ et un nouveau graphe A construit sur le même vocabulaire, l'ajout contrôlé consiste à enchaîner 4 étapes : (1) le contrôle de A, (2) l'intégration de A à G, (3) l'enrichissement par R et (4) la validation de l'ajout.

A la fin de chaque étape 1, 2 et 3, on déclenche la vérification des contraintes négatives de $\mathcal{C}^-$. Le déclenchement des étapes 2, 3, et 4 est conditionné au résultat des étapes précédentes. À l'issue de l'étape 3 du mécanisme d'ajout contrôlé, si aucune contrainte négative n'a été violée, la modification de la base est définitive et une TM correspondant à la connaissance inférée à l'étape 3 est retournée à ITM, sinon on revient au graphe G. La validation du graphe G' permet de signaler à ITM un manque de connaissances sur certains topic ou association mais ne remet pas en cause la "cohérence" de la nouvelle base.

En cas de viol d'une contrainte un rapport d'erreur est envoyé à ITM contenant une TM correspondant à la portion de la base violant la contrainte (l'image de la projection de la contrainte dans le cas des contraintes négatives, ou l'image de la projection de la partie condition de la contrainte dans le cas des contraintes positives). Les TM retournées à ITM sont obtenues par application inverse de la transformation f_{TM2SG}.

Les différentes étapes sont détaillées ci-après.

4.3.1 Contrôle de l'annotation

On commence par vérifier que les relations du graphe A respectent les signatures définies dans S. Pour chaque relation non valide, un message d'erreur accompagné du graphe composé du sommet relation et de ses voisins est retourné à ITM et le mécanisme s'arrête. On vérifie ensuite que A respecte les contraintes négatives, c'est-à-dire qu'aucune contrainte de C^- ne se projette dans A. Si tel n'est pas le cas, les images des contraintes violées sont retournées à ITM et le mécanisme s'arrête.

4.3.2 Insertion de l'annotation dans la base

On ajoute ensuite A au graphe G en fusionnant les sommets concepts qui désignent des entités identiques (même marqueur individuel). Puis on contrôle qu'aucune contrainte négative de C^- ne se projette dans la base sur une partie du sous-graphe induit par les sommets non fusionnés de A (i.e. on ne cherche que des violations dûes à l'ajout de A à G puisque G est supposé valide avant l'ajout). Si une contrainte est violée le mécanisme s'arrête, on restaure G et les images des contraintes violées sont retournées à ITM.

4.3.3 Enrichissement

La portion ajoutée à la base est enrichie par saturation à partir des règles de R. Cette saturation est faite de façon incrémentale : pour toute règle r, on s'intéresse aux projections π de l'hypothèse de r dans la base qui projettent au moins un sommet de l'hypothèse sur la portion ajoutée. Pour chacune de ces projections, on vérifie ensuite que l'application de la règle selon cette projection apporte effectivement de l'information, c'est-à-dire qu'on s'assure qu'il n'existe pas de projection de la conclusion de r dans la base qui projette chaque sommet frontière sur son image par π. Si tel est le cas, on raccorde la conclusion aux images par π des sommets frontières. On réitère ce processus, en considérant que la portion ajoutée est maintenant celle formée par les nouveaux sommets inférés et ceci jusqu'à ce que cette portion soit vide (ce qui se produit forcément au bout d'un temps fini si les règles sont à saturation finie - cf section 2).

On contrôle ensuite que l'enrichissement n'a pas conduit à violer une des contraintes négatives de C^-. Comme précédemment, si une contrainte est violée , on restaure G et on retourne un message à ITM.

4.3.4 Validation

Finalement on vérifie le respect des contraintes positives de C^+. Cette dernière vérification est faite en fin de processus car l'enrichissement sémantique a pu "réparer" une absence initiale de connaissances qui aurait pu conduire au viol d'une contrainte positive. En cas de viol de contraintes positives, on signale à ITM les portions violées mais on ne restaure pas le graphe G. Les contraintes positives sont vues comme des déclencheurs d'avertissement de connaissances incomplètes et non comme des "incohérences". Cela laisse la possibilité de "compléter" lors de l'insertion d'une nouvelle annotation les connaissances manquantes.

5 L'application de veille médiatique

La veille médiatique est un service très prisé dans les secteurs politiques, industriels ou commerciaux. Dans le cas du projet PressIndex, il s'agit d'une veille concurrentielle ciblant les secteurs du commerce et de l'économie afin de récolter des informations sur l'environnement concurrentiel des sociétés. Elle porte sur plus de 9500 sources médiatiques. L'objectif est d'automatiser deux tâches du travail du veilleur. La première consiste à identifier les faits, les rumeurs et les annonces officielles extraits des dépêches provenant des grandes agences de presse (telles que l'AFP, Reuters, etc). La seconde consiste à évaluer la pertinence des informations extraites en repérant des confirmations, des infirmations ou des contradictions de rumeurs et d'annonces. Les indications obtenues permettent ensuite d'analyser à partir des faits l'environnement concurrentiel d'une société ; ou d'anticiper son évolution à partir des annonces et des rumeurs.

La masse et le débit d'informations visés par ce projet nécessite d'utiliser un mécanisme d'annotation automatique et de disposer de délais de traitement adaptés. L'outil ITM répond à ces besoins grâce à son infrastructure de stockage massif de différents types de connaissances, son service d'annotation automatique de documents basé sur des techniques de traitement automatique du langage naturel et son potentiel de mise en œuvre de raisonnements basé sur les transformations vers le formalisme $\mathcal{SG}$ qui permet d'envisager une validation et un enrichissement des annotations construites.

Le cœur de l'application PressIndex de veille médiatique est donc une base ITM alimentée par le service d'annotation automatique de dépêches relevant du secteur veillé (Amardeilh (2006)) couplée au service $\mathcal{SG}$ de validation et d'enrichissement des annotations construites. Sur cette base on vient greffer un second service $\mathcal{SG}$: le service d'évaluation de la pertinence.

La première tâche du veilleur qui consiste à identifier les faits, les annonces ou les rumeurs dans les dépêches est réalisée par le service d'annotation. Ce service utilise une "cartouche linguistique" qui établit les motifs syntaxiques intéressants à repérer. Cette cartouche a été conçue par une société spécialisée à partir de l'expertise des différents types de dépêches relevant du secteur veillé. Les motifs syntaxiques de la cartouche sont définis conformément à un vocabulaire structuré qu'il est nécessaire de mettre en correspondance avec celui d'ITM pour pouvoir représenter dans la base les informations extraites. Pour chaque dépêche, le service d'annotation produit une TM et attribue à chaque association une nature *rumeur*, *annonce* ou *fait* en lui affectant un `pointeur` vers le topic indiquant sa nature. Par exemple, la TM de la figure 6 présente une association `Acquisition` dont la nature est *annonce*.

Le service $\mathcal{SG}$ de validation et d'enrichissement (voir partie 4) permet de contrôler la cohérence des faits et de déduire de nouvelles connaissances à partir de celles extraites par le service d'annotation. Il est défini à partir des ensembles de règles R_A et de contraintes ${C_A}^+$ et ${C_A}^-$, et comme indiqué en partie 4.3, le support S à partir duquel est défini sa base est issu de la transformation f_{TM2S}.

La seconde tâche du veilleur qui consiste à détecter des confirmations, infirmations ou contradictions de rumeurs et d'annonces est accomplie par le service $\mathcal{SG}$ d'évaluation de la pertinence. Ce service est un mécanisme d'ajout contrôlé (voir partie 4.3) défini à partir du support et de la base du service de validation et d'enrichissement. L'ensemble des règles qui lui est assigné est noté R_P et ses ensembles de contraintes sont vides.

Le support à partir duquel est définie la base des deux services $\mathcal{SG}$ est divisé en deux parties :

– le *vocabulaire d'annotation* : il regroupe les types d'entités et d'événements du domaine veillé à partir desquels sont définies les annotations traduites en SG. Dans le support de la figure 8 les types de ce vocabulaire ne sont pas en italique.
– le *vocabulaire d'évaluation de la pertinence* : il définit de manière permanente des relations (confirmation, infirmation, contradiction) permettant de qualifier la pertinence des connaissances extraites. Dans la figure 8, il est représenté en italique.

Les ensembles de règles R_A et de contraintes ${C_A}^+$ et ${C_A}^-$ sont associés au vocabulaire d'annotation. Les conclusions des règles et les contraintes dont un exemple est présenté à la figure 9 ne sont définies qu'avec les types du vocabulaire d'annotation. Ces ensembles sont obtenus à partir des fonctions f_{H2RC^-}, f_{MAX2C^-} et f_{MIN2C^+}, et peuvent être complétés par un expert du domaine via l'interface CoGUI.

L'ensemble R_P des règles d'évaluation de la pertinence est associé au vocabulaire de la pertinence. Il rassemble deux genres de règles différentes manuellement crées par un expert du domaine : (1) les règles assurant la détection de contradictions entre annonces ou entre rumeurs et (2) celles permettant de confirmer/infirmer des rumeurs ou des annonces à partir des faits. La conclusion de ces règles est exclusivement formée de sommets dont les types appartiennent au *vocabulaire de la pertinence*. Ces règles ne peuvent donc pas créer de violations parmi les contraintes de ${C_A}^-$ et ${C_A}^+$.

La détection de contradictions et d'infirmations est réalisée par inférence et non par vérification de contraintes négatives car des configurations singulières qui n'en sont pas pour autant absurdes, telles que "l'infirmation par un fait d'une annonce qui confirmait une rumeur" peuvent intéresser le veilleur si, par exemple, il cherche à déterminer le plus rapidement possible l'événement qui crée la surprise, et permet de prendre à contre-pied les concurrents qui s'attendaient fortement à ce que se produise celui décrit dans la rumeur appuyée par l'annonce. Or, utiliser la violation d'une contrainte négative pour détecter une telle configuration aboutit à un rejet de l'annotation (voir 4.3) qui prive le veilleur de ces indications et supprime de la base des connaissances qui auraient pu conduire à de nouvelles inférences.

Formalisées en SG, les règles de détection de contradictions entre annonces (resp. rumeurs) produisent un sommet concept de type `Contradiction` relié par deux relations `contradictoires` à deux concepts réifiant des annonces (resp. rumeurs). Par exemple, l'existence d'une acquisition de la société A par la société B et d'une acquisition opposée de B par A est considérée comme contradictoire et déclenche ce type de règle.

Une annonce ou rumeur peut-être confirmée (resp. infirmée) par un fait. Pour cela il faut que les deux portent sur le même type d'association et concernent les mêmes entités. Formalisées en SG, la conclusion de telles règles produit un sommet concept de type `Confirmation` (resp. `Infirmation`) relié par deux relations `confirmant` (resp. `infirmant`) `confirmée` (resp. `infirmée`) à un concept de type `Annonce` (ou `Rumeur`) et un autre de type `Fait`.

En plus d'être confirmée (resp. infirmée) par un fait, une rumeur peut l'être par une annonce. Dans ce cas le sommet concept confirmant (resp. infirmant) est de type `Annonce`.

La figure 11 montre un exemple de règle de détection d'infirmation de l'annonce du lancement d'un produit par une société, par le fait que ce produit est mis en vente par une autre société, suivi d'une règle de confirmation d'une rumeur d'acquisition de société par une annonce équivalente.

L'ensemble de l'application fonctionne de la manière suivante :

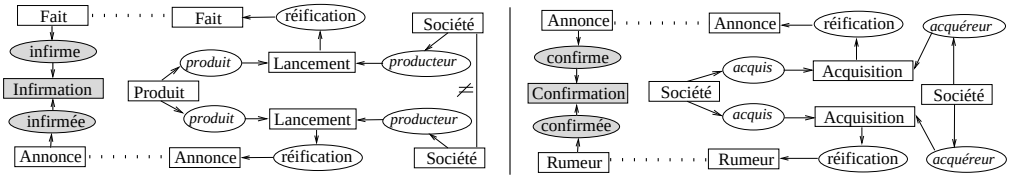

FIG. 11 – *Une règle infirmant une annonce à partir d'un fait et une règle qui confirme une rumeur à partir d'une annonce.*

Lorsque le service d'annotation crée une nouvelle annotation, cette dernière est transférée au service de validation et d'enrichissement qui valide et enrichie l'annotation dans sa base. Dans le cas ou elle n'est pas rejetée, la connaissance inférée est retournée à ITM. L'annotation enrichie est ensuite transférée au service d'évaluation de la pertinence qui ne réalise pas l'étape d'intégration et infère les associations de contradiction, confirmation et infirmation directement sur l'annotation dans sa base avant de les fournir à ITM. Dans le cas ou l'annotation est rejetée par le service de validation et d'enrichissement, elle n'est pas transmise au service d'évaluation de la pertinence. La réception par ITM des associations inférées sur la pertinence ou des rapports sur la violation des contraintes lève des alertes de différentes natures exploitables par l'utilisateur.

Cette première version de l'application n'offre pas encore la possibilité au veilleur d'entretenir personnellement le vocabulaire ou les jeux de règles et de contraintes ; car ces derniers sont communs à tous les utilisateurs et le droit de les modifier manuellement est réservé à un expert du domaine formé à l'entretien de la base. Le veilleur n'est autorisé qu'à accéder en lecture au niveau modèle d'ITM. Il peut cependant modifier librement le niveau instance (création/suppression).

L'application a été présentée lors du Semantic Web Challenge 2006 (Amardeilh et al. (2006)). A la mise en place du système, les principales difficultés ont consisté à synchroniser le vocabulaire interne à chaque service avec celui d'ITM puis à entretenir cette correspondance au fur et à mesure de l'évolution de l'ontologie. Les expérimentations réalisées ensuite ont porté sur douze mille documents. L'extraction linguistique a duré près de 6 heures et a conduit à l'extraction de plus de mille annotations et à la création de trois mille topics reliés entre eux par mille associations. La base de connaissances du serveur CoGITaNT contenait 50 règles et 76 contraintes à l'initialisation. A la fin du processus, elle comptait approximativement quatre mille sommets concept et le même nombre de sommets relation. 9 annotations ont été jugées incohérentes dès réception (étape 4.3.1) et 5 après ajout et enrichissement dans la base (étape 4.3.3). Les 150 acquisitions d'entreprises extraites des dépêches ont permis au système d'inférer 213 associations de contrôle et de confirmer 21 rumeurs. Le temps de traitement au cours de l'ajout contrôlé de chaque annotation reste raisonnable même quand la charge augmente (entre 40 et 160 ms).

6 Conclusion

Dans cet article, nous avons montré l'apport que peut constituer l'intégration d'un moteur d'inférence en graphes conceptuels à un outil de gestion des connaissances. Dans le cadre d'une application d'aide à la veille concurrentielle, nous proposons deux services $\mathcal{SG}$ l'un pour la validation et l'enrichissement des annotations et l'autre pour l'évaluation de leur pertinence.

Le caractère novateur de ce travail réside en ce qu'il établit une correspondance entre les topic maps et les graphes conceptuels, ce qui à notre connaissance n'avait pas encore été fait. En plus de munir ITM de mécanismes d'ajout contrôlé, cette correspondance peut aussi améliorer le dispositif d'interrogation de sa base de connaissances tel que décrit dans Carloni et al. (2006).

Certains travaux se rapprochent des notres dans le sens ou ils proposent une correspondance entre les TM et un formalisme de représentation des connaissances et de raisonnement.

Dans Moore (2001) deux sortes de transformation des TM vers RDF sont considérées. La première appelée "modelling the model", représente en RDF le modèle des TM (i.e. les primitives de modélisation) en utilisant les constructeurs RDF. Ce type de transformations peut être considéré comme syntaxique. La seconde appelée "mapping the model" tente d'assigner à chaque primitives de modélisation TM au moins une primitive RDF en essayant de préserver leur sémantique intuitive. Le problème de cette approche est qu'elle perd de l'information ce qui conduit Moore (2001) à proposer d'étendre le langage des TM. Dans Lacher et Decker (2001) une transformation à partir des TM considérés comme des graphes (tel que défini dans Biezunski et Newcomb (2001)) vers les graphes RDF est explicitement décrite. Cette transformation peut être rangée parmi celles issues de l'approche "modeling the model" et a la propriété d'être inversible.

Garshol (2005a) propose d'unifier les deux langages dans un model unique (appelé "Q"), ceci afin d'obtenir des transformations inversibles plus naturelles et moins verbeuses. Ce modèle est utilisé dans Garshol (2005b) pour munir d'une sémantique formelle un langage d'interrogation des TM.

Une troisième approche pour rendre les TM interrogeables consiste à les pourvoir de primitives d'interrogations intégrées au langage tel que décrit dans Barta et Salzer (2005). Ce travail propose un langage formel de description de chemins dans une TM permettant de naviguer et d'extraire des informations.

De façon générale, les précédents travaux n'évoquent pas la conséquence sémantique ou la déduction logique ; et décrivent davantage des méthodes d'interrogations de TM que des méthodes de raisonnement pour les TM.

Références

Amardeilh, F. (2006). Ontopop or how to annotate documents and populate ontologies from texts. In *Proceedings of ESWC 2006 Workshop on Mastering the Gap : From Information Extraction to Semantic Representation*. CEUR Workshop Proceedings. http ://ceur-ws.org/Vol-187/6.pdf.

Amardeilh, F., O. Carloni, et L. Noël (2006). PressIndex : a Semantic Web Press Clipping Application. Semantic Web Challenge Papers.

Amardeilh, F., P. Laublet, et J. Minel (2005). Annotation documentaire et peuplement d'ontologie à partir d'extractions linguistiques. In *Actes de la conférence d'Ingénérie des Connaissances*.

Auillans, P., P. O. D. Mendez, P. Rosenstiehl, et B. Vatant (2002). A formal model for topic maps. In *Proceedings of International Semantic Web Conference*, pp. 69–83.

Baget, J.-F. et M.-L. Mugnier (2002). Extensions of Simple Conceptual Graphs : The Complexity of Rules and Constraints. *Journal of Artificial Intelligence Research 16*, 425–465.

Baget, J.-F. et E. Salvat (2006). Rules dependencies in backward chaining of conceptual graphs rules. In *Proceedings of International Conference on Conceptual Structures*, pp. 102–116.

Barta, R. A. et G. Salzer (2005). The tau model, formalizing topic maps. In *Proceedings of 2th Asia-Pacific Conference on Conceptual Modelling*, pp. 37–42.

Biezunski, M. et S. R. Newcomb (2001). A processing model for xml topic maps. http ://www.topicmaps.net/pmtm4.htm.

Carloni, O., M. Leclere, et M.-L. Mugnier (2006). Introducing Graph-based reasoning in a knowledge management tool. In *Proceedings of The 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*.

Chein, M. et M.-L. Mugnier (1992). Conceptual Graphs : Fundamental Notions. *Revue d'Intelligence Artificielle 6*(4), 365–406.

Garshol, L. M. (2005a). Q : A model for topic maps : Unifying rdf and topic maps. In *Proceedings of Extreme Markup Languages*.

Garshol, L. M. (2005b). tolog - a topic maps query language. In *Proceedings of Topic Maps Research and Applications*, pp. 183–196.

Genest, D. (1997). Cogitant. http ://cogitant.sourceforge.net.

ISO/IEC :13250 (2000). Topic maps.
http ://www.y12.doe.gov/sgml/sc34/document/0129.pdf.

Lacher, M. S. et S. Decker (2001). On the integration of topic maps and rdf data. In *Proceedings of Extreme Markup Languages*.

Leclère, M. et M.-L. Mugnier (2006). Simple Conceptual Graphs with Atomic Negation and Difference. In *Proceedings of 14th International Conference on Conceptual Structures*, pp. 331–345.

Moore, G. (2001). Rdf and topic maps : An exercise in convergence. In *Proceedings of XML Europe Conference*.

Salvat, E. et M.-L. Mugnier (1996). Sound and Complete Forward and Backward Chainings of Graph Rules. In *Proceedings of 4th International Conference on Conceptual Structures*, Volume 1115 of *LNAI*, pp. 248–262. Springer.

Sowa, J. F. (1984). *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley.

TopicMaps.Org (2001). Xml topic maps (xtm) 1.0. http ://www.topicmaps.org/xtm/1.0/.

Summary

This article is devoted to an annotation validation and enrichment service built for an industrial knowledge management tool using Topic Maps (TM) language. Such a service needs inference capabilities ; thus we present a mapping from TM to conceptual graph (CG) formalism which provide semantics to TM language. The solution has been implemented in an application for media monitoring. Annotations are first extracted in an automatic way from documents about economic current events, and added to the knowledge base before being sent to the validation service which validates, enriches the annotation and proposes a relevance evaluation in order to help the watcher.