

# Representation and Processing of Complex Events

Gian Piero ZARRI

Université Paris-Est – LISSI Lab.  
120-122, rue Paul Armangot – 94400 Vitry-sur-Seine – France  
zarri@noos.fr, gian\_piero.zarri@univ-paris-est.fr

## Abstract

We supply in this paper some basic information about NKRL, an implemented language/environment for the management of elementary and complex events.

## Introduction

A big amount of important, ‘economically relevant’ information is buried into unstructured, multimedia ‘narrative’ resources. This is true, e.g., for most of the corporate knowledge documents (memos, policy statements, reports, minutes etc.), for the news stories, the normative and legal texts, the medical records, many intelligence messages, the ‘storyboards’ describing accidents in industrial plants, the surveillance videos, the actuality photos for newspapers and magazines, lot of material (text, image, video, sound...) for eLearning etc., as well as, in general, for a huge fraction of the information stored on the Web. In these ‘narrative documents’, or ‘narratives’, *the information content (‘complex events’) consists in the association – thanks to the presence of ‘connective phenomena’ like causality, goal, indirect speech, co-ordination and subordination etc. – of several ‘elementary events’ that describe the real or intended behavior of some ‘actors’, ‘characters’, ‘personages’, etc.* (the term ‘event’ is taken here in its more general meaning, covering also strictly related notions like fact, action, state, situation etc.). These actors try to attain a specific result, experience particular situations, manipulate some (concrete or abstract) materials, send or receive messages, buy, sell, deliver etc.

Note that the actors or personages are not necessarily human beings; we can have narrative documents concerning, e.g., the vicissitudes in the journey of a nuclear submarine (the ‘actor’, ‘subject’ or ‘personage’) or the various avatars in the life of a commercial product. Note also that, even if a large amount of narrative documents concerns natural language (NL) texts, this is not necessarily true, and ‘narratives’ are really ‘multimedia’. A photo representing a situation that, verbalized, could be expressed as “The US President is addressing the

Congress” is not of course an NL text, yet it is still a narrative document.

## Background

Usual ontologies – both in their ‘traditional’ and ‘semantic web’ versions like RDF and OWL – are not very suitable for dealing with (elementary or complex) events.

Basically, ontologies organize the ‘concepts’ – i.e., the important notions to be represented in a given application domain – into a hierarchical structure where the nodes (the concepts) are defined by a set of binary relationships of the ‘property/value’ type (e.g., a ‘frame’). This approach is largely sufficient to provide a *static, a priori* definition of the concepts and of their properties.

Unfortunately, this is no more true when we consider the *dynamic behavior* of the concepts, i.e., we want to describe their mutual relationships when they take part in some concrete action, situation etc. (‘events’). First of all, representing an (elementary) event implies that the notion of ‘role’ must be added to the traditional generic/specific and property/value representational principles. If we want to represent adequately a narrative fragment like “An important media company ... will develop a new model of Internet cellular phone...”, besides asserting that MEDIA\_COMPANY\_1 is an instance of the concept media\_company and that we must introduce a specific instance of a concept like internet\_cellular\_phone, we have also to create a sort of ‘threefold’ relationship including a ‘predicate’ (like DEVELOP or PRODUCE), the two instances, and a third fundamental component, the ‘roles’ (like SUBJECT or AGENT for MEDIA\_COMPANY\_1 and OBJECT or PATIENT for the new cellular phone). These last will be used to specify the exact function of the two instances within the formal description of the event. Moreover, in a ‘complex events’ context, we must also handle those ‘connectivity phenomena’ already mentioned that link together the basic ‘elementary events’. It is very likely, in fact, that, dealing with the sale of a subsidiary, the global information to represent is something like: “Company X has sold its subsidiary Y to Z *because* the profits of Y have fallen dangerously these last years *due to* a lack of investments” or, returning to our previous example, that “MEDIA\_COMPANY\_1 will develop a new model of Internet cellular phone *to reply* to a direct competitor move” or that, dealing with the relationships

between companies in the biotechnology domain, “*V* made a milestone payment to *W* because they decided to pursue an in vivo evaluation of the candidate compound identified by *W*”, etc. In Computational Linguistics terms, we are here in the domain of the ‘Discourse Analysis’ which deals, in short, with the two following problems: i) determining the nature of the information that, in a sequence of statements, *goes beyond the simple addition of the information conveyed by a single statement*; ii) determining the *influence of the context* in which a statement is used on the meaning of this individual statement, or part of it.

It is now easy to imagine the awkward proliferation of *binary relationships* that, sticking to the usual ontological paradigm, it would be necessary to introduce to approximate notions like those of ‘role’ and ‘connectivity phenomena’, see in this context (Zarri, 2005a).

Solutions for representing narratives in computer-usable ways that *go beyond a strict ‘binary’ framework* have already been proposed in the past. In the context of his work – between the mid-fifties and the mid-sixties – on the set up of a mechanical translation process based on the simulation of the thought processes of the translator, Silvio Ceccato (Ceccato, 1961) proposed a representation of narrative-like sentences as a network of triadic structures (‘correlations’) organized around specific ‘correlators’ (a sort of roles). Correlators included conjunctions and prepositions, punctuation marks, and syntactic/semantic relationships like subject-predicate, substance-accident, apposition, development-modality, comparison etc. Ceccato is also credited to be one of the pioneers of the semantic network studies. In the seventies, a sort of particularly popular semantic network approach has been represented by the Conceptual Dependency theory of Roger Schank (Schank, 1972). In this theory, the underlying meaning (‘conceptualization’) of narrative-like utterances was expressed as combinations of ‘semantic predicates’ chosen from a set of twelve ‘primitive actions’ (like INGEST, MOVE, ATRANS, the transfer of an abstract relationship like possession, ownership and control, PTRANS, physical transfer, etc.) plus states and changes of states, and seven role relationships (‘conceptual case’) in the Case Grammar style. Conceptual Graphs (CGs) is the representation system developed by John Sowa (Sowa, 1999). CGs make use of a graph-based notation for representing ‘concept-types’ (organized into a type-hierarchy), ‘concepts’ (which are instantiations of concept types) and ‘conceptual relations’ that relate one concept to another. CGs can be used to represent in a formal way narratives like “A pretty lady is dancing gracefully” and more complex, second-order constructions like contexts, wishes and beliefs. An interesting, recent commercial system making use of a CG-like form of representation is SONETTO, a rule-based, business Product Information Management system currently used for retail management, see (Sarraf and Ellis, 2006). Started in the early '80 as a MCC project CYC, see (Lenat *et al.*, 1990), ended about 15 years later with the set up of an enormous knowledge

base containing about a million of hand-entered ‘logical assertions’ including both simple statements of facts and rules about what conclusions can be inferred if certain statements of facts are satisfied. The ‘upper level’ of the ontology that structures the CYC knowledge base is accessible on the Web, see [www.cyc.com/cyc/opencyc](http://www.cyc.com/cyc/opencyc). An analysis of the origins, developments and motivations of CYC can be found in (Bertino *et al.*, 2001: 275-316).

## Narrative Knowledge representation Language (NKRL)

With the exception of CYC and of some CGs applications, many of the solutions evoked at the end of the previous Section concern pure academic work implying sketchy forms of implementation. NKRL, Narrative Knowledge Representation Language (Zarri, 2003; 2005b; 2009), represents in contrast a *fully implemented solution* to the problem of representing complex events without a too important loss of the original ‘meaning’. See (Zarri, 2009: Appendix A) for details about the two versions, ‘file-oriented’ and ‘Oracle-supported’ of the NKRL software, and (Zarri, 2000) for the (inevitably idiosyncratic) ‘translation’ of NKRL into RDF (binary) terms.

NKRL innovates with respect to the usual ontology paradigm by associating with the traditional, ‘binary’ ontology of concepts an ‘ontology of events’, i.e., a new sort of hierarchical organization where the nodes correspond to ‘*n*-ary’ structures called ‘templates’. Instead of using the usual *object (class, concept) – attribute – value* organization, templates are generated from the combination of quadruples where each of them connect together the *symbolic name* of the template, a *predicate*, and the *arguments* of the predicate introduced by named relations, the *roles*. The quadruples have in common the ‘name’ and ‘predicate’ components. If we denote then with  $L_i$  the generic symbolic label identifying a given template, with  $P_j$  the predicate used in the template, with  $R_k$  the generic role and with  $a_k$  the corresponding argument, the NKRL core data structure for templates has the following general format:

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

see the example in Table 1 below. Predicates pertain to the set {BEHAVE, EXIST, EXPERIENCE, MOVE, OWN, PRODUCE, RECEIVE}, and roles to the set {SUBJ(ect), OBJ(ect), SOURCE, BEN(e)F(iciary), MODAL(ity), TOPIC, CONTEXT}. An argument of the predicate can consist of a simple ‘concept’ (according to the traditional, ‘ontological’ meaning of this word) or of a structured association (‘expansion’) of several concepts. The NKRL standard ‘ontology of concepts’ is called HClass, ‘hierarchy of classes’.

Templates are included into an inheritance hierarchy, HTemp(lates), which implements then the new ‘ontology of events’, see again Figure 1 below. They represent formally *generic classes of elementary events* like ‘move a

physical object’, ‘be present in a place’, ‘produce a service’, ‘send/receive a message’, ‘build up an Internet site’, etc. When a particular elementary event pertaining to one of these general classes must be represented, the corresponding template is ‘*instantiated*’ to produce what, in the NKRL’s jargon, is called a ‘*predicative occurrence*’.

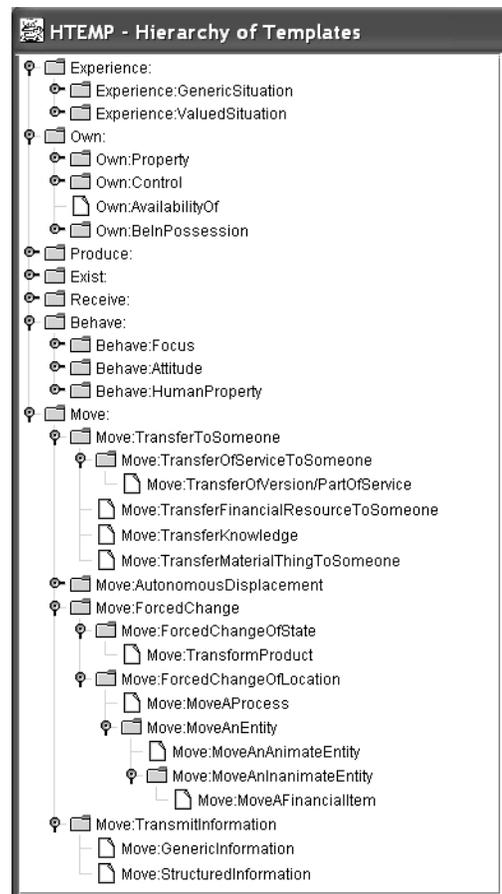
To represent then a simple narrative like: “British Telecom will offer its customers a pay-as-you-go (payg) Internet service in autumn 1998”, we must select firstly in HTemp the template corresponding to ‘supply a service to someone’, represented in the upper part of Table 1.

**Table 1:** Deriving a predicative occurrence from a template.

<i>name:</i> Move:TransferOfServiceToSomeone		
<i>father:</i> Move:TransferToSomeone		
<i>position:</i> 4.11		
<i>natural language description:</i> 'Transfer or Supply a Service to Someone'		
MOVE	SUBJ	var1: [var2]
	OBJ	var3
	[SOURCE	var4: [var5]]
	BENF	var6: [var7]
	[MODAL	var8]
	[TOPIC	var9]
	[CONTEXT	var10]
		{[modulators]}
var1 = human_being_or_social_body		
var3 = service_		
var4 = human_being_or_social_body		
var6 = human_being_or_social_body		
var8 = process_, sector_specific_activity		
var9 = sortal_concept		
var10 = situation_		
var2, var5, var7 = geographical_location		
c1) MOVE SUBJ BRITISH_TELECOM		
	OBJ	payg_internet_service
	BENF	(SPECIF customer_
		BRITISH_TELECOM)
	date-1:	after-1-september-1998
	date-2:	

This template is a specialization of the particular MOVE template of HTemp corresponding to ‘transfer of resources to someone’ – Figure 1 below reproduces a fragment of the ‘external’ organization of HTemp that includes some offsprings of Move:TransferToSomeone.

In a template, the arguments of the predicate (the  $a_k$  terms in (1)) are represented by variables with associated *constraints* – which are expressed as concepts or combinations of concepts using the terms of the NKRL standard ‘ontology of concepts’ (HClass). The constituents (as SOURCE in Table 1) included in square brackets are optional. When deriving a *predicative occurrence* (an *instance of a template formalizing an elementary event*) like c1 in Table 1, the role fillers in the occurrence *must conform to the constraints of the father-template*. In c1, e.g., BRITISH\_TELECOM is an individual, instance of company\_ (i.e., of a concept specialization of human\_being\_or\_social\_body); payg\_internet\_service is a specialization of the concept service\_, specific term of social\_activity, etc.



**Figure 1:** ‘MOVE’ etc. branch of the HTemp hierarchy.

The meaning of the expression “BENF (SPECIF customer\_ BRITISH\_TELECOM)” in c1 is self-evident: the beneficiaries (role BENF) of the service are the customers of – SPECIF(ication) – British Telecom. The ‘attributive operator’, SPECIF(ication), is one of the four operators that make up the AECS sub language, used for the set up of the structured arguments (expansions), see (Zarri, 2003). In the occurrences, the two operators date-1, date-2 materialize the temporal interval associated the elementary events; a description of the way of representing temporal data in NKRL can be found in (Zarri, 2009: 80-86, 194-201).

More than 150 templates are permanently inserted into HTemp; HTemp, the NKRL ontology of events, corresponds then to a sort of ‘catalogue’ of narrative formal structures, that are very easy to ‘customize’ in order to derive the new templates that could be needed for a particular application. This approach is particularly advantageous for practical applications, and it implies, in particular, that: i) a system-builder does not have to create himself the structural knowledge needed to describe the events proper to a (sufficiently) large class of narrative documents; ii) it becomes easier to secure the reproduction or the sharing of previous results.

What we have expounded until now illustrates the NKRL solutions to the problem of providing a *coherent*

and complete representation of elementary events. To deal now with the ‘connectivity phenomena’, we make use of second order structures created through reification of the conceptual labels of predicative occurrences; the resulting structures correspond then to the formal representation of ‘complex events’. For example, the ‘binding occurrences’ are NKRL constructions that consist of lists of symbolic labels ( $c_i$ ) of predicative occurrences; the lists are differentiated making use of specific binding operators like GOAL and CAUSE see, e.g., (Zarri, 2003).

Returning now to the Table 1 example, let us suppose we would now state that: “We can note that, on March 2008, British Telecom *plans to offer* to its customers, in autumn 1998, a pay-as-you-go (payg) Internet service...”, where the specific elementary event corresponding to the offer is still represented by occurrence  $c_1$  in Table 1. To encode correctly the new information, we must introduce first an *additional predicative occurrence* labelled as  $c_2$ , see Table 2, meaning that: “at the specific, particular date associated with  $c_2$  (March 1998), it can be noticed, modulator *obs(erve)*, that British Telecom is planning to act in some way”. *obs(erve)* is a ‘temporal modulator’, see (Zarri, 2003), used in NKRL to identify a *specific timestamp* where a specific event can be detected. We will eventually add a *binding occurrence*  $c_3$  to link together the conceptual labels  $c_2$  (the planning activity) and  $c_1$  (the intended result). The global meaning of  $c_3$  (the complex event) is then: “the activity described in  $c_2$  is focalized towards (GOAL) the realization of  $c_1$ ”. Note also that, to respect the semantics of the GOAL operator,  $c_2$  is now characterized by the presence of an *uncertainty attribute code*, “\*”, to specify that, at the time of planning (March 1998), the reality of the offer cannot be absolutely confirmed.

**Table 2:** Binding and predicative occurrences.

c2)	BEHAVE	SUBJ MODAL { obs } date1: date2:	BRITISH_TELECOM planning_  march-1998
Behave:ActExplicitly (1.12)			
*c1)	MOVE	SUBJ OBJ BENF  date-1: date-2:	BRITISH_TELECOM payg_internet_service (SPECIF customer_ BRITISH_TELECOM)  after-1-september-1998
Move:TransferOfServiceToSomeone (4.11)			
c3)	(GOAL	c2	c1)

## General principles about ‘reasoning’ in NKRL

Reasoning in NKRL ranges from the *direct questioning* of a knowledge base of narratives represented according to the NKRL format – by means of *search patterns* that unify

information in the base thanks to the use of a *Filtering Unification Module (Fum)* – to *high-level inference procedures*. These last – implemented as ‘rules’ working according to the ‘inference by resolution’ principles see, e.g., (Bertino *et al.*, 2001: 107-121) – make use of the richness of the representation system to automatically establish ‘interesting’ relationships among the narrative items and/or the ‘personages’ mentioned within them.

The NKRL rules are characterised by the following general properties:

- All of them can be conceived as *implications* of the type:
 
$$X \text{ iff } Y_1 \text{ and } Y_2 \dots \text{ and } Y_n \quad (2)$$
- In (2),  $X$  is a *predicative occurrence* or a *search pattern* and  $Y_1 \dots Y_n$  – the NKRL translation of the ‘reasoning steps’ that make up the rule – correspond to *partially instantiated templates*. They include then, see Table 1 above, explicit variables of the form *var*.
- According to the ‘inference by resolution’ principle, the NKRL *InferenceEngine* will understand each implication as a *procedure*, reducing then the ‘problems’ of the form  $X$  to a succession of ‘sub-problems’ of the form  $Y_1$  and  $\dots Y_n$ .
- Each  $Y_i$  is interpreted in turn as a *procedure call* that try to convert – using, in case, *backtracking procedures* –  $Y_i$  into (at least) a *successful search pattern*  $p_i$ . These last should be able to *unify one or several of the occurrences*  $c_j$  of the NKRL knowledge base.
- The success of the unification operations of the pattern  $p_i$  derived from  $Y_i$  means that the ‘reasoning step’ represented by  $Y_i$  *has been validated*. *InferenceEngine* continues then its work trying to validate the reasoning step corresponding to the sub-problem  $Y_{i+1}$ .
- In line with the presence of the operator ‘and’ in (2), the implication represented by (2) is fully validated *iff all the reasoning steps*  $Y_1, Y_2 \dots Y_n$  *are validated*.

All the unification operations  $p_i/c_j$  make use only of the unification functions supplied by the *Filtering Unification Module (Fum)* already mentioned. Apart from being used for the direct questioning operations, *Fum* constitutes as well, therefore, the ‘inner core’ of *InferenceEngine*.

NKRL high-level inference procedures concern *mainly* two classes of rules, ‘transformations’ and ‘hypotheses’, see (Zarri, 2005b).

Let us consider, e.g., the ‘transformations’. These rules try to ‘adapt’, from a *semantic* point of view, a search pattern  $p_i$  that ‘failed’ (that was unable to find an unification within the knowledge base) to the *real contents* of this base making use of a sort of ‘*analogical reasoning*’. In a transformation context, the ‘head’  $X$  of formula (2) is then represented by a search pattern,  $p_i$ . The transformation rules try to *automatically ‘transform’*  $p_i$  into one or more *different*  $p_1, p_2 \dots p_n$  that are not strictly ‘equivalent’ but only ‘*semantically close*’ to the original one. For example, let us suppose that, in the context of a recent NKRL application about ‘Southern Philippine terrorism’, see

(Zarri, 2005b), we ask: “Search for the existence of links between ObL (a well-known ‘terrorist’) and Abubakar Abdurajak Janjalani, the leader of the Abu Sayyaf group (a separatist group in Southern Philippines)”. In the absence of a direct answer, the corresponding search pattern can be transformed into: “Search for the attestation of the transfer of economic/financial items between the two”. This could lead to retrieve: “During 1998/1999, Abubakar Abdurajak Janjalani has received an undetermined amount of money from ObL through an intermediate agent”.

For clarity’s sake, it can be useful to denote the transformation rules as made up of a *left-hand side*, the ‘*antecedent*’ – i.e. the formulation, in search pattern format, of the ‘query’ to be transformed – and one or more *right-hand sides*, the ‘*consequent(s)*’ – the NKRL representation(s) of one or more queries (search patterns) to be substituted for the given one. Denoting then with *A* the antecedent and with *Cs* all the possible consequents, a transformation rule can be expressed as:

$$A(var_i) \Rightarrow Cs(var_j), \quad var_i \subseteq var_j \quad (3)$$

With respect then to formula (2) above, *X coincides* now with *A* – operationally, a *search pattern* – while the reasoning steps  $Y_1, Y_2 \dots Y_n$  are used to produce the *search pattern(s) Cs to be used in place of A*. The restriction  $var_i \subseteq var_j$  – all the variables declared in the antecedent *A must also appear in Cs* accompanied, in case, by *additional variables* – has been introduced to assure the logical congruence of the rules. The ‘transformation arrow’ of (3), ‘ $\Rightarrow$ ’, has a *double meaning*:

- *Operationally speaking, the arrow indicates the direction of the transformation.* The original search pattern (which is a specialisation of the left-hand side *A* of the transformation rule) is then *removed* and *replaced* by one or several new search patterns obtained through the updating, using the parameters of the original pattern, of the right-hand side *Cs*.
- *From a logical/semantic point of view, we assume that between the information retrieved through Cs and the information we wanted to obtain through an instantiation of A there is a sort of implication relationship.* Normally, this relationship denotes solely a possible (a *weak*) implication.

More formal details are given, e.g., in (Zarri, 2003). A representation of the above ‘financial transfer’ transformation is reproduced in Table 3. Many of the transformation rules are characterized by the simple format of Table 3 implying only one ‘consequent’. This is not true in general: examples of ‘multi-consequent transformations’ can be found in (Zarri, 2005) – and in Table 5 below.

With respect now to the *hypothesis rules*, these allow us to build up automatically a sort of ‘*causal explanation*’ for an elementary (a predicative occurrence  $c_j$ ) retrieved within a NKRL knowledge base using *Fum* and a search-pattern in a querying-answering mode. In a hypothesis context, the ‘head’ *X* of formula (2) is then represented by a predicative

occurrence,  $c_j$ . Accordingly, the ‘reasoning steps’  $Y_i$  of (2) – called ‘condition schemata’ in a hypothesis context – *must all be satisfied* (for each of them, at least one of the corresponding search patterns  $p_i$  must find a successful unification with the predicative occurrences of the base) *in order that the set of  $c_1, c_2 \dots c_n$  predicative occurrences retrieved in this way can be interpreted as a context/causal explanation of the original occurrence  $c_j$* . For example, to mention a ‘classic’ NKRL example, let us suppose we have directly retrieved, in a querying-answering mode, an information like: “Pharmacoepia, an USA biotechnology company, has received 64,000,000 dollars from the German company Schering in connection with an R&D activity” that corresponds then to  $c_j$ . We can then be able to automatically construct, using a ‘hypothesis’ rule, a sort of ‘causal explanation’ of this event by retrieving in the knowledge base information like: i) “Pharmacoepia and Schering have signed an agreement concerning the production by Pharmacoepia of a new compound” ( $c_1$ ) and ii) “in the framework of the agreement previously mentioned, Pharmacoepia has actually produced the new compound” ( $c_2$ ).

**Table 3.** A simple example of ‘transformation’ rule.

<i>t1</i> ) BEHAVE	SUBJ	(COORD1 <i>var1 var2</i> )	$\Rightarrow$
	OBJ	(COORD1 <i>var1 var2</i> )	
RECEIVE	SUBJ	<i>var2</i>	
	OBJ	<i>var4</i>	
	SOURCE	<i>var1</i>	
	MODAL	<i>var3</i>	
<i>var1</i>	=	human_being_or_social_body	
<i>var2</i>	=	human_being_or_social_body	
<i>var3</i>	=	business_agreement, mutual_relationship	
<i>var4</i>	=	economic/financial_entity	

*To verify the existence of a relationship or of a business agreement between two (or more) people, try to see if one of these people has received a ‘financial entity’ (e.g., money) from the other.*

In Table 4, we give now the informal description of the reasoning steps (hypothesis’ ‘condition schemata’) to be validated to prove that – in the context of the already mentioned application about ‘Southern Philippine terrorism’ – a generic ‘kidnapping’ corresponds, in reality, to a more precise ‘kidnapping for ransom’. When many reasoning steps must be *simultaneously* validated, as usual in a hypothesis context, a failure is always possible. To overcome this problem – and, at the same time, discover all the possible *implicit information* associated with the original data – the two inference modes, transformation and hypotheses, can also be used in an *integrated way*, see again (Zarri, 2005). In practice, we make use of ‘transformations’ within a ‘hypothesis’ inference environment. This means that, whenever a ‘search pattern’ is derived from a ‘condition schema’ of a hypothesis to implement, using *Fum*, one of the steps of the reasoning process, we can use it ‘as it is’ – i.e., as originally coded when the inference rule has been built up – but also in a

'transformed' form if the appropriate transformation rules exist within the system.

**Table 4.** Inference steps for the 'kidnapping for ransom' hypothesis

(Cond1)	The kidnapers are part of a separatist movement or of a terrorist organization.
(Cond2)	This separatist movement or terrorist organization currently practices ransom kidnapping of particular categories of people.
(Cond3)	In particular, executives or assimilated categories are concerned.
(Cond4)	It can be proved that the kidnapped is really a businessperson or assimilated.

Making use of transformations, the hypothesis represented in an informal way in Table 4 becomes, in practice, *potentially equivalent* to the hypothesis of Table 5. For example, the proof that the kidnapers are part of a terrorist group or separatist organization (reasoning step Cond1 of Table 4) can be now obtained *indirectly*, transformation T3, by checking whether they are members of a specific subset of this group/organization. Note that transformations T2 and T6 imply only one step of reasoning, whereas all the residual transformations are 'multi-consequent'.

## Conclusion

NKRL deals with the representation and management of 'events' (both 'elementary' and 'complex' events) making use of *n*-ary and second order knowledge representation structures. One of its main characteristics concerns the addition of an *ontology of events* (a catalogue of formalized representation of characteristic situations and events) to the usual *ontology of concepts*. Its inference solutions employ causal- and analogical-based reasoning techniques to deal in an advanced way with the events and their relationships. Several successful applications in the most different domains (from 'terrorism' to 'corporate domain' to the management of 'storyboards/historians' for the gas/oil industry...) have proved the possible, practical utility of this tool.

## References

- Bertino, E., Catania, B., and Zarri, G.P. (2001). *Intelligent Database Systems*. London: Addison-Wesley and ACM Press.
- Ceccato, S., ed. (1961). *Linguistic Analysis and Programming for Mechanical Translation* (Technical report RADC-TR-60-18). Milano: Feltrinelli.
- Lenat, D.B., and Guha, R.V. (1990). *Building Large Knowledge Based Systems*. Reading (MA): Addison-Wesley.
- Schank, R.C. (1973). Identification of Conceptualizations Underlying Natural Language. In: *Computer Models of Thought and Language*, Schank, R.C., and Colby, K.M., eds. San Francisco: W.H. Freeman and Co.

- Sarraf, Q., and Ellis, G. (2006). Business Rules in Retail: The Tesco.com Story. *Business Rules Journal* 7(6): <http://www.BRCommunity.com/a2006/n014.html>.
- Sowa, J.F. (1999). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove (CA): Brooks Cole Publishing Co.
- Zarri, G.P. (2000). A Conceptual Model for Capturing and Reusing Knowledge in Business-Oriented Domains. In: *Industrial Knowledge Management: A Micro-level Approach*, Roy, R., ed. London: Springer.
- 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.
- Zarri, G.P. (2005a). An *n*-ary Language for Representing Narrative Information on the Web. In: *SWAP 2005, Semantic Web Applications and Perspectives – Proceedings of the 2<sup>nd</sup> Italian Semantic Web Workshop* (CEUR-WS.org/Vol-166), Boquet, P., and Tummarello, G., eds. Aachen: Sun SITE.
- Zarri, G.P. (2005b). Integrating the Two Main Inference Modes of NKRL, Transformations and Hypotheses. *Journal on Data Semantics (JoDS)* 4: 304-340.
- Zarri, G.P. (2009). *Representation and Management of Narrative Information – Theoretical Principles and Implementation*. London: Springer.

**Table 5.** 'Kidnapping' hypothesis in the presence of transformations concerning the intermediary inference steps.

- |         |   |
|---------|---|
| (Cond1) | The kidnapers are part of a separatist movement or of a terrorist organization.   |
|         | – (Rule T3, Consequent1) <i>Try to verify whether a given separatist movement or terrorist organization is in strict control of a specific sub-group and, in this case,</i> |
|         | – (Rule T3, Consequent2) <i>check if the kidnapers are members of this sub-group. We will then assimilate the kidnapers to 'members' of the movement or organization.</i>   |
| (Cond2) | This movement or organization practices ransom kidnapping of given categories of people.  |
|         | – (Rule T2, Consequent) <i>The family of the kidnapped has received a ransom request from the separatist movement or terrorist organization.</i>                            |
|         | – (Rule T4, Consequent1) <i>The family of the kidnapped has received a ransom request from a group or an individual person, and</i>   |
|         | – (Rule T4, Consequent2) <i>this second group or individual person is part of the separatist movement or terrorist organization.</i>  |
|         | – (Rule T5, Consequent1) <i>Try to verify if a particular sub-group of the separatist movement or terrorist organization exists, and</i>                                    |
|         | – (Rule T5, Consequent2) <i>check whether this particular sub-group practices ransom kidnapping of particular categories of people.</i>                                     |
|         | – ...   |
| (Cond3) | In particular, executives or assimilated categories are concerned.  |
|         | – (Rule T0, Consequent1) <i>In a 'ransom kidnapping' context, we can check whether the kidnapped person has a strict kinship relationship with a second person, and</i>     |
|         | – (Rule T0, Consequent2) <i>(in the same context) check if this second person is a businessperson or assimilated.</i>   |
| (Cond4) | It can be proved that the kidnapped person is really an executive or assimilated.   |
|         | – (Rule T6, Consequent) <i>In a 'ransom kidnapping' context, 'personalities' like physicians, journalists, artists etc. can be assimilated to businesspersons.</i>          |