Differentiating Between "Functional" and "Semantic" Roles in a HighLevel Conceptual Data Modeling Language

Conference Paper · May 2011

Source: DBLP

1 author:

Gian Piero Zarri
Université Paris-Sorbonne - Paris IV

Differentiating Between "Functional" and "Semantic" Roles in a High-Level Conceptual Data Modeling Language

Gian Piero Zarri

University Paris-Est/UPEC, LiSSi Laboratory
120-122, rue Paul Armangot – 94400 Vitry sur Seine - France
zarri@noos.fr, gian-piero.zarri@u-pec.fr

Abstract

We discuss in this paper, from a pragmatic and operational point of view, the need of a clear differentiation between *functional* and *semantic* "roles". In the first case, according to the linguistic and computational linguistics tradition, roles are seen as *relations* linking a semantic predicate to its arguments. In the second, in conformity with the ontological and Semantic Web practice, roles are equated to ordinary *concepts* to be inserted into a standard ontology. As we will show here, the two notions can successfully co-exist in the framework of a high level conceptual modeling language.

Introduction

According to the common ontological practice, "roles" are dealt with as *binary-structured concepts or classes* that, like the usual concepts, can be inserted into a specific branch of a standard ontology. From a structural and semantic point of view, they are not really different, then, from traditional ontological notions like "human being" or "physical object".

For example, in a general model like the OpenCyc upper level, see http://www.cyc.com/cyc/opencyc, Role is a *standard ontological entity*, specialization of ObjectPredicate. It derives, through intermediary steps like Predicate, ThruthFunction, MathematicalObject, from IntangibleIndividual and, eventually, from the top concept of the CYC ontology, Thing. Looking at the more recent, W3C-focused work – where the ontologies are expressed making use of the so-called W3C languages, RDF(S), OWL, OWL-2 etc. – the way of dealing with the notion of role is not really different. To give only two simple examples, in an ontology like the "general ontology of social roles and interactions" used in the EU-supported CASCOM project (Cáceres et al., 2006), the authors make use of a *unique, structured ontological organization* based on a differentiation between "social" and "communicative" roles. Each one of these concept-roles are then inserted in a binary structure where a Patient is a MedicalAdvisee, specific term of Advisee, HealthStatusInformer is a specific term of Informer, etc. In the "Task-based service navigation ontology" (Fukazawa et al., 2006), the top Role node gives rise to two branches, which stem respectively from the nodes TaskRole and SocialRole. In the first branch, we find then a PassengerRole having as specific terms FlightPassengerRole or TaxiPassengerRole. In the second, MotherRole is a specific term of FamilyRole; etc.

This 'roles as standard (binary) concepts' vision is also reflected in recent ontological developments like the so-called "Ontology Design Patterns (ODPs)". Several ODPs – classed as "Structural ODPs", "Correspondence ODPs", "Content ODPs", "Presentation ODPs" etc. – can be found at http://ontologydesignpatterns.org/wiki/Main_Page. These patterns – that originate from the D&S (Descriptions and Situations) work, see (Gangemi and Mika, 2003), a plug-in extension of the DOLCE "upper ontology" (Gangemi *et al.*, 2002) – consist, in practice, in the composition of *fragments of standard (binary) ontologies*. Mizoguchi and his colleagues (2007: 160) note, with respect to the "…idiosyncratic patterns…", that they can lead "... to a decrease of the semantic interoperability of ontologies because ... such patterns will lack compatibility with others".

Dealing with "roles" exclusively as static binary concepts/classes/entities that can be stuck on a standard ontology is *intuitively disturbing*, given that roles are naturally seen as *functions* and *relationships* having then a general *dynamic* function. Faced with this unsatisfactory situation, a high-level conceptual language like NKRL – the "Narrative Knowledge Representation Language", see Zarri (2005; 2009) – has adopted a sort of 'radical' solution for dealing with the notion of role by differentiating between "*semantic*" and "*functional*" roles.

Semantic roles take into account the *static, classificatory aspects* of this notion. In NKRL, they are then dealt with, classically, as "*concepts*", to be inserted in a sub-hierarchy having semantic_role at its top and being part of a traditional binary ontology – called HClass (hierarchy of classes) in an NKRL context. We can note immediately, see also Fig. 1 and next Section, that the semantic_role sub-hierarchy is a specialization of the non_sortal_concept branch of HClass. This means that all its specific terms, like student_, customer_ or employee_, *cannot be endowed with direct instances*: in NKRL, we can state that, at a specific date, John is a student, but the creation of a possible instance like STUDENT_1 is semantically forbidden.

The representation of the semantic roles follows then the ontological tradition; that of the *functional roles*, on the contrary, will surely sound as quite familiar in a Linguistics/Computational Linguistics context. NKRL functional roles are in fact dealt with – according to an approach similar (at least partly) to that used for Case Grammars in Linguistics and recent projects like FrameNet and VerbNet – *as full-fledged relationships denoted by "primitives" instead of "concepts"*. To represent then a simple situation or event like "John gives a book to Mary" we will make use of the three *functional roles* SUBJ(ect), OBJ(ect) and BEN(e)F(iciary) to introduce, respectively, the *instances* (with respect to HClass) JOHN_, BOOK_1 and MARY_ as the *arguments* of the semantic predicate MOVE. Being primitive, the functional roles constitutes a *closed set* of formal entities in NKRL, whilst the sub-hierarchy of HClass including the semantic roles is *open* and new terms can be added when necessary.

The general context of the separation into semantic and functional roles will be discussed in the following sub-Sections. A short "Conclusion" will close the paper.

The notion of role in an NKRL context

"Plain/static" knowledge

The differentiation between "semantic" and "functional" roles can be fully understood only by differentiating in turn between ontological categories that – in the absence of clear suggestions in the literature – we will denote here as "*plain/static*" and "*structured/dynamic*" knowledge.

Plain/static information/knowledge corresponds to *self-contained, basic and permanent notions (concepts)* that must be considered as absolutely necessary to take correctly into account the general context of a given, practical application. These notions can be very general (like "human being", "amount", "color" or "artifact") – and proper, then, to several application domains – or specially linked to a given application/set of applications (like "control room operator", "level of temperature" or "valve"). Plain/static notions are characterized by the following properties:

- They correspond to a '*stable*' vision of the world or of some of its fragments. These notions can then, at least in principle, be defined and classified *a priori* – in theory, genus/species criteria could be sufficient for this aim – independently from their successive use within a specific application; they are then, in a sense, '*a-temporal*' and '*universal*'.
- When, as usual, *specific, formal definition/descriptions* are added to these 'static' notions, these can then be considered (at least in the short term, e.g., in the context of a given application) *as firm and not subject to change*. Of course, they can evolve in the long term, as a consequence of the progress of our knowledge or because they must be used in a different domain.
- Accordingly, these formal definitions/descriptions can be kept as relatively simple and based on the *traditional ontological model*, where the properties or attributes that define a given notion/concept are then expressed as *binary* (i.e., linking only two arguments) *relationships of the "property/value" type*. And this independently from the fact that these relations are organized, e.g., into frame format, or take the form of a set of "property statements" defining a class in a language like OWL.

NKRL follows then the traditional *binary approach* with respect to the representation of this first category of knowledge, through the use of its proper ontology of (plain/static) concepts – called HClass (hierarchy of classes) as already stated. Making use of a simple frame-like type of representation HClass – which includes presently (February 2011) more than 7,500 "concepts" – is not fundamentally different, then, from one of the ontologies that can be built up by using tools in the original, frame-oriented Protégé style. From a general, ontological point of view, therefore, *HClass is more important because of its autonomous existence in order to take into account a well-defined class of cognitive phenomena than for the originality of the conceptual structures used.*

We will limit ourselves to mention here that the main architectural principle underpinning the HClass' *upper level* concerns the partition between sortal_concept and non_sortal_concept. This corresponds to the differentiation between "*(sortal) notions* that can be instantiated directly into enumerable specimens (individuals)", like chair_ (a physical object) and "*(non-sortal) notions* which cannot be instantiated directly into specimens", like gold_ (a substance), white_ (a color) or student_ (a property corresponding to a "semantic role"). A fragment of HClass is reproduced in Fig. 1, where it appears clearly that the semantic_role sub-hierarchy is a specialization – through animate_entity_property, qualifier_ and property_ – of non_sortal_concept in HClass.

"Structured/dynamic" knowledge

The structured/dynamic knowledge concerns the representation, as autonomous entities, of *temporally ordered and logically/semantically coherent streams of elementary events* – in other terms, it deals with the symbolic representation of those complex information structures denoted for example, according to the context, as "narratives", "eChronicles" or "complex events". Each elementary event included in the stream, *represented in turn as an autonomous entity*, consists of the description of a particular action/state/situation/episode etc., involving a number ≥ 1 of the "plain/static" entities introduced in the previous sub-Section. Note that – according to the evolving nature of the structured/dynamic knowledge and at the difference of the fixed, predictable properties of the plain/static information – both the type of the plain/static entities concerned by an elementary event and the nature of their semantic/temporal relationships cannot be anticipated at the beginning of a given, concrete application.

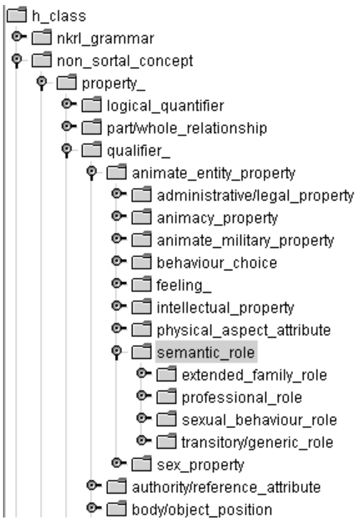

Fig. 1. semantic_role as specialization of non_sortal_concept.

Following a typical "neo-Davidsonian" approach, each elementary event is recognized through the detection, within the natural language formulation of the whole stream/narrative/complex event, of generalized predicates corresponding normally to the usual syntactic/grammatical "verbs" but also to adjectives ("… *worth* several dollars…"), nouns ("…Jane's *amble* along the park…") etc. when they have a predicative function. A (verbalized) example of structured/dynamic entity – of a *stream formed by two elementary events* – can be: "The Control Room operator presses a button to initialize a start-up sequence", where the two elementary events that make up the stream correspond to the two verbs "press" and "initialize". In examples like "Lucy was looking for a taxi" or "Peter lives presently in Paris", the stream is reduced, on the contrary, to the presence of a unique elementary event.

From what expounded until now, it is evident that the formalization of the structured/dynamic knowledge *must* necessarily be based on the formalization of the notion of *elementary event*. In this last context, we can note:

- The necessity of making use of *conceptual predicates* – translations, at a deep conceptual level, of *surface/linguistic level* predicates like "amble", "press", "initialize", "live" etc. met in the previous examples – to specify the basic type of state, action, process etc. described in each elementary event.
- The necessity of utilizing the notion of *functional role* introduced above to denote the logical and semantic function of each of the "plain/static" notions involved in the different elementary events – in "The Control Room operator presses a button …" example, the meaning of this "structured/dynamic" entity is fully specified only by stating that the instance (individual) CONTROL_ROOM-OPERATOR_1 is the SUBJ(ect)/ACTOR of the action of "pressing" and that BUTTON_1 is the corresponding OBJ(ect)/PATIENT.

It is then extremely difficult to utilize the simple binary approach used for the plain/static knowledge to represent *correctly and in an 'economic' way* the structured/dynamic information. In this last case, NKRL makes then use – to represent in the best way each one of the elementary events that make up the global narrative/complex event – of a *structured* n*-ary schema* whose 'core' is denoted by Eq. 1:

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

where:

- L_i is a "*symbolic label*" identifying the elementary event to be represented (e.g., the event corresponding to: "The Control Room operator presses a button").
- P_j is a "*conceptual predicate*", i.e., a deep level generalization of a particular surface predicate, independent then from a specific natural language.
- R_k is a generic "*functional role*", like SUBJ(ect), OBJ(ect), etc., i.e., the formalization of the relationship between the predicate and one of its arguments a_k that explains the specific *function* of a_k in the context of the global meaning of the elementary event.
- a_i is then one a generic "*argument*" of the predicate introduced by a specific functional role (e.g., the individuals CONTROL_ROOM-OPERATOR_1 and BUTTON_1 etc. in the previous example).

We can now introduce what represents, from an ontological point of view, the main characteristic of NKRL. This symbolic language makes use, in fact, of *two structurally dissimilar but strictly integrated ontologies*, the first one represented by the (binary) HClass ontology, introduced in the previous sub-Section, that takes into account the plain/static *knowledge*, and the second by an *n*-ary "*ontology of events*" that deals with the structured/dynamic knowledge. The last ontology is a new sort of hierarchical organization where the nodes are represented by *n*-ary structures, called "*templates*", which follow the conceptual schema represented by Eq. 1. This "ontology of events" is then labelled as HTemp (hierarchy of templates) in NKRL. Templates correspond to the formal representation of *general classes of elementary events* like "move a physical object", "be present in a place", "produce a service", "send/receive a message", "make a change of state happen", etc. More precisely, in the templates, the predicates (P_j in Eq. 1) pertain to the set {BEHAVE, EXIST, EXPERIENCE, MOVE, OWN, PRODUCE, RECEIVE}, and the functional roles (R_k) to the set {SUBJ(ect), OBJ(ect), SOURCE, BEN(e)F(iciary), MODAL(ity), TOPIC, CONTEXT}. An argument a_i of the predicate can consist of a simple "concept" (i.e., of an HClass element corresponding to a plain/static entity) or of a structured association ("expansion") of several concepts/HClass elements. The logical integration of the two ontologies is then assured by the use of HClass elements to fulfil the function of HTemp a_i arguments.

In NKRL, *predicates* and *functional roles* are then limited in number and represented as *primitives*: a discussion on this topic can be found in Zarri (2009: 56-61). On the contrary HClass – that, as we have seen, supply the a_i terms (the arguments of the conceptual predicate) in Eq. 1 – is basically a sort of *controlled hierarchical lexicon* whose low levels must be always updated as soon as a new application in a new domain has to be considered.

Additional details about the functional roles

We have emphasized previously the proximity of NKRL's approach to that proper to a (computationally-exploitable) linguistic theory like "case grammars". In this context, some lists of (functional-like) roles that have an *explicit pragmatic/practical flavor* are described in, e.g., Bruce (1975), Spark Jones and Boguraev (1987), Sowa (2000), etc. In a collective report on "Lexical Semantic Encoding", see http://www.ilc.cnr.it/EAGLES96/EAGLESLE.PDF, the EAGLES researchers supply "…a list of the most popular roles and the properties usually associated with them" that is widely reproduced in the literature as a sort of consensus list about semantic relationships. This list includes 7 items: Agent, Patient, Experiencer, Theme, Location, Source and Goal. A Beneficiary role is added in Palmer et al. (2010: 4).

When comparing the seven NKRL functional roles with the above solutions a first, fundamental principle to be kept in mind is that NKRL functional roles *are strictly relative to an elementary event framework*. This means that their duty consists solely in denoting, in the best possible way, the *functional relationships* of the a_i arguments with respect to the predicate P_j within the context of Eq. 1. This principle allows us to discard all the 'roles' that, in the above solutions, can be associated with notions in the CAUSE (e.g., Force and Reason in Spark Jones/Boguraev) or GOAL (e.g., Goal in Spark Jones/Boguraev and EAGLES/Palmer and, at least partially, Completion, Destination and Result in Sowa) style. These last 'roles' do not concern, in fact, the *internal structure* of an elementary event but, on the contrary, the *mutual relations between two (or more) of these events*. Let us consider, e.g., examples like "The girl died from an accident" and "John went to town in order to buy a shirt", introduced by Spark Jones and Boguraev as illustrations of the use of their Force and Goal 'roles'. For each of them – as for the example "The Control Room operator presses a button to initialize a start-up sequence" of the previous sub-Section – we have to deal in reality with the *logical/temporal relationships of the CAUSE/GOAL type between two (or more) elementary events*, identified by recognizing the presence of surface predicative forms like "die" and "accident" or "go" and "buy" – or "press" and "initialize".

The above 'roles' refer then, in reality, to a set of *surface syntactic constructions like causality, goal, indirect speech, co-ordination, subordination*, etc. denoting, at the *deep level*, those logical/temporal relationships that, in an NKRL context, are collectively gathered under the term "*connectivity phenomena*" see, e.g., Zarri (2009: 7). They represent what, in a stream of elementary events:

- leads to a '*global meaning*' of the stream that goes beyond the simple addition of the 'meanings' conveyed by the single elementary events;
- defines *the influence of the context* where a particular event is used on the meaning of this individual event.

The connectivity phenomena correspond then to a fundamental component of that "*structured/dynamic*" knowledge introduced in the previous sub-Section.

A second principle allows us to avoid the inclusion, in the NKRL functional roles, of all those 'roles' dealing with *temporal and spatial notions*, like Locus and Time in some proposals mentioned in Bruce (1975), After, Before, Location etc. in Spark Jones/Boguraev, Duration, PointInTime in Sowa, Location again in EAGLES/Palmer et al., etc.

In the NKRL model, the (single) "semantic predicate", the seven "roles" and the "(simple or complex) arguments" of Eq. 1 are the three *basic building blocks* strictly necessary to give rise to a 'meaningful' representation of an elementary event or of a class of elementary events (a *template*, see the previous sub-Section). These three blocks cannot, however, receive *separately* an interpretation in terms of elementary events: *a valid interpretation will only arise after their (mandatory) assembling has been carried out*. This implies also that all the residual conceptual elements (locations, temporal information, modalities etc.) to be associated, when necessary, with the representation of an elementary event/template, are dealt with simply, in NKRL, as *"determiners" or "attributes"*. They can then introduce further details about the basic core of the representation of a template/elementary event, but are never strictly necessary for its meaningful interpretation.

For example, templates and their instances – "*predicative occurrences*" in NKRL's terms, i.e., representations of specific elementary events – may be accompanied by "*modulators*" (like "non intentional", "social", "possible") that, as their name suggests, are there to refine or modify the basic interpretation of the template or occurrence. Moreover, as we will see in the next sub-Section, the predicative occurrences are *necessarily associated* with two temporal attributes, date-1 and date-2, linked, in case, with specific values. Other determiners are the *"location" attributes*; the NKRL determiners are described in detail in Zarri (2009: 70-86).

Eventually, an informal description of the seven NKRL's functional roles is given in Table 1.

Examples of "Structured/dynamic" knowledge

When a particular elementary event pertaining to one of the general classes included in HTemp must be represented, *the corresponding template is then instantiated* to produce a "*predicative occurrence*", i.e., the formal, NKRL representation of this event. To encode then an elementary event – which concerns a recent NKRL application in the gas/oil industry domain – like: "On October 16th, 2008, the Control Room operator pushes the SEQ1_BUTTON to initialize the a particular sequence of operations, SEQ1, associated with the start-up of the turbine", we must select firstly in HTemp the template corresponding to 'perform a task or an activity', represented in the upper part of Table 2.

Table 1. NKRL's functional roles.

Role	**Acronym**	**Mnemonic Description**
Subject	SUBJ	The *main protagonist* (the 'agent') of the elementary event, independently from the grammatical/syntactic form of the corresponding expression in natural language, see "Caesar has been stabbed by Brutus". The 'filler' of this role is often, but not necessarily, an animate entity or a group of animate entities.
Object	OBJ	The entity, *animate* (e.g., Caesar, the 'patient' in the previous example) *or not* (e.g., the book that is moved from John to Mary), which is *acted upon* in the context of the elementary event.
Source	SOURCE	The animate entity, *if any*, who is *responsible for* the behavior, situation, state etc. of the SUBJ of the event.
Beneficiary	BENF	The animate entity, ("Mary" in the "book" example), or a group of entities, who constitutes the '*addressee*' (the 'recipient' etc.) of the OBJ mentioned in the event (or, more generally, the addressee of the global behavior of the SUBJ of the elementary event.
Modality	MODAL	The (often inanimate) *entity* (e.g., the "knife") or the *process* (e.g., "stabbing", if the elementary event to be represented was "Brutus killed Caesar by stabbing him") that is *instrumental* in producing the situation described in the event.
Topic	TOPIC	The *theme* ('à propos of…') of the fact(s) or situation(s) that are represented in the elementary event (e.g., "Mary's birthday", in the absence of further details, in the "book" example).
Context	CONTEXT	The *general context* ('in the context of…') of the fact(s) or situation(s) that are represented in the elementary event, e.g., "Roman Senate's fears about Caesar's ambitions", "John's love for Mary", etc. Note that the context is often represented by other events or streams of events, or by "reified events", like US_PRESIDENTIAL_ADDRESS_1 see, e.g., Zarri (2009: 137).

Fig. 2 reproduces a very small fragment of the 'external' organization of HTemp hierarchy. As it appears from this figure, HTemp is structured into *seven branches*, where each one of them includes *only* the templates organized – following the syntax of Eq. 1 – around one of the seven conceptual predicates used in NKRL. HTemp includes presently (February 2011) more than 150 templates.

Table 2. Deriving a predicative occurrence from a template**.**

name: Produce:PerformTask/Activity
father: Produce:
position: 6.3
natural language description: 'Execution of Intellectual or Industrial Procedures, of Economic Interest Activities, etc.'

```
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     =  activity_, process_, temporal_development
var4     =  human_being_or_social_body
var6     =  human_being_or_social_body
var8     =  activity_, artefact_, process_,
            temporal_sequence
var9     =  pseudo_sortal_concept, sortal_concept
var10    =  situation_, symbolic_label
var2, var5, var7  =  location_

virt2.c32) PRODUCE SUBJ    INDIVIDUAL_PERSON_102:
                       (GP1Z_MAIN_CONTROL_ROOM)
                   OBJ     button_pushing
                   TOPIC   SEQ1_BUTTON
                   CONTEXT (SPECIF
                             SEQ1_GREASING_PUMP
                             (SPECIF member_of
                           F17_STARTUP_SEQUENCE))
                   date-1:  2008-10-16-08:26
                   date-2:
```

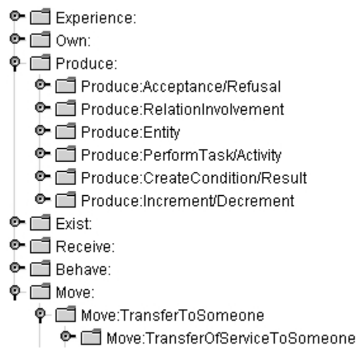

Fig. 2. "PRODUCE" etc. branches of the HTemp hierarchy.

Returning now to Table 2 we see that, in an 'actual' template, the arguments of the predicate (the a_i terms in Eq. 1) are represented *by variables with associated constraints*. The constraints are expressed as *concepts or combinations of concepts*, i.e., using the terms of the 'standard' ontology, HClass, corresponding to the NKRL representation of the *plain/static knowledge*. When creating an occurrence as virt2.c32, the role fillers must conform to the constraints of its father-template. INDIVIDUAL_PERSON_102, e.g., is an instance of the concept individual_person, specialization in turn of human_being_or_social_body, see the constraint on the *var1*. The meaning of the expression "(SPECIF SEQ1_GREASING_PUMP…" in virt2.c32 is: the general framework (role CONTEXT) of the action of pushing the button is a particular process_phase (i.e, SEQ1_GREASING_PUMP) that is part (member_of) of the specific industrial_temporal_sequence represented by F17_STARTUP_SEQUENCE. The "attributive operator", SPECIF(ication), is one of the *four operators* used for the set up of *structured arguments (expansions)*, see Zarri (2009: 68-70). In the occurrences, the two operators date-1, date-2 materialize the temporal interval normally associated with elementary narrative events. A description of the NKRL methodology for representing and managing temporal data can be found, e.g., in Zarri (2009).

What expounded until now illustrates the NKRL solutions to the problem of providing a coherent and complete representation of elementary events. To deal with those "*connectivity phenomena*" introduced previously, the basic NKRL knowledge representation tools have been complemented by *second order structures created through reification of the predicative occurrences' conceptual labels*, see Zarri (2009: 86-98). For example, several predicative occurrences, denoted by their symbolic labels L_i (see Eq. 1) can be associated within the scope of second order structures called "*binding occurrences*", i.e., labeled lists formed of a "*binding operator Bn" with its arguments*. The *Bn* operators include those used in NKRL to represent the "*taxonomy of causality*", see Zarri (2009: 97-101), i.e. CAUSE, REFER(ence) – the "weak causality operator", introducing two arguments where the second is necessary but not sufficient to explain the first – GOAL, MOTIV(ation) – the "weak intentionality operator", where the first argument is not necessary to carry out the second, which is however sufficient to explain the first. The general expression of a binding occurrence is then:

$$(Bn_k \ \ \mathrm{arg}_1 \ \mathrm{arg}_2 \ \ldots \ \mathrm{arg}_n) \ , \qquad (2)$$

Note that the arguments arg_i of Eq. 2 can correspond *directly* to L_i labels – i.e., they can denote simply the temporally ordered presence of particular elementary events – or *correspond recursively to new Bn_i lists in Eq. 2 format*, as in the case a given sequence of elementary events is the CAUSE of another sequence of events.

Returning then to Table 2's example, let us suppose we would now state that: "… the production activities leader pushes the SEQ1_BUTTON … *in order to* start the auxiliary lubrication pump", where the specific elementary event corresponding to the action of pushing is still represented by virt2.c32 in Table 2. To encode correctly the new information, we must introduce first an additional predicative occurrence labeled, e.g., as virt2.c33 and meaning that: "[the aim of the previous action is to …] move AUXILIARY_LUBRICATION_PUMP_M202 from an idle_ to a running_ state". We will eventually add *a binding occurrence* virt2.c30 having the form: "virt2.c30) (GOAL virt2.c32 virt2.c33)"; this last corresponds then to Eq. 2 syntax and can be used to link together the conceptual labels virt2.c32 (the planning activity) and virt2.c33 (the intended result). The global meaning of virt2.c30 can then be paraphrased as: "the activity described in virt2.c32 is focalized towards (GOAL) the realization of virt2.c33".

Conclusion

NKRL is not only a representation language, but also a wholly implemented computer science environment, see Zarri (2005) about the querying/inference procedures.

In this paper, we have discussed, *from a strict pragmatic and operational point of view*, an important feature of NKRL (Narrative Knowledge Representation Language) that can be of general interest from an ontological point of view. This concerns the differentiation between *functional* and *semantic* roles. In the first case, roles are *primitive* symbols, interpreted as "relations" – like "subject/agent", "object", "source", "beneficiary" etc. – linking a semantic predicate to its arguments within a conceptual structure of the *n*-ary type. In the second, they are equated, as usual in an ontological context, to *standard concepts* like "student".

References

Bruce, B. 1975. Case Systems for Natural Language. *Artificial Intelligence* **6**, 327-360.

Cáceres, C., Fernández, A., Ossowski, S., Vasirani, M. 2006. Agent-Based Service Discovery for Healthcare: An Organizational Approach. *IEEE Intelligent Systems* **21**(6), 11-20.

Fukazawa, Y., Naganuma, T., Fujii, K, Kurakake, S. 2006. Construction and Use of Role-ontology for Task-based Service Navigation System. In: *The Semantic Web – ISWC 2006*. Berlin: Springer.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L. 2002. Sweetening Ontologies with DOLCE. In*: Knowledge Engineering and Knowledge Management – Proceedings of EKAW'2002*. Berlin: Springer.

Gangemi, A., Mika, P. 2003. Understanding the Semantic Web through Descriptions and Situations. In: *Proceedings of the Confederated International Conferences*. Berlin: Springer.

Mizoguchi R., Sunagawa E., Kozaki K., Kitamura Y. 2007. *A Model of Roles within an Ontology Development Tool: Hozo. Journal of Applied Ontology* **2**, 159-179.

Palmer, M., Gildea, G., Xue, N. 2010. *Semantic Role Labeling*. San Rafael: Morgan and Claypool Publishers.

Sowa, J.F. 2000. *Knowledge Representation*. Pacific Grove (CA): Brooks/Cole.

Spärck Jones, K., Boguraev, B. 1987. A Note on a Study of Cases. *Computational Linguistics* **13**, 65-68.

Zarri, G.P.. 2005. 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.