

L'approche NKRL pour l'analyse de certains documents de type défense

Gian Piero ZARRI

LaLIC, Université Paris 4-Sorbonne

Maison de la Recherche – 28, rue Serpente – 75006 Paris

gpzarri@paris4.sorbonne.fr, zarri@noos.fr

Résumé – Dans le panorama des outils informatiques de type ‘sémantique’, NKRL (*Narrative Knowledge Representation Language*) se caractérise par l’ajout d’une deuxième ontologie, l’ ‘ontologie des événements’, à l’ontologie standard des concepts. La représentation informatique des ‘événements élémentaires’ qui décrivent les agissements des protagonistes de situations données (du type ‘terrorisme’ par exemple) s’obtient ainsi par ‘spécialisation’ des structures propres à cette seconde ontologie. Des règles d’inférence sophistiquées permettent ensuite d’utiliser le matériel codé pour des opérations d’interrogation ‘intelligente’ et de mise en évidence de l’information implicite.

Abstract – Within the computer tools of a ‘semantic’ type, NKRL (*Narrative Knowledge Representation Language*) is characterized by the fact that it adds a second ontology, the ‘ontology of events’, to the standard ontology of concepts. The Computer Science representation of the ‘elementary events’ that describe the behaviour of the characters involved in given situations (e.g., terrorism situations) is then obtained by instantiation of the structures pertaining to this second ontology. Advanced inference techniques allow then us to make use of the resulting coded material for ‘intelligent’ querying operations and for automatically recovering some sort of ‘implicit information.

1. Introduction

Cet article se propose d’illustrer certaines caractéristiques de NKRL (*Narrative Knowledge Representation Language*), un langage de représentation de connaissances et, dans le même temps, un environnement informatique, utilisé pour formaliser et gérer des informations de type ‘narratif’ qui présentent un intérêt particulier d’un point de vue ‘économique’ (documents d’entreprise, dossiers médicaux, textes juridiques, dépêches d’agence, procès-verbaux, messages militaires, documents multimédia pour l’apprentissage à distance, etc.). Les exemples illustratifs se réfèrent à une application de NKRL dans le domaine ‘défense’ développée grâce au soutien partiel de la Commission Européenne dans le cadre du projet Parménides (IST P2001-39023), voir [1, 2]. Le corpus (partiellement) traité concernait à l’origine un millier de documents (déclassifiés) concernant le terrorisme dans les Philippines dans les années ‘90, mis à la disposition du consortium Parménides par le Ministère de la Défense grec, l’un des partenaires du consortium.

2. Templates et occurrences

2.1 L’ontologie des événements

Le mode de fonctionnement de NKRL (*Narrative Knowledge Representation Language*) se caractérise par

l’interaction continue entre deux structures hiérarchiques (deux ontologies). La première est une ontologie standard (‘binaire’) de concepts, très semblable aux ontologies qui peuvent être créées en utilisant des outils du style Protégé [3]. La deuxième, l’ ‘ontologie des événements’, est une structure de type nouveau dont les nœuds sont constitués de structures n -aires appelées ‘templates’ (modèles) qui suivent le format général représenté par (1) :

$$(L_i (P_j (R_1 a_1) (R_2 a_2) \dots (R_n a_n))) . \quad (1)$$

Dans cette formule, L_i dénote le nom symbolique du *template* ; P_j est un ‘prédicat sémantique’, R_k un rôle (cas) générique et les a_k correspondent aux différents arguments du prédicat introduits par les rôles respectifs. Les prédicats sont des ‘primitives’ et ils font partie de l’ensemble {BEHAVE, EXIST, EXPERIENCE, MOVE, OWN, PRODUCE, RECEIVE}. Les rôles sont aussi des primitives, et ils sont dénotés conventionnellement par {SUBJ(ect), OBJ(ect), SOURCE, BEN(e)F(iciary), MODAL(ity), TOPIC, CONTEXT}. La hiérarchie d’héritage qui représente l’ontologie des événements est appelée HTemp (hiérarchie des *templates*): la Figure 1 reproduit un fragment de la représentation ‘externe’ de HTemp qui montre en particulier les ‘noms symboliques’ (L_i) de quelques-uns des *templates* caractérisés par l’utilisation d’un prédicat sémantique (P_j) du type MOVE – voir la Table 1 ci-dessous pour les détails concernant la ‘structure interne’ (la

syntaxe) des différents *templates*. Comme il résulte des étiquettes symboliques associées aux *templates* de la Figure 1, chaque *template* correspond à la description formelle d'une *classe générale* d'événements narratifs – 'transférer des connaissances' ou 'transmettre un message' en Figure 1, mais aussi 'déplacer un objet', 'avoir une attitude positive ou négative envers quelqu'un', 'être présent à un certain endroit', 'créer un objet', 'créer une société', etc.

2.2 La représentation d'événements

Les instances des *templates* – c.-à-d., la représentation concrète d'événements élémentaires spécifiques comme « Demain, je vais déplacer l'armoire » ou « Marie était en train d'attendre son taxi » – sont appelées 'occurrences prédictives' dans le jargon de NKRL. Toute occurrence prédictive est normalement associée à un intervalle temporel sur l'axe des temps ; cet intervalle est délimité par deux 'repères temporels' ('*timestamps*') $\langle t_1 t_2 \rangle$ tels que $t_1 \leq t_2$ – la signification du symbole ' \leq ' est ici évidente. [4] fournit une description détaillée du système de représentation des données temporelles dans NKRL.

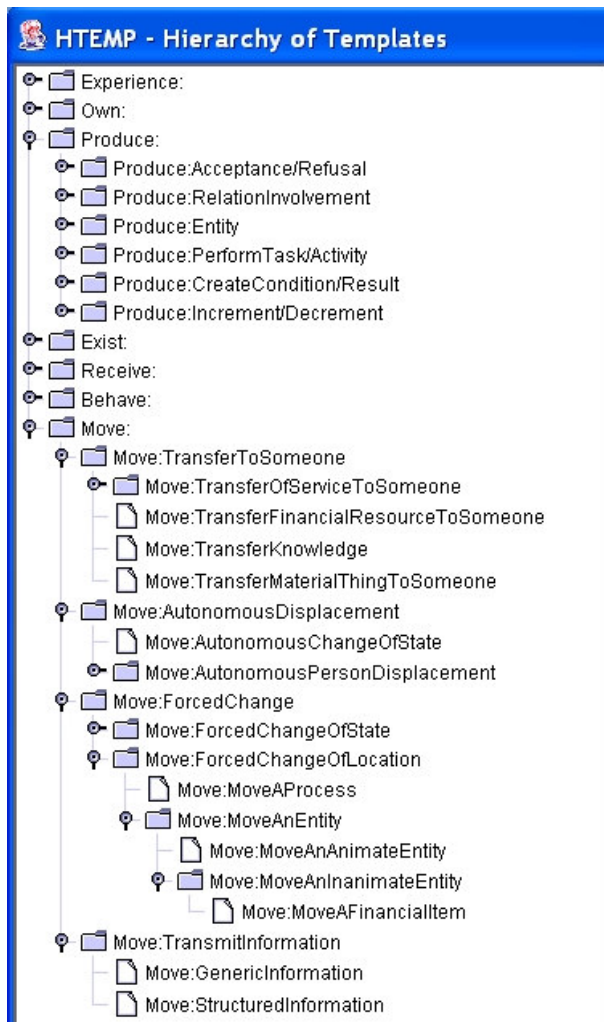


FIG. 1. Un fragment de la hiérarchie HTemp.

La Table 1 illustre les opérations nécessaires pour dériver du *template* Produce:Violence l'occurrence prédictive correspondante à la représentation formelle dans les termes de NKRL de l'événement élémentaire : « Le 23 novembre 1999, dans un village non spécifié, des hommes armés ont enlevé Robustiniano Hablo ». Cet exemple se réfère à l'application « terrorisme aux Philippines » déjà citée.

TAB. 1. Construction d'occurrences prédictives.

a)	
<i>name:</i> Produce:Violence	
<i>father:</i> Produce:PerformTask/Activity	
<i>position:</i> 6.35	
<i>NL description:</i> 'Execution of Violent Actions on the Filler of the BEN(e)F(iciary) Role'	
PRODUCE	SUBJ <i>var1</i> : [(<i>var2</i>) OBJ <i>var3</i> [SOURCE <i>var4</i> : [(<i>var5</i>)] BENF <i>var6</i> : [(<i>var7</i>)] [MODAL <i>var8</i> [TOPIC <i>var9</i> [CONTEXT <i>var10</i> { [modulators], +(abs) }
<i>var1</i>	= human_being_or_social_body
<i>var3</i>	= violence_
<i>var4</i>	= human_being_or_social_body
<i>var6</i>	= human_being_or_social_body
<i>var8</i>	= violence_, weapon_, small_portable_equipment, criminality/violence_related_tool, machine_tool, general_characterising_property,
<i>var9</i>	= h_class
<i>var10</i>	= situation_, spatio/temporal_relationship, symbolic_label
<i>var2, var5, var7</i>	= geographical_location
b)	
mod3.c5) PRODUCE SUBJ	(SPECIF INDIVIDUAL_PERSON_20 weapon_wearing (SPECIF cardinality_several_): (VILLAGE_1)
	OBJ kidnapping_
	BENF ROBUSTINIANO_HABLO
	CONTEXT #mod3.c6 date-1: 1999-11-20 date-2:
Produce:Violence (6.49)	
<i>Le 23 novembre 1999, dans un village non spécifié, des hommes armés ont enlevé Robustiniano Hablo.</i>	

Dans un *template*, les arguments du prédicat – c.-à-d., les termes a_i dans la formule (1) de la Section précédente – sont représentés par des variables var_i associées à des 'contraintes' qui définissent exactement le 'type' des valeurs, 'concepts' ou 'individus', qui peuvent être substituées à ces variables au moment de l'actualisation du *template*. Comme il apparaît clairement dans la Table 1, ces contraintes sont représentées par des concepts appartenant à l'ontologie 'standard' de NKRL, appelée HClass (hiérarchie des classes). Les rôles inclus entre parenthèses carrées, comme SOURCE (la personne, organisme etc. qui est à l'origine de l'événement), MODAL(ity) etc. dans le *template* de Table 1, représentent

des rôles ‘facultatifs’, c.-à-d., qui ne doivent pas être nécessairement remplis au moment de la création d’occurrences prédicatives. Dans le même *template*, des rôles comme SUBJ(ect), OBJ(ect), ou BEN(e)F(iciary) – le ‘destinataire’ de la violence – sont par contre ‘obligatoires’. Cela signifie que leur absence dans les occurrences prédicatives dérivées d’un *template* donné empêcherait de reconnaître ces occurrences comme des exemples particuliers de la classe générale d’événements représentée par le *template* en question.

Dans l’occurrence prédicative mod3.c5 de la Table 1b, le rôle SUBJ(ect) est ‘rempli’ par un ‘argument structuré’ (ou ‘expansion’) qui utilise l’ ‘opérateur d’attribution’ (*‘attributive operator’*), SPECIF(ication). SPECIF est l’un des quatre opérateurs utilisés pour construire des *fillers* complexes (‘expansions’), voir [5]; les listes SPECIF, caractérisées par le format : (SPECIF e p₁ ... p_n), sont utilisées pour représenter certains propriétés ou attributs p_i déclarés à propos du premier élément e de la liste. L’argument structuré de la Figure 1b signifie que l’enlèvement a été réalisé par un groupe, (SPECIF cardinality_ several_)), d’individus inconnus désignés collectivement en tant que INDIVIDUAL_PERSON_20. (SPECIF cardinality_ several_)) est la structure syntaxique utilisée dans NKRL pour représenter la cardinalité d’ensembles de taille totalement indéfinie, comme ceux qui correspondent à des référents pluriels indéterminés du type « hommes » ou « livres ». Les inconnus étaient armés, *weapon_wearing*. *weapon_wearing* représente l’une des spécialisations du concept *clothing_attribute* de HClass et correspond aussi, au moyen des généralisations *clothing_attribute* et *physical_aspect_attribute*, à une spécialisation de *animate_entity_property*. A noter également que cet argument structuré est associé – au moyen du symbole ‘.’ – à un ‘attribut de localisation’, représenté par l’individu VILLAGE_1 in Table 1b.

Nous pouvons ajouter maintenant quelques précisions à propos de la hiérarchie HTemp. Actuellement (novembre 2007) elle inclut 150 *templates*, et peut être considérée comme une sorte de ‘catalogue’ où puiser, en cas de besoin, les structures narratives à utiliser. Dans la mesure où elle fait partie en pratique de la définition du langage, on pourrait en déduire que ses éléments sont fixes et invariables. En réalité, HTemp est une structure mouvante, du moins en partie. Chaque fois qu’une nouvelle application NKRL est mise en chantier, il est possible en effet que l’on doive dériver des nouveaux *templates* de ceux déjà existants et les ‘spécialiser sur commande’ (*customize*) pour les besoins de l’application. Si, à la fin de l’application, l’on considère que certaines de ces nouveaux *templates* ont des caractéristiques de généralité, ils sont insérés dans HTemp.

Nous noterons encore que cette approche au traitement de documents narratifs fondé sur la notion de ‘catalogue’ est particulièrement avantageuse du point de vue des applications pratiques. Elle implique, en particulier, que : a) les développeurs ne seront pas obligés de créer eux-mêmes les structures syntaxiques et sémantiques

nécessaires pour décrire les événements propres à un ensemble (suffisamment) important de documents narratifs ; b) il devient aussi beaucoup plus facile d’assurer la reproduction et le partage de résultats précédents.

1.3 Propriétés additionnelles du langage

L’un des problèmes les plus difficiles dans le traitement d’informations narratives comme les dépêches d’agence concerne la représentation des ces ‘phénomènes de jonction’ (*connectivity phenomena*) représentés par les liens de causalité, but, coordination, subordination, discours indirect etc. – en bref, par tous ces phénomènes qui permettent de relier ensemble dans un cadre plus vaste plusieurs événements élémentaires, donnant lieu à une ‘signification sémantique’ globale de l’ensemble qui va bien au-delà de la simple addition des significations partielles. Dans NKRL, les phénomènes de ce type sont pris en compte en utilisant un nombre limité de structures ‘de second ordre’, créées par des opérations de ‘réification’ qui font usage des ‘étiquettes symboliques’, comme mod3.c5 dans la Table 1, des occurrences prédicatives – rappelons ici qu’une occurrence prédicative correspond à un événement élémentaire.

La ‘construction complétive’ (voir [5] pour plus de détails) fournit un premier exemple de structure de deuxième ordre dans NKRL : elle consiste dans l’accepter en tant que *filler* d’un rôle d’une occurrence prédicative l’étiquette symbolique propre à une autre occurrence prédicative. Dans NKRL, les rôles qui peuvent accepter une étiquette en tant que *filler* sont les rôles OBJ(ect), MODAL(ity), TOPIC et CONTEXT. Par exemple, en revenant à l’exemple de Table 1, nous pouvons constater que le contexte de l’enlèvement est représenté par le *contenu global* d’une occurrence mod3.c6, non représentée en Table 1 et totalement indépendante de l’occurrence mod3.c5 qui concerne l’enlèvement. Le *filler* de TOPIC dans mod3.c5 est ainsi l’étiquette mod3.c6, précédée du symbole ‘#’ pour signaler sa condition de *filler* particulier.

Une deuxième modalité de mise en relation d’événements élémentaires, plus générale, fait appel aux ‘occurrences de liaison’ (*binding occurrences*). Les occurrences de liaison sont représentées par des listes d’étiquettes symboliques d’occurrences prédicatives ; ces listes sont caractérisées par la présence dans la première position de la liste d’ ‘opérateurs de liaison’ (*binding operators*) comme CAUSE et GOAL – ces derniers se réfèrent à la ‘taxonomie de la causalité’ de NKRL, voir [5, 6] – COND(ition), ALTERN(ative), COORD(ination), etc.

Le Table 2 fournit ainsi la représentation en termes de NKRL de ce fragment de texte narrative : « ... À une date inconnue qui précède celle de la dépêche, le 9 mai 1999, Abubakar Abdurajak Janjalani (le chef du groupe Abu Sayyaf) a reçu de l’argent de Bin Laden pour l’achat d’environ 40 fusils automatiques M16 ». Le groupe Abu Sayyaf est un mouvement islamiste actif dans le sud des Philippines et qui prône l’indépendance de cette région. À noter, dans le codage des occurrences prédicatives de la

Table 2, l'utilisation dans mod243.c5 de la formule NKRL standard pour représenter le coût d'une transaction et le type de devise utilisé : (SPECIF money_ <currency_> (SPECIF amount_ <real_>)). Dans cette formule, les concepts money_ et amount_ sont invariables, le concept currency_ est normalement remplacé par l'une des ses instances (individus, composante énumérative) comme USA_DOLLAR, PHILIPINO_PESO, YEN_ etc., et real_ par le montant du paiement. Dans l'exemple de Table 2, ce montant n'est pas exprimé car il n'est pas précisé dans la dépendance.

TAB. 2. Un exemple d'occurrence de liaison.

mod243.c3)	(GOAL	mod243.c4	mod243.c6)
mod243.c4)	BEHAVE	SUBJ	OSAMA_BIN_LADEN
		MODAL	#mod243.c5
		date-1:	before-9-mai-1998
		date-2:	
mod243.c5)	MOVE	SUBJ	OSAMA_BIN_LADEN
		OBJ	(SPECIF money_ USA_DOLLAR (SPECIF amount_))
		BENF	ABUBAKAR ABDURAJAK JANJALANI
		date-1:	before-9-mai-1998
		date-2:	
(*)mod243.c6)	PRODUCE	SUBJ	ABU_SAYYAF_GROUP
		OBJ	purchase_
		TOPIC	(SPECIF m16_ (SPECIF cardinality_ (SPECIF approximate_amount 40)))
		date-1:	circa-9-mai-1998
		date-2:	

À partir de la Table 2, nous pouvons remarquer que l'idée générale de 'agir de façon à obtenir un certain résultat' est traduite dans NKRL par :

- La présence d'une occurrence de liaison de type GOAL comportant deux arguments (étiquettes d'occurrences).
- Le premier argument, occurrence mod243.c4 dans notre exemple, correspond nécessairement à une occurrence prédicative qui est l'instance d'un *template* appartenant à la branche Behave:Focus de HTemp. Ce premier argument correspond ainsi à l'élément 'prédisposition à l'action' de la situation globale, et permet de localiser l'instigateur initial de l'action (OSAMA_BIN_LADEN dans notre exemple), le cas échéant les modalités qu'il utilise (exprimées dans l'exemple par une construction complétive), les éléments temporels etc.
- Le deuxième élément représente par contre le 'résultat attendu', et il peut correspondre, comme dans cet exemple, à une occurrence prédicative unique ou à plusieurs occurrences prédicatives liées à leur tour par une occurrence de liaison ; à noter que cette deuxième occurrence (groupe d'occurrences) est très souvent 'incertaine' – c.-à-d., caractérisée par un 'code de validité' du type 'incertain', voir "*" en Table 2 – car elle à lieu 'dans le futur' par rapport à l'élément 'prédisposition'.

La syntaxe générale utilisée dans NKRL pour exprimer les situations du type 'agir pour obtenir un certain résultat' peut être ainsi résumée de la façon suivante :

c_α) (GOAL c_β c_γ)
 c_β) BEHAVE SUBJ <human_being_or_social_body>
 $(*)c_\gamma$) <predicative_occurrence(s), with any syntax>

où le symbole "*" représente le (possible) 'code de validité' du type 'incertain', voir [5].

3. Interrogation et inférence

3.1 Les modèles de recherche

Ces modèles – qu'il faut considérer comme les correspondants formels de questions en langage naturel – sont des structures de données NKRL destinées à fournir le cadre général des informations qui doivent être retrouvées, par simple filtrage ou par unification, à l'intérieur d'une base de connaissances formée d'occurrences NKRL. Les modèles de recherche sont, dans la pratique, des *templates* de la hiérarchie HTemp spécialisés et partiellement actualisés : plus précisément, les 'variables explicites' *var_i* associées aux rôles de ces *templates*, voir la Table 1, sont remplacées par des concepts ou des individus compatibles, d'un point de vue HClass, avec les contraintes originalement imposées sur ces variables. Quand les variables explicites sont remplacés par des concepts, ces derniers se comportent comme des 'variables implicites', c.-à-d. que, au moment de l'unification/filtrage avec les données de la base de connaissances, ils peuvent s'apparier avec des concepts identiques, avec les spécialisations de ces concepts, et avec tous les individus possibles qui sont des instances de tous les concepts en question. La Table 3 montre ainsi un modèle de recherche très simple qui se réfère à l'application Parménides sur le terrorisme aux Philippines, et qui correspond à une question en langage naturel du type : « Il y a-t-il des informations dans le système correspondant à une action de kidnapping menée par des personnes à l'encontre, rôle BEN(e)F(iciary), d'autres personnes ? ». L'une des réponses possibles correspond évidemment à l'occurrence mod3.c5 représentée dans la partie b) de la Table 1.

TAB. 3. Un exemple de modèle de recherche NKRL.

PRODUCE
SUBJ : human_being :
OBJ : kidnapping_
BENF : human_being
}
date1 : 1/1/1998
date2 : 31/12/2000

Le modèle a été obtenu par actualisation partielle du *template* Produce:Violence, voir la Table 1a, où les variables explicites ont été remplacées par des concepts : le concept kidnapping_ par exemple, spécifique de violence_ dans HClass, a pris la place d'une variable explicite du *template* d'origine ayant violence_ comme

contrainte. Les deux repères temporels associés au modèle représentent maintenant l' 'intervalle de recherche' à utiliser pour limiter l'unification avec les données de la base à la tranche temporelle que l'on veut explorer – dans l'exemple, les années entre 1998 et 2000. Le module NKRL utilisé en général pour interroger la base de connaissances et effectuer les opérations d'inférence est le module *QueryModule2004* : l'unification proprement dite est effectuée par un sous-module appelé *Fum* (*Filtering Unification Module*) qui effectue principalement des comparaisons d'arbres. D'un point de vue algorithmique, l'intérêt majeur du *Fum* réside a) dans les opérations complexes qu'il doit effectuer pour unifier correctement les 'expansions' des occurrences avec celles qui apparaissent éventuellement dans le modèle, et b) dans l'utilisation d'une série d'index fondés sur l'utilisation d'informations temporelles qui sont superposés à la base de connaissances pour optimiser les opérations d'unification et filtrage, voir [4].

3.2 Les règles de transformation

Dans le cas d'insuccès d'un modèle de recherche, il est encore possible d'obtenir des éléments de réponse en utilisant des analogies sémantiques entre l'information qui avait été demandée au départ et celle qui est réellement présente à l'intérieur de la base. Le principe consiste dans l'utilisation d'une catégorie particulière de règles NKRL, les 'règles de transformation', qui se chargent de 'transformer' automatiquement un modèle de recherche qui a échoué dans un ou plusieurs modèles différents qui – à la différence, par exemple, des 'questions transformées' utilisées dans un contexte de bases de données – ne sont pas strictement 'équivalentes' à la question d'origine, mais uniquement 'proches d'un point de vue sémantique'.

Les règles de transformation sont formées d'une partie gauche ('antécédent') qui correspond à une *généralisation* du modèle de recherche à transformer et d'une partie droite ('conséquent') utilisée pour générer les structures du modèle transformé. La formulation de l'antécédent inclut des *variables explicites*, qui doivent se retrouver dans le conséquent accompagnées de variables nouvelles. Le conséquent peut inclure plusieurs schémas, qui donnent donc origine à plusieurs modèles de recherches : dans ce cas, après actualisation avec les paramètres du modèle à transformer récupérés dans l'antécédent, ces schémas (ces modèles) doivent *tous* trouver une correspondance à l'intérieur de la base de connaissances afin que la transformation soit valide. Si l'on dénote par A et C_s respectivement l'antécédent et le conséquent, une règle de transformation peut être exprimée de la façon suivante :

$$A(var_i) \Rightarrow C_s(var_j), \quad var_j \subset var_i \quad (2)$$

Le symbole de transformation, ' \Rightarrow ', a une double signification :

- d'un point de vue opérationnel, la flèche indique la direction de la transformation : le modèle de recherche d'origine, qui est une actualisation de l'antécédent A , est supprimé et remplacé par un ou plusieurs modèles

de recherche obtenus par actualisation du conséquent C_s avec les paramètres du modèle d'origine ;

- du point de vue logique, on peut assumer qu'entre l'information que l'on pourra récupérer au moyen des C_s et celle que l'on aurait du obtenir au moyen de A existe une relation d'*implication* qui, normalement, dénote uniquement une implication *possible*.

Pour d'autres détails formels à propos de transformations voir, par exemple, [5].

Un exemple de règle de transformation est reproduit dans la Table 4. Elle fait partie d'un groupe de règles d'inférence, transformations et 'hypothèses' (voir ci-dessous), mises à point dans le cadre de Parménides pour étudier le phénomène du 'kidnapping dans le but d'obtenir une rançon' – d'autres types d'enlèvement concernent, par exemple, des enlèvements de type politique, pour vengeance, dans un contexte de crime sexuel, dans le cadre d'opérations militaires, etc. Supposons ainsi d'être en train de travailler dans le contexte de l'enlèvement aux Philippines de l'individu Robustiniano Hablo que nous avons déjà rencontré dans la Table 1, et supposons d'interroger la base de connaissances pour savoir s'il est riche ou s'il exerce une profession libérale – ces deux catégories de personnes sont les cibles préférées dans le cadre d'enlèvement pour rançon. En l'absence d'une attestation explicite dans la base du fait que Robustiniano Hablo est associable à l'une de ces deux catégories, nous pouvons utiliser la règle de transformation de la Table 4 qui peut être énoncée de la façon suivante : « *Dans le cadre d'un enlèvement pour rançon, le parent d'un homme riche ou d'une personne qui exerce une profession libérale peut être considéré à son tour – du point de vue donc des ravisseurs – comme riche ou comme exerçant une profession libérale (avocat, médecin, journaliste, homme d'affaires etc.)* ». À noter que, dans le deuxième schéma du conséquent, la variable $var2$ introduite par l'antécédent a été renommée $var5$ pour des raisons de généralisation – la règle est, en réalité, la fusion de deux règles, l'une à propos des 'riches' et l'autre à propos des 'professionnels' – sans que cela ne remette en jeu la contrainte formelle selon laquelle toutes les variables introduites par l'antécédent doivent se retrouver dans le conséquent.

En utilisant maintenant la composante 'transformation' du sous-module *InferenceEngine* qui fait partie de *QueryModule2004* nous obtiendrons, dans le cas d'un échec de la question directe à propos de la 'richesse' de Robustiniano Hablo – voir la première partie de la Table 4 et la Figure 2 – les réponses de la Figure 3.

Sans entrer dans les détails du fonctionnement du moteur d'inférence – voir [6] à ce propos – nous remarquerons que l'appariement entre le modèle de recherche d'origine et l'antécédent de la transformation permet de lier la variable $var1$ à la valeur ROBUSTINIANO_HABLO, et de 'spécialiser' ainsi la règle vers la résolution du cas particulier de l'enlèvement de cet individu. Nous pouvons paraphraser le résultat des Figures 2 et 3 en affirmant que : « Dans le cadre de nos recherches à propos du kidnapping de Robustiniano Hablo, nous ne

pouvons pas établir si cet individu est riche, mais le système est toutefois en mesure de nous fournir des informations sémantiquement proches, à savoir que son père est un riche homme d'affaires ». Nous ajouterons ici que la stratégie d'exécution de l'*InferenceEngine* est fondamentalement celle d'un 'chaînage arrière' avec *backtracking* chronologique. À chaque pas de l'inférence, on essayera ainsi de construire des modèles de recherche standard à partir des différents schémas de conséquent en utilisant les valeurs associées aux variables dans les pas précédents, et à unifier ces modèles (via le *Fum*) avec les données de la base : en cas d'échec, le moteur reviendra au point de choix antérieur.

TAB. 4. Un exemple de règle de transformation.

<p>BEHAVE SUBJ : ROBUSTINIANO_HABLO : MODAL : wealthy_ {} date1 : date2 :</p> <p><u>Modèle à transformer</u></p> <p>TRANSFORMATION #0 <u>antecedent</u> :</p> <p>BEHAVE SUBJ var1 MODAL var2</p> <p>var1 = individual_person var2 = wealthy_, professional_role</p> <p><i>On cherche à prouver qu'une personne donnée est riche ou qu'elle exerce une profession libérale (médecin, journaliste, homme d'affaires...).</i></p> <p><u>first consequent schema (cons1) :</u></p> <p>BEHAVE SUBJ var1 MODAL var3 TOPIC var4</p> <p>var1 = individual_person var3 = extended_family_role var4 = individual_person</p> <p><i>On cherche à prouver que la personne dont on s'occupe a une quelque relation de parenté avec une deuxième personne.</i></p> <p><u>second consequent schema (cons2) :</u></p> <p>BEHAVE SUBJ var4 MODAL var2</p> <p>var1 = individual_person var5 = wealthy_, professional_role</p> <p><i>On cherche à prouver que cette deuxième personne est a son tour riche, médecin, journaliste, home d'affaires....</i></p>
--

3.3 Hypothèses et intégration des règles d'inférences de NKRL

Les règles du type 'hypothèse' sont les plus connues parmi les procédures d'inférence de NKRL, et on fait l'objet de nombreuses publications voir, par exemple, [6]. À la différence des transformations qui ont comme point de

départ un modèle de recherche échoué, les hypothèses prennent origine d'une information attestée dans la base de connaissances (une occurrence prédicative) et, dans l'absence d'un 'contexte explicatif' pour l'information en question, essayent d'en construire un automatiquement. Pour reprendre un exemple qui a été largement utilisé dans les publications sur NKRL, supposons d'avoir retrouvé dans la base une information (une occurrence prédicative) qui correspond à : « Pharmacopeia, une société américaine qui opère dans le domaine des biotechnologies, a reçu 64.000.000 dollars USA de la société allemande Schering dans le cadre des ses activités de R&D ». Dans l'absence d'explications explicites, nous utiliserons une inférence de type 'hypothèse' en essayant de vérifier dans la base (grâce au *Fum*) la vraisemblance de certains 'pas de raisonnement' du type : a) « Pharmacopeia et Schering ont signé une quelque forme d'accord concernant la création par Pharmacopeia d'un nouveau 'produit', par exemple, une nouvelle molécule » et b) « nous pouvons vérifier que, dans le cadre d'un accord de ce type, Pharmacopeia a effectivement crée une nouvelle molécule ». Il est bien évident qu'une 'règle' de ce type doit être interprétée comme faisant partie d'une famille entière de règles, où chacune est en mesure de donner une explication différente à propos du transfert d'une somme importante d'argent d'une société à une autre.

Même si leur interprétation est totalement différente – les transformations cherchent, dans le cas d'impossibilité d'une réponse directe à une question donnée, à fournir des réponses 'sémantiquement proches' de celle initialement cherchée, tandis que les hypothèses visent à construire automatiquement un cadre explicatif pour une information déjà attestée dans la base – ces deux catégories de règles sont très semblables du point de vue formel, et la composante 'hypothèses' de *InferenceEngine* partage le 90% de ces modules avec la composante 'transformation'.

D'un point de vue formel, une règle d'inférence du type 'hypothèse' peut être exprimée de la façon suivante :

$$P(var_i) \leftarrow Cn_k(var_j), \quad var_j \subset var_i \quad (3)$$

où P représente maintenant la 'prémisse' de la règle – c.-à-d., l'expression formelle (qui utilise des variables explicites var_i) de la classe d'événements qui peuvent être 'expliqués' au moyen de cette règle – et Cn_k représentent une série de 'schémas de condition' qui, comme dans le cas des transformations, sont exprimés au moyen des variables var_i de la prémisse auxquelles s'ajoutent d'autres variables propres aux Cn_k . L'application de la règle consiste dans la dérivation de modèles de recherche à partir des schémas Cn_k que l'on essaiera ensuite d'apparier aux données de la base : si cette opération a succès pour tous les Cn_k , on assumera l'ensemble des occurrences prédicatives ainsi retrouvées comme une 'explication possible' de l'événement particulier, instance de la prémisse P , que l'on aura essayé d'expliquer'. Le symbole ' \leftarrow ' a donc ici la signification de : « il est possible que P dérive des Cn_k ».

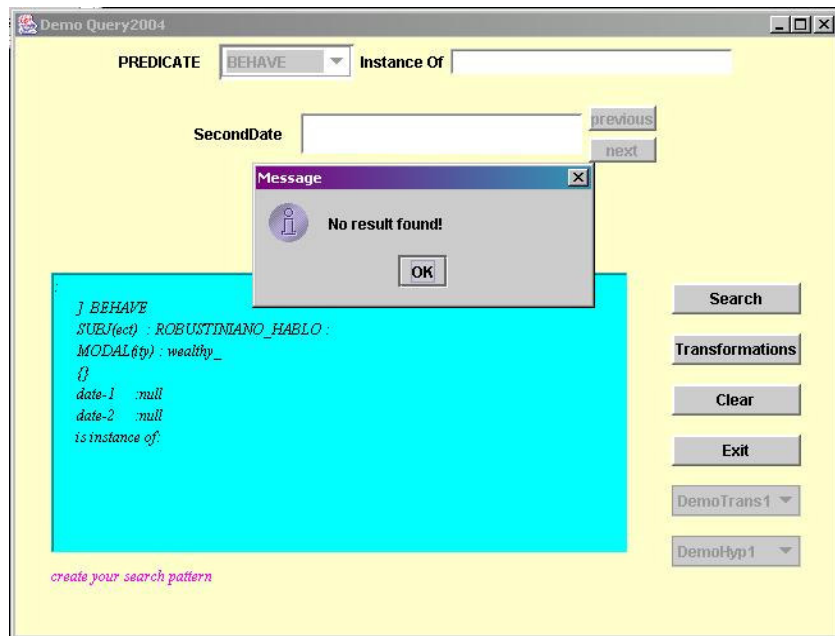


FIG. 2. Échec d'un modèle de recherche ('search pattern').

```

The start pattern
:
] BEHAVE
SUBJ(ect) : ROBUSTINIANO_HABLO :
MODAL(ity) : wealthy_
{ }
date-1 :null
date-2 :null
is instance of:

*****
The result for the Consequent 1

mod3.c11:
] BEHAVE
SUBJ(ect) : ROBUSTINIANO_HABLO :
MODAL(ity) : son_
TOPIC : INDIVIDUAL_PERSON_21
{obs }
date-1 :21/11/1999
date-2 :null
is instance of:Behave:Kinship
Natural language description :
On November 21, 1999, we can remark that Robustiniano Hablo is a son of the individual mentioned
in occurrence mod3.c6.

*****
The result for the Consequent 2

mod3.c12:
] BEHAVE
SUBJ(ect) : INDIVIDUAL_PERSON_21 :
MODAL(ity) : ( SPECIF business_person wealthy_ )
{obs }
date-1 :21/11/1999
date-2 :null
is instance of:Behave:Role
Natural language description :
On November 21, 1999, we can remark that the individual characterised in mod3.c12 as father of
Robustiniano Hablo is a wealthy businessman.

```

FIG. 3. Résultats pour la transformation du Tableau 6.

En renvoyant encore une fois à [6] pour des exemples plus détaillés, nous nous limiterons maintenant à évoquer de façon informelle, voir la Table 5, l'une des possibles hypothèses qui sont à même d' 'expliquer', dans un contexte de kidnapping pour rançon, un enlèvement comme celui de Robustiniano Hablo (voir la Table 1 ci-dessus) effectué par un groupe de personnes. En examinant la Table 5, il apparaît que l'hypothèse en question est formée de quatre schémas de condition, qui doivent être tous satisfaits – les modèles de recherche dérivés de ces schémas doivent tous trouver un appariement dans la base – afin que la règle soit validée.

TAB. 5. Description informelle d'une hypothèse dans un contexte 'kidnapping pour rançon'.

- (**Cond1**) On cherche à prouver que les ravisseurs font partie d'un mouvement séparatiste ou d'une organisation terroriste.
- (**Cond2**) On cherche à prouver que ce mouvement séparatiste ou cette organisation terroriste pratique couramment le kidnapping pour rançon envers certaines catégories de victimes.
- (**Cond3**) On vérifie en particulier que ces victimes sont des 'riches', ou des hommes d'affaires et des cadres supérieurs du privé : d'autres hypothèses vont traiter de l'enlèvement de membres du clergé, de l'enlèvement de fonctionnaires, de membres de l'armée, etc.
- (**Cond4**) On vérifiera maintenant que le kidnappé, Robustiniano Hablo par exemple, est effectivement 'riche', ou un homme d'affaires ou un cadre supérieur.

Dans le contexte des procédures d'inférence de NKRL, une avancée particulièrement importante a été récemment effectuée grâce au projet Parménides, qui a permis la réalisation de logiciels permettant d'effectuer l'intégration des deux classes de procédures d'inférence, hypothèses et transformations, que l'on vient d'évoquer. Le but de cette intégration est celui d'élargir encore les possibilités de NKRL de mettre en évidence de façon automatique les *relations implicites* entre les données de la base. De ce que nous avons exposé jusqu'à maintenant, il apparaît évident que le 'cœur' de l'exécution des procédures d'inférence de NKRL consiste a) dans la création de modèles de recherche standard à partir des schémas du conséquent, dans les transformations, et des schémas de condition, dans les hypothèses, et b) dans l'utilisation du *Fum* pour appairer ces modèles avec les données de la base d'annotations/occurrences. Dans la mesure où ces 'modèles dérivés' sont des modèles de recherche standard, il sera donc possible de les transformer : d'où l'idée d'utiliser les procédures de transformation à l'intérieur des procédures d'hypothèses pour élargir encore les possibilités d' 'explication' de celles-ci en utilisant toutes les 'variantes sémantiques' des formulations de départ. D'un point de vue informatique – même si la réalisation de l'intégration semble se réduire à des simples problèmes de co-routines

impliquant les deux sous-modules, pour les transformations et les hypothèses, d'*InferenceEngine* – des difficultés considérables ont dû être résolues, concernant par exemple l'établissement d'une correspondance entre les variables propres à un contexte 'hypothèse' et celles propres à un contexte 'transformations'. La 'théorie' de l'intégration, et la description de l'implémentation informatique, sont décrites en détails dans [6].

À titre d'exemple, la Table 6 reproduit la même description informelle de l'hypothèses de la Table 5 où, pour chaque pas d'inférence, nous avons ajouté des règles de transformations (T0, T1, T2 etc.) qui sont effectivement présentes dans la base de règles de l'environnement NKRL et qu'il est possible d'activer à partir des modèles de recherches tirés des schémas de condition 'normaux' de l'hypothèse – le conséquent des règles T0, T1, T3 et T5 est formé de deux schémas, celui des règles T2 et T4 d'un seul schéma. Par exemple, la règle T3 pourra être activé dans le cas où il n'existe pas dans la base d'annotations/occurrences l'attestation directe que le groupe de ravisseurs fait partie d'une composante connue – comme Abu Sayyaf, le Moro Islamic Liberation Front (MILF) etc. – de la nébuleuse séparatiste dans les Philippines du Sud. On ira alors chercher si les ravisseurs font par hasard parti d'un sous-groupe quelconque de l'une de ces composantes connues – par exemple, s'ils font partie du *Special Operation Group* du MILF où des différentes mouvances des *renegades* du même MILF – à condition, bien entendu, que l'on puisse vérifier que le groupe principal a toujours le contrôle de ses différents sous-groupes. Dans ce cas, on pourra assumer, pour pouvoir continuer dans l'hypothèse, que les ravisseurs font encore partie d'Abu Sayyaf ou du MILF etc. , et l'on pourra ainsi transmettre la valeur récupérée, MORO_ISLAMIC_LIBERATION_FRONT ou ABU_SAYYAF aux pas successifs de l'hypothèse. Pour le deuxième schéma de condition, nous avons à disposition une 'famille' entière de règles de transformation, toutes fondées sur le principe que, si des organisations comme Abu Sayyaf ou le MILF envoient des requêtes de rançon aux familles de certains kidnappés, ou si elles ont reçu de l'argent par les familles, ou si des sous-groupes de ces organisations ont envoyé de requêtes, reçu de l'argent etc., tous ces 'événements' prouvent de manière indirecte que les organisations en question pratiquent effectivement le kidnapping pour rançon. On remarquera finalement que la règle de transformation T0 est celle que l'on a déjà utilisée, de façon autonome, pour répondre à la question à propos de la 'richesse' de Robustiniano Hablo, voir la Section 2.2 ci-dessus.

4. À propos d'implémentation

D'un point de vue informatique, NKRL se présente comme un environnement complet, implémenté en Java 2, qui contient tant les outils nécessaires à la construction des occurrences (prédicatives et de liaison) et des règles

d'inférences, que les modules d'interrogation (et d'inférence) indispensables pour exploiter ces occurrences et règles dans le cas d'applications concrètes. L'environnement se présente en deux versions, une version 'orientée-fichiers' – qui peut donc être installée même sur des ordinateurs de faible puissance – et une version 'orientée-ORACLE'. À part les différences évidentes à propos de la 'taille' des applications possibles, la différence la plus importante entre les deux versions consiste dans le fait que la version-fichier permet seulement un nombre très limité d'opérations à propos de l'ontologie standard des concepts (HClass) – en particulier, dans la version fichier, il n'est pas possible actuellement d'associer des *frames* aux différents concepts : pour ce faire, il est ainsi nécessaire d'utiliser systématiquement la version ORACLE. La version fichier permet par contre une exécution accélérée des procédures d'inférences les plus complexes, comme celles que l'on active dans le cadre d'une intégration entre 'hypothèses' et 'transformations', voir la Section 2.3 ci-dessus.

En ce qui concerne les améliorations possibles, plusieurs d'entre elles se réduisent à des retouches de type 'cosmétique'. Par exemple, un certain nombre des procédures d'affichage des résultats (y compris la visualisation des résultats de procédures d'inférence, voir par exemple la Figure 3) proviennent de vieux logiciels réalisés dans le cadre d'anciens projets européens : elles sont donc plutôt 'inesthétiques' et ne rendent pas justice à la complexité et à l'intérêt des résultats. Des améliorations plus 'substantielles' concernent les points suivants :

- L'ajout de procédures pour permettre l'interrogation en langage naturel (NL) de bases de connaissances NKRL, c.-à-d., des procédures destinées à permettre une traduction automatique de questions NL dans des modèles de recherche 'externes', du type de celui de la Table 3. Des résultats très encourageants ont été obtenus dans ce contexte grâce à l'utilisation de techniques d'analyse linguistique 'de surface' (*shallow parsing*) du type celles effectuées par les grammaires AGFL, voir [7], associées avec des simples règles du type 'règles de production' pour générer les *templates* et avec l'aide des procédures d'inférence standard de NKRL. En ce qui concerne le problème plus général (et plus difficile) concernant la production automatique ou semi-automatique d'occurrence directement à partir des documents d'origine, les recherches sont moins avancées, même si des progrès considérables ont été récemment obtenus dans le cadre de Parménides, voir [2].
- L'introduction de procédures d'optimisation pour améliorer les performances du 'retour en arrière de type chronologique' (*chronological backtracking*) qui est à la base du fonctionnement de l' *InferenceEngine* de NKRL. Nous pensons avoir recours dans ce contexte aux techniques d'optimisation bien connues développées dans un contexte de 'Logic Programming', voir [8] par exemple. Cela devrait permettre, entre autres, d'aligner le temps d'exécution des règles d'inférence

dans un contexte ORACLE avec celui (quelques seconds dans les situations les plus complexes) que l'on observe pour la version orientée-fichier.

TAB. 6. Application des plusieurs règles de transformation à l'intérieur d'un même règle d'hypothèse.

- **(Cond1)** On cherche à prouver que les ravisseurs font partie d'un mouvement séparatiste ou d'une organisation terroriste.
 - **(Règle T3, Conséquent1)** *On cherche à vérifier si les kidnappeurs font partie d'un quelque 'sous-groupe' d'un mouvement séparatiste ou organisation terroriste et, dans ce cas,*
 - **(Règle T3, Conséquent2)** *on cherche à vérifier si le mouvement séparatiste ou l'organisation terroriste ont toujours le contrôle du sous-groupe en question. Dans ce cas, on peut considérer que les ravisseurs font encore partie du mouvement séparatiste ou de l'organisation terroriste.*
- **(Cond2)** On cherche à prouver que ce mouvement séparatiste ou cette organisation terroriste pratique couramment le kidnapping pour rançon envers certaines catégories de victimes.
 - **(Règle T1, Conséquent1)** *On cherche à vérifier si le mouvement séparatiste ou l'organisation terroriste repérés au pas précédent se composent de plusieurs sous-groupes et, dans ce cas,*
 - **(Règle T1, Conséquent2)** *on cherche à vérifier si l'un ou plusieurs de ces sous-groupes pratiquent couramment le kidnapping pour rançon.*
 - **(Règle T4, Conséquent)** *La famille du kidnappé a reçu une requête de rançon de la part du mouvement séparatiste ou organisation terroriste.*
 - **(Règle T5, Conséquent1)** *La famille du kidnappé a reçu une requête de rançon de la part d'un groupe de personnes ou d'un individu, et dans ce cas,*
 - **(Règle T5, Conséquent2)** *ce groupe ou cet individu font partie du mouvement séparatiste ou de l'organisation terroriste.*
 - ...
- **(Cond3)** On vérifie que, en particulier, ces victimes sont des 'riches', des hommes d'affaires et des cadres supérieurs (d'autres hypothèses vont traiter des journalistes, du clergé, des fonctionnaires, etc.).
 - **(Règle T2, Conséquent)** *Dans un contexte de 'kidnapping pour rançon', on peut affirmer que des 'personnalités' comme des membres d'organisations internationales, des médecins, des artistes, des vedettes de la télévision etc. peuvent être considérés comme 'riches' ou assimilés à des hommes d'affaires ou cadres supérieurs.*
- **(Cond4)** On vérifiera maintenant que le kidnappé est effectivement un homme d'affaires ou un cadre supérieur.
 - **(Règle T0, Conséquent1)** *Dans un contexte de 'kidnapping pour rançon', on cherche à prouver que le kidnappé a une quelque relation de parenté avec une deuxième personne et, dans ce cas,*
 - **(Règle T0, Conséquent2)** *on ira contrôler si cette deuxième personne est 'riche', est un homme d'affaires ou un cadre supérieur.*

5. Conclusions

Dans cet article, nous avons cherché à mettre en évidence deux aspects fondamentaux de NKRL :

- Les capacités analytiques du langage qui garantissent une exhaustivité élevée des descriptions formelles obtenues et une bonne ‘qualité’ de ces descriptions en ce qui concerne leur contenu conceptuel ;
- Les possibilités inférentielles avancées associées aux procédures d’interrogation et d’exploration des bases de connaissances NKRL qui découlent de ces niveaux de précision et d’exhaustivité.

Les exemples qui ont illustré cet article et qui se réfèrent à une application concrète du type ‘défense’ (terrorisme dans les Philippines du Sud) devraient permettre d’entrevoir les possibilités de cette approche pour le traitement d’informations du type défense similaires.

Des comparaisons détaillées avec d’autres approches ‘conceptuelles’ comme les Graphes Conceptuels [9] ou CYC [10] – et une critique de l’approche ‘Web Sémantique’ à cause du bas niveau d’expressivité de ce type d’outils, voir aussi [11] – peuvent être trouvées dans [6] et [12].

Références

- [1] Rinaldi, F., Dowdall, J., Hess, M., Ellman, J., Zarri, G.P., Persidis, A., Bernard, L., and Karanikas, H. (2003). Multilayer Annotations in PARMENIDES. In: *Proceedings of the K-CAP (International Conference on Knowledge Capture) 2003 Workshop on Knowledge Markup and Semantic Annotation* (October 25-26, 2003, Sanibel Island, Florida, USA).
- [2] Black, W.J., Jowett, S., Mavrouidakis, T., McNaught, J., Theodoulidis, B., Vasilakopoulos, A., Zarri, G.P., and Zervanou, K. (2004). Ontology-Enablement of a System for Semantic Annotation of Digital Documents. In: *Proceedings of the 4th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2004) – 3rd International Semantic Web Conference* (November 8, 2004, Hiroshima, Japan).
- [3] Noy, F.N., Ferguson, R.W., and Musen, M.A. (2000). The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility. In: *Knowledge Acquisition, Modeling, and Management – Proceedings of EKAW’2000* (LNCS 1937), Dieng, R., and Corby, O., eds. Berlin: Springer.
- [4] Zarri, G.P. (1998). Representation of Temporal Knowledge in Events: The Formalism, and Its Potential for Legal Narratives. *Information & Communications Technology Law* 7: 213-241.
- [5] Zarri, G.P. (2003). A Conceptual Model for Representing Narratives. In: *Innovations in Knowledge Engineering*, Jain, R., Abraham, A., Faucher, C., and van der Zwaag, eds. Adelaide: Advanced Knowledge International.
- [6] Zarri, G.P. (2005). Integrating the Two Main Inference Modes of NKRL, Transformations and Hypotheses. *Journal on Data Semantics (JoDS)* 4: 304-340.
- [7] Koster, C.H.A. (2004). Head/Modifier Frames for Information Retrieval. In: *Computational Linguistics and Intelligent Text Processing – Proceedings of the 5th International Conference, CICLing 2004*. Berlin: Springer.
- [8] Clark, K.L., and Tärnlund, S.-A., eds. (1982). *Logic Programming*. London: Academic Press.
- [9] Sowa, J.F. (1999). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove (CA): Brooks Cole Publishing Co.
- [10] Lenat, D.B., and Guha, R.V. (1990). *Building Large Knowledge Based Systems*. Reading (MA): Addison-Wesley.
- [11] Zarri, G.P. (2005). An *n*-ary Language for Representing Narrative Information on the Web. In: *SWAP 2005, Semantic Web Applications and Perspectives – Proceedings of the 2nd Italian Semantic Web Workshop* (CEUR-WS.org/Vol-166), Boquet, P., and Tummarello, G., eds. Aachen: Sun SITE Central Europe.
- [12] Zarri, G.P. (2006). Représentation et traitement avancé de documents narratifs complexes au moyen du *Narrative Knowledge Representation Language* (NKRL). Dans : *Le Storytelling. Concepts, outils et applications*, Soulier, E., dir. Paris: Hermès/Lavoisier.