An *n*-ary Language for Representing Narrative Information on the Web

Gian Piero Zarri

LaLICC, University Paris 4/Sorbonne – Maison de la Recherche
28, rue Serpente – 75006 Paris, France
gpzarri@paris4.sorbonne.fr, zarri@noos.fr

Abstract. In this paper, we evoke first the ubiquity and the importance of the so-called 'narrative' information, showing that the usual ontological tools, both the 'traditional' and the 'Semantic Web' ones, are unable to offer complete and reliable solutions for representing and exploiting this type of information. We supply then some details about NKRL (Narrative Knowledge Representation Language), a knowledge representation and inferencing environment especially created for an 'intelligent' exploitation of narrative knowledge.

1 Introduction

A big amount of important, 'economically relevant' information, is hidden within the huge mass of multimedia documents that correspond to some form of 'narrative' description. Examples of 'narrative' documents are the 'corporate knowledge' documents (memos, policy statements, reports, minutes etc.), the news stories, the normative and legal texts, the medical records, many intelligence messages, as well as, in general, a huge fraction of the documents stored on the Web. In these narrative documents, or 'narratives', the main part of the information content consists in the description of temporally and spatially bounded 'events' that relate the behaviour or the condition of some 'actors' (characters, personages, etc.): they try to attain a specific result, experience particular situations, manipulate some (concrete or abstract) materials, send or receive messages, buy, sell, deliver etc.

For simplicity's sake, the term 'event' is taken here in its most general meaning, covering also strictly related notions like fact, action, state, situation, episode, activity etc., see [1]. Note that, in the events evoked by the narrative documents, 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. A photo representing a situation that, verbalized, could be expressed as "Three nice girls are lying on the beach" is not of course an NL text, yet it is still a narrative document. Because of the ubiquity of these 'narrative' resources, being able to represent in a general, accurate, and effective way their semantic content – i.e., their key 'meaning' – is then both conceptually relevant and economically important.

In this paper, we present the main properties of NKRL ('Narrative Knowledge Representation Language'), see [2, 3], a language expressly designed for representing, and then 'intelligently' exploiting, the 'meaning' of complex multimedia narrative documents. In the next Section, "Narratives and Knowledge Representation", we will examine previous and current solutions proposed for the representation and processing of narratives. The following Section, "The NKRL Approach", will describe some of the NKRL's solutions, mentioning in particular the inference techniques proper to this language. A short 'Conclusion' will end the paper.

2 Narratives and Knowledge Representation

2.1 The 'Standard' Ontology Approach and the '*n*-ary' Problem

Usual ontologies – both in their 'traditional', see [4], and 'Semantic Web (W3C)' versions, see [5] – are not very suitable for dealing with narratives. Basically, ontologies organize the 'concepts' – that we can identify here with the important notions to be represented in a given application domain – into a hierarchical structure, able to supply an elementary form of definition of these concepts through their mutual generic/specific relationships ('IsA' links). A more precise definition is obtained by associating with them a set of *binary relationships* of the 'property/value' type (e.g., a 'frame'). Semantic Web (W3C) languages like RDF and OWL – and the great majority of the languages/environments for setting up 'ontologies' – are then denoted as '*binary*'. The combination of these two representational principles is largely sufficient to provide a *static*, *a priori* definition of the concepts and of their properties.

Unfortunately, this is not true when we consider the *dynamic behaviour* of the concepts, i.e., we want to describe their *mutual relationships* when they take part in some concrete action, situation etc. ('events'), see the very simple narrative "John has given a book to Mary". In this example, "give" is now an *n-ary (ternary) relationship* that, to be represented in a *complete and unambiguous way*, asks for some form of complex syntax where the arguments of the predicate, i.e., "John", "book" and "Mary", are introduced by some sorts of '*conceptual roles*' such as, e.g., "agent of give", "object of give" and "beneficiary of give" respectively. For representing the 'meaning' of narrative documents, the notion of '*role*' must then be necessarily added to the traditional 'generic/specific' and 'property/value' representational principles in order to specify the *exact function* of the different components of an event within the *formal description* of this event. Moreover, in a narrative context, we must also take care of those '*connectivity phenomena*' like causality, goal, indirect speech, coordination and subordination etc., that link together the basic 'elementary events'. It is very likely, in fact, that, dealing with the sale of a company, 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 "John gave a book to Mary yesterday *as a* present for her birthday" or that, dealing with the relationships between companies in the biotechnology domain, "X made a milestone payment to Y *because* they decided to pursue an in vivo evaluation of the candidate compound identified by X", etc. We are then 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 traditional ontological paradigm, it would be necessary to introduce to approximate high-level notions like those of 'role' and 'connectivity phenomena'. The 'binary' approach to the set up of ontologies is then insufficient to deal with narrative documents.

Note that a common misunderstanding consists in saying that the definition of specific *n*-ary languages for managing narratives is not at all necessary given that any *n*-ary relationship, with $n > 2$, can be reduced in a very simple way to a set of binary relationships. More formally – and leaving aside, for simplicity's sake, problems like those introduced by temporal information – we can say that an *n*-ary relation $R(t_1, ..., t_n)$ can normally be represented, with the aid of the existential quantifier, as: (exists e) $(R(e)$ & $Rb_1(e, t_1)$ & $Rb_2(e, t_2)$ & ... & $Rb_n(e, t_n))$. In this last expression, e must be understood as an event or situation of type R; in a triadic situation involving a predicate like "give", we have then GIVE(e). Rb_1, Rb_2 ... Rb_n is some fixed set of binary relations, corresponding to "agent of give" etc. in the previous example. This sort of decomposition is not only formally interesting, but also important for many practical problems, e.g., for storing efficiently *n*-ary relationships into standard databases.

However, the decomposition *does not eliminate* the need for argument e, which is still necessary to link all the binary relationships together. In a relational databases context, this is equivalent to say that – after having decomposed a relationship in the GIVE style into three 2-columns tables named "agent", "object" and "beneficiary" where the first column is reserved to the predicate – to recover the *global information* it is now necessary to '*join*' again the three 2-column table on the column that represents the predicate (GIVE). This implies also that, if we want to execute some 'reasoning' about "John gave a book to Mary yesterday" by respecting the true 'meaning' of this event, recognizing the existence of binary relationships between a given book or a human being and this event is not really useful without i) 'returning' to the original, ternary meaning of GIVE, and ii) taking into account that the relationships among GIVE and its three 'arguments', "John", "book" and "Mary" are labelled in different ways, as "agent", "object" and "beneficiary". Now, with respect to the current vogue of 'translating' any pre-existing high-level (*n*-ary) knowledge representation language into the (more fashionable) W3C (binary) languages, an important point to emphasize here is that, notwithstanding the formal transformation, *the* n-*ary languages are still* n-*ary after that the 'translation' has been realized*, like the GIVE relationship above is still ternary even if formally reduced to a set of binary relationships. This means that, to exploit in full the representational power of *n*-ary languages like NKRL, Conceptual Graphs [6] or CycL [7] – e.g., for executing complex inference operations – the original *n*-ary inferencing tools of those languages must be used. As a consequence, these languages – after a possible conversion into binary format – must in fact be 'translated back' into the original format *n*-ary to get the best of their characteristics and to avoid the usual 'binary' limitations.

The discussion above should have demonstrated the need of disposing of 'true' *n*-ary language to deal adequately with narrative documents. Unfortunately, suggestions in this context are quite scarce in a Semantic Web (W3C) context. The most recent one consists in a working paper from the W3C Semantic Web Best Practices and Deployment Working Group (SWBPD WG) about "Defining N-ary Relations on the Semantic Web", see [8]. That paper proposes some extensions to the binary paradigm to allow the correct representation of narratives like: "Christine has breast tumour with high probability", "Steve has temperature, which is high, but failing", "John buys a 'Lenny the Lion' book from books.Example.com for $15 as a birthday gift" or "United Airlines flight 3177 visits the following airports: LAX, DFW, and JFK". This working paper has aroused many criticisms. Leaving aside the fact that only four, very particular 'narratives' are examined, without any convincing justification for that choice, the criticisms have focused mainly on the arbitrary introduction, through reification processes, of fictitious (and inevitably *ad hoc*) 'individuals' to represent the *n*-ary relations – note that this one is the usual (*ad hoc*) solution used by the builders of 'classical' binary ontology to simulate some simple *n*-ary situations. More specific and 'technical' remarks have been also formulated, about, e.g., the use in the proposed *n*-ary solutions of 'infamous' W3C constructs like the RDF 'blank nodes', aka 'anonymous resources'. We can also note that the paper says nothing about the way of dealing, in concrete situations, with those 'connectivity phenomena' evoked above.

2.2 Previous '*n*-ary' Solutions and Related Problems

A well-formed and complete solution to the *n*-ary problem has been long known: it is based on the notions of 'conceptual predicate' and, as already stated, of 'conceptual role'. Returning then to the "John gave a book…" example above, a complete representation that captures all the 'meaning' of this elementary narrative amounts to:

– Defining john_, mary_, book_1 and yesterday_ as 'individuals', instances of general 'concepts' like *human_being*, *information_support* and *calendar_day* or of more specific concepts. Concepts and instances (individuals) are, as usual, collected into a 'binary' ontology (built up using a tool like, e.g., Protégé).
– Defining an *n*-ary structure organised around a conceptual predicate like, e.g., PHYSICAL_TRANSFER, and associating the above individuals (the arguments) with the predicate through the use of conceptual roles that specify their 'function' within the global narrative. john_ will then be introduced by an AGENT (or SUBJECT) role, book_1 by an OBJECT (or PATIENT) role, mary_ by a BENEFICIARY_ role, yesterday_ by a TEMPORAL_ANCHOR role.

Formally, an *n*-ary structure defined as above can be described as:

$$(L_i\ (P_j\ (R_1\ a_1)\ (R_2\ a_2)\ \ldots\ (R_n\ a_n)))\ , \qquad (1)$$

where L_i is the symbolic label identifying the particular *n*-ary structure (e.g., that corresponding to the "John gave a book…" example), P_j is the conceptual predicate, R_k is the generic role and a_k the corresponding argument (the individuals john_, mary_ etc.). Note that if, in the binary decomposition $Rb_i(e, Arg_i)$ introduced above, we equate: e with P_j, Rb_i with R_k, and Arg_i with a_k, we obtain a set of binary relationships that coincide with the (R_i a_i) cells of (1) taken *individually*. As already stated, the whole conceptual structure represented by (1) must be considered *globally*.

Many solutions that can be reduced formally to (1) have been suggested in the last sixty years. To limit ourselves to some 'Knowledge Representation-oriented' examples, we can mention here Silvio Ceccato's 'correlators' (a sort of roles) used, in the context of his experiments of Mechanical Translation, to represent narratives as a network of triadic structures, see [9]. Other examples concern the Conceptual Dependency theory of Roger Schank [10], Schubert's Propositional Notation [11], Sowa's Conceptual Graphs [6, 12], NKRL, etc. Linguistic theories that make use of notions similar to that of 'role' are Case Grammars [13], Jackendoff's thematic roles [14], Kamp's Discourse Representation Structures [15], etc. Always in a 'linguistic' context, a recent trend concerns the fact of dealing with 'events' like 'grammatical objects', see [16], in the sense that the semantics of events can be seen to depend from verbs' syntactic structures. The 'story trees' of Mani and Pustejovsky see, e.g., [17], introduced from a Computational Linguistics and Text Summarization perspective have strong similarities with the 'binding structures' of NKRL used to deal with the already mentioned 'connectivity phenomena'. Eventually, in a Description Logics context, a language called DLR that supports *n*-ary relations has been defined, see [18] for details about its use for reasoning on UML (Unified Modeling Language) class diagrams. The basic conceptual elements of DLR are 'concepts' and '*n*-ary relations'.

We can then wonder why we have to re-discuss the *n*-ary problem again. The answer lies in the *combinatorial explosion* linked with all the possible associations among predicates, roles and arguments that arise when passing from binary to *n*-ary expressions like (1). The situation gets also worse when we consider that:

– No universal agreement exists on the list of roles.
– Predicates are primitives in Schank's Conceptual Dependency and NKRL, then lowering the number of possible combinations, but totally free in, e.g., Schubert, Sowa or in the linguistic theories.
– The arguments a_k in (1) can be represented, in turn, by complex structures introducing new conceptual operators and new conceptual arguments. If the above example becomes, e.g.: "John gave a book and some flowers to Mary yesterday", a correct representation of the argument introduced by the OBJECT/PATIENT role must in fact i) include an operator in the COORDINATION style, and ii) ways of SPECIFYING 'flowers' (a plural…) as 'some' – see, in the next Sections, the 'binding operators' and the 'expansions' in NKRL.

When we add also the problem of taking into account the 'connectivity phenomena', it is easy to understand why the need for producing an *operational, standardized, theoretical and practical solution* to *n*-ary relationships problems still exists.

3 A Brief Survey of the NKRL Solution

3.1 Generalities

NKRL innovates with respect to the current ontological paradigms by adding to the usual ontologies of concepts an 'ontology of events', i.e., a new sort of hierarchical organization where the nodes correspond to *n*-ary structures called 'templates'. In the NKRL environment, the 'ontology of concepts' is called HClass (hierarchy of classes): HClass is not fundamentally different from one of the ontologies that can be built up by using tools in the Protégé style. The 'ontology of events' – HTemp, hierarchy of templates – is, on the contrary, basically different; the two hierarchies operate in a strictly integrated way in an NKRL context. A partial representation of HClass is given in Fig. 1; a full description is given in [19]. Fig. 2 reproduces the 'symbolic labels' of part of the templates included in the Produce: branch of HTemp.

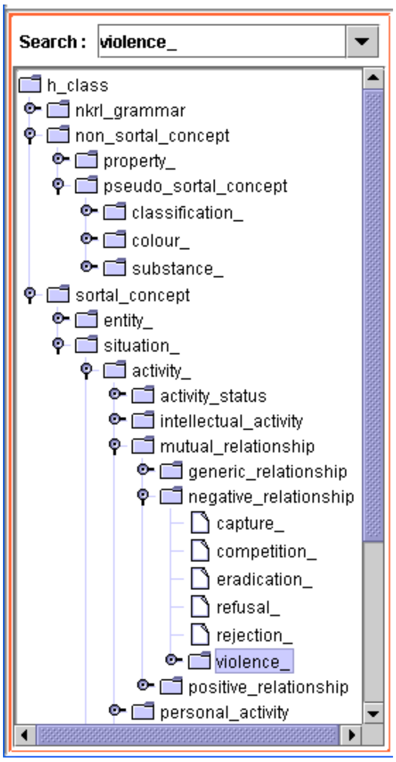

Fig. 1. Partial representation of HClass, the 'traditional' ontology of concepts

Instead of using the traditional *object (class, concept) – attribute – value* organization, templates are generated from the association of *quadruples* that follow the general schema (1) supplied above. Predicates pertain to the set {BEHAVE, EXIST, EXPERIENCE, MOVE, OWN, PRODUCE, RECEIVE}, and roles to the set

EXPERIENCE, MOVE, OWN, PRODUCE, RECEIVE}, and roles to the set {SUBJ(ect), OBJ(ect), SOURCE, BEN(e)F(iciary), MODAL(ity), TOPIC, CONTEXT}; predicates and roles are then ‘primitives’. An argument a_k of the predicate, see (1), denotes indirectly through a ‘variable’ either a simple ‘concept’ or a structured association (‘expansion’) of several concepts. In both cases, the concepts can only be chosen among those included in the HClass hierarchy; this fact, linked with the ‘primitive’ character of predicates and roles, allows to reduce considerably the potential combinatorial explosion associated with formulas like (1).

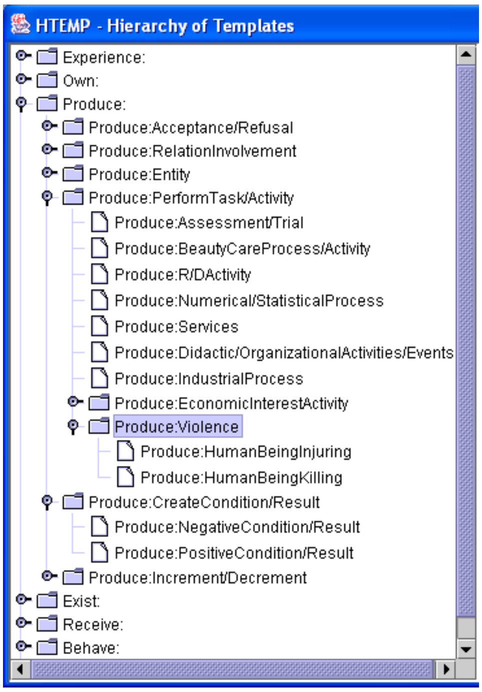

Fig. 2. Partial representation of the PRODUCE branch of HTemp, the ‘ontology of events’

Templates formally represent 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. – for additional details and a full description of HTemp, see again [19]. When a particular 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 a simple narrative – extracted from one of the new stories of the ‘Philippines’ original corpus – like: “On November 20, 1999, in an unspecified village, an armed group of people has kidnapped Robustiniano Hablo”, we must then select firstly in the HTemp hierarchy the template corresponding to ‘execution of violent actions’, see Fig. 2 and Table 1a below. This template is a specialization (see the ‘father’ code in Table 1a) of the particular PRODUCE template corresponding to “perform some task or activity”.

Table 1. Building up and querying predicative occurrences

a)

name: Produce:Violence
father: Produce:PerformTask/Activity
position: 6.35
NL description: 'Execution of Violent Actions on the Filler of the BEN(e)F(iciary) Role'

PRODUCE	SUBJ	*var1*: [(*var2*)]
	OBJ	*var3*
	[SOURCE	*var4*: [(*var5*)]]
	BENF	*var6*: [(*var7*)]
	[MODAL	*var8*]
	[TOPIC	*var9*]
	[CONTEXT	*var10*]
	{[modulators], ≠abs}	

var1 = *<human_being_or_social_body>*
var3 = *<violence_>*
var4 = *<human_being_or_social_body>*
var6 = *<human_being_or_social_body>*
var8 = *<criminality/violence_related_tool>* | *<general_characterising_property>* | *<machine_tool>* | *<small_portable_equipment>* | *<violence_>* | *<weapon_>*
var9 = *<h_class>*
var10 = *<situation_>* | *<spatio/temporal_relationship>* | *<symbolic_label>*
var2, var5, var7 = *<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:	20/11/1999
		date-2:	

Produce:Violence (6.35)

*On November 20, 1999, in an unspecified village (*VILLAGE_1*), an armed group of people has kidnapped Robustiniano Hablo.*

c)

PRODUCE
SUBJ : human_being :
OBJ : violence_
BENF : human_being :
{}
date1 : 1/1/1999
date2 : 31/12/1999

Is there any information in the system concerning violence activities during 1999?

As it appears from Table 1a, the arguments of the predicate (the a_k terms in (1)) are represented by variables with associated constraints. When deriving a predicative occurrence like mod3.c5 in Table 1b, the role fillers in this occurrence must conform to the constraints of the father-template. For example, ROBUSTINIANO_HABLO (the 'BEN(e)F(iciary)' of the action of kidnapping) and INDIVIDUAL_PERSON_20 (the unknown 'SUBJECT', actor, initiator etc. of this action) are both 'individuals', instances of the HClass concept individual_person: this last is a specialization of human_being_or_social_body, see, in Table 1a, the constraint on the variables *var1* and *var6*. The 'attributive operator', SPECIF(ication), of Table 1b, is one of the four operators that make up the AECS sub-language, used for the set up of 'structured arguments' ('expansions'). Apart from SPECIF = S, AECS includes also the disjunctive operator, ALTERN(ative) = A, the distributive operator, ENUM(eration) = E, and the collective operator, COORD(ination) = C. In particular, the (recursive) SPECIF lists, with syntax (SPECIF e_i p_1 … p_n), are used to represent the properties or attributes which can be asserted about the first element e_i, concept or individual, of the list.

Until now, we have evoked the NKRL solutions to the problem of representing *elementary (simple) events*. To deal now with those *'connectivity phenomena'* that arise when several elementary events are connected through causality, goal, coordination and subordination etc. links, NKRL makes use of second order structures created through *reification* of the conceptual labels of the predicative occurrences. A simple example concerns the filler of the CONTEXT role in the occurrence mod3.c5 of Table 1b: in this case ('completive construction'), the 'context' of the kidnapping is supplied by a whole predicative occurrence, mod3.c6, telling us that the kidnapping happened when Robustiniano Hablo was on his way home with his father. More complex examples of second order constructions are the 'binding occurrences', that consist of lists of symbolic labels of predicative occurrences; the lists are differentiated making use of specific binding operators like GOAL and CAUSE, see [1, 2, 3].

3.2 'Search Patterns' and the First Level of Inference Procedures

The *basic building block* for all the NKRL querying and inference procedures is the Fum, Filtering Unification Module, see also [20]. It takes as input specific NKRL data structures called 'search patterns'.

Search patterns are the NKRL counterparts of natural language queries; they offer then the possibility of querying *directly* an NKRL knowledge base of conceptual annotations. Formally, these patterns correspond to *specialized/partially instantiated templates* pertaining to the HTemp hierarchy, where the '*explicit variables*' that characterize the templates (var_i, see Table 1a) *have been replaced by concepts/individuals compatible with the constraints imposed on these variables in the original templates*. In a search pattern, the concepts are used as '*implicit variables*'. When trying to unify a search pattern with the predicative occurrences of the knowledge base, a concept can then *match* the individuals representing its own instances and all its subsumed concepts in HClass with their own instances. The set of predicative occurrences unified by a search pattern constitutes the *answer* to the query represented by the pattern.

A simple example of search pattern, translating the query: “Is there any information in the system about violence events occurred during the year 1999?” is reproduced in Table 1c, producing the occurrence mod3.c5 (Table 1b) as one of the possible answers.

3.3 High Level Inference Procedures, ‘Hypotheses’ and ‘Transformations’

The *high-level inferencing operations* of NKRL correspond mainly to the use of two complementary classes of inference rules, hypotheses and transformations. Execution of both hypotheses and transformations require the use of a real InferenceEngine, having Fum as its core mechanism. Let us then suppose we have directly retrieved, thanks to an appropriate search pattern, the occurrence conc2.c34, see Table 2a, which corresponds to the information: “Pharmacopeia, an USA biotechnology company, has received 64,000,000 USA dollars from the German company Schering in connection with a R&D activity”. We will suppose, moreover, that this occurrence is not *explicitly* related with other occurrences in the base by second order elements. Under these conditions, we can activate the InferenceEngine module of NKRL, asking it to try to *link up automatically* the information found by the search pattern with other information present in the base. If this is possible, this last information will represent a sort of ‘causal explanation’ of the information originally retrieved – i.e., in our example, an ‘explanation’ of the money paid to Pharmacopeia by Schering. A hypothesis rule that could fit our case is hypothesis *h1* reproduced in Table 2b.

From an algorithmic point of view, InferenceEngine works according to a backward chaining approach with chronological backtracking. Four ‘*environment variables*’ are used:

– VALAFF (*valeurs affectables* in French), holds the values provisionally assigned to the variables var_i of the three schemata of Table 2 (*premise*, *cond1* and *cond2*) that implement the reasoning steps of the hypothesis: these values can be deleted after a backtracking operation;
– DESVAR holds the final values associated with the variables var_i when the successful processing of one of the reasoning schemata has been completed;
– RESTRICT holds all the constraints (HClass terms) associated with the variables var_i of the different reasoning schemata: these constraints will be used to build up systematically *all* the search patterns that can be derived from these schemata;
– OCCUR holds the list of the symbolic names of all the occurrences retrieved by the search patterns derived from the reasoning schemata: the values bound to var_i retrieved in these occurrences are used to build up the VALAFF lists.

The first set of operations corresponds to the execution of the Exeprem submodule of InferenceEngine, and consists in trying to unify, *using Fum*, the *premise* of the hypothesis, see Table 2b, and the event (the payment in our case, see conc2.c34) to be ‘explained’ – more exactly, in trying to unify (using Fum) *the event and the different search patterns derived from the premise by systematically substituting to the variables var1 and var2, see Table 1b, the associated constraints*. In our case, the premise variable *var1* can only be substituted by the constraint company_; on the contrary, two substitutions, *var2* = human_being and *var2* = company_ are possible for the variable *var2*. A first search pattern will be then built up by substituting human_being for *var2*, i.e., a first unification with the event to explain will be tried by using a search pattern corresponding to a payment done by an *individual person* instead of a *company*. This unification obviously fails.

Table 2. An example of hypothesis rule

a)

```
conc2.c34)   RECEIVE   SUBJ     (SPECIF PHARMACOPEIA_
                                    (SPECIF biotechnology_company USA_))
                       OBJ      (SPECIF money_ usa_dollar
                                    (SPECIF amount_ 64,000,000))
                       SOURCE   (SPECIF SCHERING_
                                    (SPECIF pharmaceutical_company GERMANY_))
                       TOPIC     r_and_d_activity
                       date1 :
                       date2 :
```

b)

HYPOTHESIS *h1*

<u>premise</u> **:**

```
RECEIVE      SUBJ      var1
             OBJ       money_
             SOURCE    var2
```

var1 = company_
var2 = human_being, company_

A company has received some money from another company or a physical person.

<u>first condition schema (`cond1`)</u> **:**

```
PRODUCE      SUBJ      (COORD var1 var2)
             OBJ       var3
             BENF      (COORD var1 var2)
             TOPIC     (SPECIF process_ var4)
```

var3 = mutual_relationship, business_agreement
var4 = artefact_

The two parties mentioned in the premise have concluded an agreement about the creation of a some sort of 'product'.

<u>second condition schema (`cond2`)</u> **:**

```
PRODUCE      SUBJ       var1
             OBJ        var4
             MODAL      var5
             CONTEXT    var3
```

var5 = industrial_process, technological_process

The company that received the money has actually created the product mentioned in the first condition schema.

The engine then 'backtracks' making use of a second sub-module of InferenceEngine, Reexec. The association *var2* = human_being is removed and, using the constraint values stored in RESTRICT, the engine builds up a new pattern making use now of the value *var2* = company_, that will unify the value SCHERING_ in conc2.c34. The engine can then continue the processing of the hypothesis *h1*; the two values *var1* = PHARMACOPEIA_ and *var2* = SCHERING_ will then be stored in DESVAR and passed to the first condition schema (*cond1*), see Table 2b. The search patterns derived from this condition schema – by taking into account the values already bound in DESVAR to *var1* and *var2* and by replacing systematically, as usual, all the other variables with the associated constraints – will be tested by a third sub-module of InferenceEngine, Execond. This last is called *whenever there exist conditions favourable for advancing in the hypothesis*, in other words, for being able to process a new condition schema. Exeprem and Execond perform then the forward traversal of the choice tree, with Reexec being systematically called whenever the conditions for a backtracking exist. The difference between Exeprem and Execond consists mainly in the fact that, in an Execond context, the unification of the search patterns derived from the condition schemata is tested *against the general knowledge base of predicative occurrences to (try to) find possible unifications with these occurrences* while, in an Exeprem context, the unification concerns only the search patterns derived from the premise and the (unique) starting occurrence.

As usual, many deadlocks are generated in the course of the Execond operations. Without entering into further details – see [3] for additional information – we will, eventually, find in the base an instantiation of *cond1* corresponding to an event of the form: "Pharmacopeia and Schering have signed two agreements concerning the production by Pharmacopeia of a new compound, COMPOUND_1". The values associated with the variables *var3* (r_and_d_agreement and sale_agreement) and *var4* (COMPOUND_1) in *cond1* will then be used to create the search patterns derived from *cond2*. It will then be possible to retrieve an occurrence corresponding to the information: "In the framework of an R&D agreement, Pharmacopeia has actually produced the new compound". The global information retrieved through the execution of the hypothesis can then supply a sort of 'plausible explanation' of Schering's payment: Pharmacopiea and Schering have concluded some agreements for the production of a given compound, and this compound has been actually produced by Pharmacopeia.

The second class of inference rules considered here, the 'transformation rules', are used to obtain a *plausible answer* from a repository of predicative occurrences also in the absence of the explicitly requested information (i.e., when a direct query formulated in Fum terms fails), by searching *semantic affinities* between what is requested and what is really present in the repository. The principle employed consists in using these rules to automatically 'transform' the original query (i.e., the original search pattern) into one or more different queries (search patterns) that *are not strictly 'equivalent' but only 'semantically close' to the original one.*

To pass now to a Parmenides example, suppose we ask: “Search for the existence of some links between ObL (a well known international ‘terrorist’) and Abubakar Abdurajak Janjalani, the leader of the Abu Sayyaf group” – the Abu Sayyaf group is one of the Muslim independence movements 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”, which could lead to retrieve: “During 1998/1999, Abubakar Abdurajak Janjalani has received an undetermined amount of money from ObL through an intermediate agent”.

From a formal point of view, transformation rules are 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 representation(s) of one or more queries that must be substituted for the given one. A transformation rule can, therefore, be expressed as: *A* (antecedent, left-hand side) ⇒ *B* (consequent(s), right-hand side). The ‘transformation arrow’, ‘⇒’, has a double meaning:

– operationally speaking, the arrow indicates the *direction* of the transformation: the left-hand side *A* (the original search pattern) is removed and replaced by the right-hand side *B* (one or more new search patterns);
– the ‘semantic’ meaning of the arrow is that the information obtained through *B* *implies* (in a weak meaning) the information we should have obtained from *A*.

Some formal details can be found in [3]. A representation of the Parmenides ‘economic/financial transfer’ transformation introduced before is given in Table 4.

Table 4. A simple example of ‘transformation’ rule.

‘economic/financial transfer’ transformation :

T1)	BEHAVE	SUBJ	(COORD1 *var1 var2*)	⇒	RECEIVE	SUBJ	*var2*
		OBJ	(COORD1 *var1 var2*)			OBJ	*var4*
		MODAL	*var3*			SOURCE	*var1*

var1 = human_being_or_social_body
var2 = human_being_or_social_body
var3 = business_agreement, mutual_relationship
var4 = economic/financial_entity

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

3.4 Recent Developments

The two main modalities of inferencing of NKRL, ‘hypotheses’ and ‘transformations’, have been ‘integrated’ in a Parmenides context. ‘Integrating’ corresponds to:

– From a practical point of view, transformations can now be used to find some useful answers when the search patterns derived *directly* from a condition schema

of a hypothesis fail: a hypothesis deemed then to fall short can continue successfully until its normal end.
– From a more general point of view, transformations can be used to modify in an *a priori* unpredictable way the reasoning steps (condition schemata) to be executed within a 'hypothesis' context. This is equivalent to 'break' the predefined scenarios proper to the hypothesis rules, and to augment then the possibility of discovering 'implicit information' within the knowledge base.

A complete description on the integration procedures can be found in [21]; a recent, very detailed paper on this topic is [22]. Here, we will limit ourselves to supply an informal example. Let us suppose that, as one of the possible answers to a question concerning kidnapping events, we have retrieved the information: "Lieven de la Marche and Eric Brake have been kidnapped by a group of people on June 13, 1999". Using a hypothesis rule like that of Table 5 to 'explain' the kidnapping will give rise to a failure because of the impossibility of satisfying directly the 'intermediate' steps Cond1, Cond2 and Cond3 of *h2*, i.e., of founding *direct matches* of the search patterns derived from these condition schemata with information in the knowledge base.

Table 5. Inference steps for the *h2* hypothesis

(Cond1)	The kidnappers 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 (other rules deal with civil servants, servicemen, members of the clergy etc.).
(Cond4)	It can be proved that the kidnapped is really a businessperson or assimilated.

If we allow now the use of transformations in a hypothesis context, this means to make use of a hypothesis *h2* having a format *potentially equivalent* to that of Table 6. For example, the proof that the kidnappers are part of a terrorist group or separatist organization can be now obtained *indirectly*, transformation T3, by checking whether they are members of a specific subset of this group or organization. We can see, in particular, that a whole family of transformations corresponds to the condition schemata Cond2 of *h2*. They represent variants of this general scheme: the separatist movement or the terrorist organization, or some group or persons affiliated with them, have requested/received money for the ransom of the kidnapped. Note that transformation T2 implies only one 'consequent' schema, whereas all the residual transformations of the 'family' are 'multi-consequent'.

4 Conclusion

In this paper, we have first shown that the usual ontological tools, both the 'traditional' (frame-based) ones and the new ones proposed in a Semantic Web context, are unable to offer complete and reliable solutions to the problem of a non-trivial representation and exploitation of that economically important and ubiquitous type of multimedia information corresponding to the ‘narratives’. After having recalled the existence of early proposals in this field, we have supplied some details about NKRL (Narrative Knowledge Representation Language), a fully implemented, up-to-date knowledge representation and inferencing system especially created for an ‘intelligent’ exploitation of narrative knowledge. The main innovation of NKRL consists in associating with the traditional ontologies of concepts an ‘ontology of events’, i.e., a hierarchical arrangement where the nodes correspond to *n*-ary structures called ‘templates’. After having briefly discussed the query/answering tools associated with NKRL, the paper ends by mentioning recent work in the inferencing domain.

Table 6. Rule *h2* with transformations concerning the intermediary inference steps

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

References

1. 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
2. Zarri, G.P. (1997). NKRL, a Knowledge Representation Tool for Encoding the ‘Meaning’ of Complex Narrative Texts. Natural Language Engineering. 3, 231-253
3. Zarri, G.P. (2003a). A Conceptual Model for Representing Narratives. In: Innovations in Knowledge Engineering. Adelaide: Advanced Knowledge International

4. Noy, F.N., Fergerson, R.W., and Musen, M.A. (2000). The Knowledge Model of Protégé-2000: Combining Interoperability and Flexibility. In: Proceedings of EKAW'2000. Berlin: Springer-Verlag
5. McGuinness, D.L., van Harmelen, F. (2004). OWL WEB Ontology Language Overview – W3C Recommendation 10 February 2004. W3C (http://www.w3.org/TR/owl-features/)
6. Sowa, J.F. (1999). Knowledge Representation: Logical, Philosophical, and Computational Foundations. Pacific Grove (CA): Brooks Cole Publishing Co.
7. Witbrock, M., Panton, K., Reed, S.L., Schneider, D., Aldag, B., Reimers, M., and Bertolo, S. (2004). Automated OWL Annotation Assisted by a Large Knowledge Base. In: Proceedings of SEMANNOT 2004 – 3rd International Semantic Web Conference. Hiroshima (Japan): W3C
8. Noy, N., and Rector, A., eds. (2005). Defining N-ary Relations on the Semantic Web – W3C Editor's Draft 7 September 2005 2005. W3C (http://smi-web.stanford.edu/people/noy/nAryRelations/n-aryRelations-2nd-WD.html)
9. Ceccato, S. (1967). Correlational Analysis and Mechanical Translation. In: Machine Translation, Booth, A.D., ed. Amsterdam: North-Holland
10. Schank, R.C. (1973). Identification of Conceptualisations Underlying Natural Language. In: Computer Models of Thought and Language. San Francisco: Freeman and Co.
11. Schubert, L.K. (1976). Extending the Expressive Power of Semantic Networks. Artificial Intelligence**.** 7, 163-198
12. Sowa, J.F. (1984). Conceptual Structures: Information Processing in Mind and Machine. Reading (MA): Addison-Wesley
13. Fillmore, C.J. (1966). Toward a Modern Theory of Case. In: Modern Studies of English: Readings in Transformational Grammar. Englewood Cliffs (NJ): Prentice Hall
14. Jackendoff, R. (1990). Semantic Structures. Cambridge (MA): The MIT Press
15. Kamp, H., and Reyle, U. (1993). From Discourse to Logic. Dordrecht: Kluwer
16. Tenny, C.L., and Pustejovsky, J., eds. (2000). Events as Grammatical Objects: The Converging Perspectives of Lexical Semantics, Logical Semantics and Syntax. Stanford: CSLI Publications
17. Mani, I., and Pustejovsky, J. (2004). Temporal Discourse Models for Narrative Structure. In: Proceedings of the ACL Workshop on Discourse Annotation. East Stroudsburg (PA), Association for Computational Linguistics
18. Berardi, D., Calvanese, D., and De Giacomo, G. (2001). Reasoning on UML Class Diagrams Using Description Logic Based Systems. In: Proceedings of ADL-2001 (CEUR vol. 44). Aachen: CEUR (http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-44/)
19. Zarri, G.P. (2003). NKRL Manual, Part II – Description of the Ontologies (Parmenides Technical Report). Paris: University of Paris IV/Sorbonne
20. Zarri, G.P., and Bernard, L. (2004). NKRL Manual, Part III – The NKRL Software (Parmenides IST Report). Paris: University of Paris IV/Sorbonne
21. Zarri, G.P., and Bernard, L. (2004). Using NKRL Inference Techniques To Deal With MoD 'Terrorism' Information (Parmenides IST Report). Paris: University of Paris IV/Sorbonne
22. Zarri, G.P. (2005). Integrating the Two Main Inference Modes of NKRL, Transformations and Hypotheses. Journal on Data Semantics (JoDS) 4, 304-340