NKRL, a Knowledge Representation Language for Narrative Natural Language Processing

Gian Piero Zarri
Centre National de la Recherche Scientifique
CNRS - CAMS
54, boulevard Raspail
75270 PARIS Cedex 06, France
zarri@cams.msh-paris.fr

Abstract

NKRL is a conceptual language which intends to provide a normalised, pragmatic description of the semantic contents (in short, the "meaning") of NL narrative documents. We introduce firstly the general architecture of NKRL, and we give some examples of its characteristic features. We supply, afterward, some sketchy information about the inference techniques and the NLP procedures associated with this language.

1. Introduction

NKRL (Narrative Knowledge Representation Language) aims to propose some possible, pragmatic solutions for the set up of a standardised description of the semantic contents (in short, the "meaning") of natural language (NL) narrative documents. With the term "narrative documents" we denote here NL texts of an industrial and economic interest corresponding, e.g., to news stories, corporate documents, normative texts, intelligence messages, etc.

The NKRL code can be used according to two main modalities. It can be employed as a standard vehicle for the interchange of content information about narrative documents. It can also be utilised to support a wide range of industrial applications, like populating large knowledge bases which can support, thereafter, all sort of "intelligent" applications (advanced expert systems, case-based reasoning, intelligent information retrieval, etc.). NKRL is a fully implemented language ; the most recent versions have been realised in the framework of two European projects : NOMOS, Esprit P5330, and COBALT, LRE P61011.

2. The architecture of NKRL

NKRL is a two layer language.

The lower layer consists of a set of general tools which are structured into several integrated components, four in our case.

The descriptive component concerns the tools used to produce the formal representations (called predicative templates) of general classes of narrative events, like "moving a generic object", "formulate a need", "be present somewhere". Predicative templates are characterised by a threefold format, where the central piece is a semantic predicate (a primitive, like BEHAVE, EXPERIENCE, MOVE, PRODUCE etc.) whose arguments (role fillers) are introduced by roles as SUBJ(ect), OBJ(ect), SOURCE, DEST(ination), etc. ; the data structures proper to the descriptive component are then similar to the case-grammar structures. Templates are structured into a hierarchy, H_TEMP(lates), corresponding, therefore, to a "taxonomy of events".

Templates' instances (predicative occurrences), i.e., the NKRL representation of single, specific events like "Tomorrow, I will move the wardrobe", "Lucy was looking for a taxi", "Peter lives in Paris", are in the domain of the factual component.

The definitional component supplies the NKRL representations, called concepts, of all the general notions, like *physical_entity*, *human_being*, *taxi_*, *city_*, etc., which can play the role of arguments within the data structures of the two components above. The concepts correspond to sets or collections, organised according to a generalisation/specialisation (tangled) hierarchy which, for historical reasons, is called H_CLASS(es). The data structures used for the concepts are, substantially, frame-like structures ; H_CLASS corresponds relatively well, therefore, to the usual ontologies of terms.

The enumerative component of NKRL concerns the formal representation of the instances (concrete, countable examples, see lucy_, wardrobe_1, taxi_53) of the concepts of H_CLASS ; their formal representations take the name of individuals. Throughout this paper, we will use the italic type style to represent a "*concept_*", the roman style to represent an "individual_".

The upper layer of NKRL consists of two parts.

The first is a "catalogue", giving a complete description of the formal characteristics and the modalities of use of the well-formed, "basic templates" (like "moving a generic object" mentioned above) associated with the language — presently, about 150, pertaining mainly to a (very general) socio-economico-political context where the main characters are human beings or social bodies. By means of proper specialisation operations it is then possible to obtain, from the basic templates, the (specific) "derived" templates that could be concretely needed to implement a particular, practical application — e.g., "move an industrial process" — and the corresponding occurrences. In NKRL, the set of legal, basic templates can be considered, at least in a first approach, as fixed.

Analogously, the general concepts which pertain to the upper levels of H_CLASS — such as *human_being*, *physical_entity*, *modality_*, etc. — form a sort of upper-level, invariable ontology.

3. Some characteristic NKRL features

Fig. 1 supplies a simple example of NKRL code. It translates a small fragment of COBALT news : "Milan, October 15, 1993. The financial daily Il Sole

24 Ore reported Mediobanca had called a special board meeting concerning plans for capital increase".

```
c1)   MOVE    SUBJ     (SPECIF sole_24_ore
                          financial_daily): (milan_)
              OBJ      #c2
              date-1:  15_october_93
              date-2:

c2) PRODUCE  SUBJ     mediobanca_
              OBJ      (SPECIF summoning_1
                         (SPECIF board_meeting_1
                          mediobanca_ special_))
              TOPIC    (SPECIF plan_1 (SPECIF
                         cardinality_ several_)
                          capital_increase_1)
              date-1:  circa_15_october_93
              date-2:
```

Figure 1. An NKRL coding.

In Fig. 1, c1 and c2 are symbolic labels of occurrences ; MOVE and PRODUCE are predicates ; SUBJ, OBJ, TOPIC ("à propos of...") are roles. With respect now to the arguments, sole_24_ore, milan_, mediobanca_ (an Italian merchant bank), summoning_1, etc. are individuals ; *financial_daily*, *special_*, *cardinality_* and *several_* (this last belonging, like *some_*, *all_* etc., to the *logical_quantifier* intensional sub-tree of H_CLASS) are concepts. The attributive operator, SPECIF(ication), with syntax (SPECIF e_1 p_1 ... p_n), is used to represent some of the properties which can be asserted about the first element e_1, concept or individual, of a SPECIF list ; *several_* is used within a SPECIF list having *cardinality_* as first element as a standard way of representing the plural number mark, see c2.

The arguments, and the templates/occurrences as a whole, may be characterised by the presence of particular codes, the determiners. For example, the location determiners, represented as lists, are associated with the arguments (role fillers) by using the colon, ":", operator, see c1. For the determiners date-1 and date-2, see (Zarri, 1992a).

A MOVE construction like that of occurrence c1 (completive construction) is necessarily used to translate any event concerning the transmission of an information ("... Il Sole 24 Ore reported ..."). Accordingly, the filler of the OBJ(ect) slot in the occurrences (here, c1) which instantiates the MOVE transmission template is always a symbolic label (c2) which refers to another predicative occurrence, i.e., that bearing the informational content to be spread out ("... Mediobanca had called a meeting ..."). We can note that the enunciative situation can be both explicit or implicit. For example, the completive construction can be used to deal with a problem originally raised by Nazarenko (1992) in a conceptual graphs context, namely, that of the correct rendering of causal situations where the general framework of the antecedent consists of an (implicit) speech situation. Let us examine briefly one of the Nazarenko's examples (1992 : 881) : "Peter has a fever since he is flushed". As Nazarenko remarks, "being flushed" is not the "cause" of "having a fever", but that of an implicit enunciative situation where we claim (affirm, assert etc.) that someone has a fever. Using the completive construction, this example is easily translated in NKRL using the four occurrences of Fig. 2.

```
c3) MOVE   SUBJ  human_being_or_social_body
           OBJ   #c4

c4) EXPERIENCE  SUBJ   peter_
                OBJ    fevered_state_1

c5) EXPERIENCE  SUBJ   peter_
                OBJ    flushing_state_1
                [obs]

c6) (CAUSE c3 c5)
```

Figure 2. An implicit enunciative situation.

We can remark that, in Fig. 2, c6 is a binding occurrence. Binding structures — i.e., lists where the elements are conceptual labels, c3 and c5 in Fig. 2 — are second-order structures used to represent the logico-semantic links which can exist between predicative templates or occurrences. The binding occurrence c6 — meaning that c3, the main event, has been caused by c5 — is labelled using one (CAUSE) of the four operators which define together the taxonomy of causality of NKRL, see (Zarri, 1992b). The presence in c5 of a specific determiner — a temporal modulator, "obs(erve)", see again (Zarri, 1992a) — leads to an interpretation of this occurrence as the description of a situation that, that very moment, is observed to exist.

We give now, Fig. 3, a (slightly simplified) NKRL representation of the narrative sentence : "We have to make orange juice" which, according to Hwang and Schubert (1993 : 1298), exemplifies several interesting semantic phenomena.

```
c7)  BEHAVE   SUBJ  (COORD informant_1
                      (SPECIF human_being
                      (SPECIF cardinality_
                       several_)))
                    [oblig, ment]
                    date1:   observed date
                    date2:

c8)  *PRODUCE  SUBJ   (COORD informant_1
                        (SPECIF human_being
                        (SPECIF cardinality_
                         several_)))
               OBJ    (SPECIF orange_juice
                        (SPECIF amount_ ()))
               date1: observed date + i
               date2:

c9)  (GOAL c7 c8)
```

Figure 3. Wishes and intentions.

Fig. 3 illustrates the standard NKRL way of representing the "wishes, desires, intention" domain. To translate the idea of "acting in order to obtain a given result", we use :

i) An occurrence (here c7), instance of a basic template pertaining to the BEHAVE branch of the H_TEMP hierarchy, and corresponding to the general meaning of focusing on a result. This occurrence is used to express the "acting" component — i.e., it identifies the SUBJ(ect) of the action, the temporal co-ordinates, etc.

ii) A second predicative occurrence, here c8, an instance of a template structured around a different predicate (e.g., PRODUCE in Fig. 3) and which is used to express the "intended result" component.

iii) A binding occurrence, c9, which links together the previous predicative occurrences and which is labelled by means of GOAL, another operator included in the taxonomy of causality of NKRL.

Please note that "oblig" and "ment" in Fig. 3 are, like "obs" in Fig. 2, "modulators", see (Zarri, 1992b), i.e., particular determiners used to refine or modify the primary interpretation of a template or occurrence as given by the basic "predicate — roles — argument" association. "ment(al)" pertains to the modality modulators. "oblig(atory)" suggests that "someone is obliged to do or to endure something, e.g., by authority", and pertains to the deontic modulators series. Other modulators are the temporal modulators, "begin", "end", "obs(erve)", see also Fig. 2. Modulators work as global operators which take as their argument the whole (predicative) template or occurrence. When a list of modulators is present, as in the occurrence c7 of Fig. 3, they apply successively to the template/occurrence in a polish notation style to avoid any possibility of scope ambiguity. In the standard constructions for expressing wishes, desires and intentions, the absence of the "ment(al)" modulator in the BEHAVE occurrence means that the SUBJ(ect) of BEHAVE takes some concrete initiative (acts explicitly) in order to fulfil the result ; if "ment" is present, as in Fig. 3, no concrete action is undertaken, and the "result" reflects only the wishes and desires of the SUBJ(ect).

4. Inferences and NL processing

Each of the four components of NKRL is characterised by the association with a class of basic inference procedures. For example, the key inference mechanism for the factual component is the Filtering and Unification Module (FUM). The primary data structures handled by FUM are the "search patterns" that represent the general properties of an information to be searched for, by filtering or unification, within a knowledge base of occurrences. The most interesting component of the FUM module is represented by the matching algorithm which unifies the complex structures — like "(SPECIF summoning_1 (SPECIF board_meeting_1 mediobanca_ *special_*))" in occurrence c2 of Fig. 1 — that, in the NKRL terminology, are called "structured arguments". Structured arguments are built up in a principled way by making use of a specialised sub-language which includes four expansion operators, the "disjunctive operator", the "distributive operator", the "collective operator", and the "attributive operator" (SPECIFication), see (Zarri, 1996) for more details.

The basic inference mechanisms can then be used as building blocks for implementing all sort of high level inference procedures. An example is given by the "transformation rules", see (Ogonowski, 1987). NKRL's transformations deal with the problem of obtaining a plausible answer from a database of factual occurrences also in the absence of the explicitly requested information, by searching semantic affinities between what is requested and what is really present in the base. The fundamental principle employed is then to "transform" the original query into one or more different queries which — unlike "transformed" queries in a database context — are not strictly "equivalent" but only "semantically close" to the original one.

With respect now to the NL/NKRL translation procedures, they are based on the well-known principle of locating, within the original texts, the syntactic and semantic indexes which can evoke the conceptual structures used to represent these texts. Our contribution has consisted in the set up of a rigorous algorithmic procedure, centred around the two following conceptual tools :

- The use of rules — evoked by particular lexical items in the text examined and stored in proper conceptual dictionaries — which take the form of generalised production rules. The left hand side (antecedent part) is always a syntactic condition, expressed as a tree-like structure, which must be unified with the results of the general parse tree produced by the syntactic specialist of the translation system. If the unification succeeds, the right hand sides (consequent parts) are used, e.g., to generate well-formed templates ("triggering rules").
- The use, within the rules, of clever mechanisms to deal with the variables. For example, in the specific, "triggering" family of NKRL rules, the antecedent variables (*a*-variables) are first declared in the syntactic (antecedent) part of the rules, and then "echoed" in the consequent parts, where they appear under the form of arguments and constraints associated with the roles of the activated templates. Their function is that of "capturing" — during the match between the antecedents and the results of the syntactic specialist — NL or H_CLASS terms to be then used as specialisation terms for filling up the activated templates and building the final NKRL structures.

A detailed description of these tools can be found, e.g., in (Zarri, 1995) ; see also Azzam (1995). Their generality and their precise formal semantics make it possible, e.g., the quickly production of useful sets of new rules by simply duplicating and editing the existing ones.

We reproduce now, Fig. 5, one of the several triggering rules to which the lexical entry "call" — pertaining to the NL fragment examined at the beginning of Section 3. — contains a pointer, i.e., one of the rules corresponding to the meaning "to issue a call to convene". This rule allows the activation of a basic template (PRODUCE4.12) giving rise, at a later stage, to the occurrence c2 of Fig. 1 ; the *x* symbols in Fig. 5 correspond to *a*-variables.

We can remark that all the details of the full template are not actually stored in the consequent, given that the H_TEMP hierarchy is part of the "common shared data structures" used by the translator. Only the parameters relating to the specific triggering rule are, therefore, really stored. For example, in Fig. 5, the list "constr" specialises the constraints on some of the variables, while others — e.g., the constraints on the variables *x1* (*human_being/social_body*) and *x4* (*planning_activity*) — are unchanged with respect to the constraints permanently associated with the variables of template PRODUCE4.12.

trigger: "call"
syntactic condition:

```
(s (subj (np (noun x1)))
  (vcl (voice active) (t = x2 = call))
  (dir-obj
   (np (modifiers (adjs x31))
     (noun x3)
     (modifiers (pp (prep about | concerning | ... )
       (np (noun x4)
         (modifiers (pp (prep of  | for ...)
           (np (noun x5))))))))))
```

parameters for the template :

```
(PRODUCE4.12 (roles subj x1 obj (SPECIF x2
        (SPECIF x3 x31)) +topic (specif x4 x5))
        (constr x3 assembly_ x31 quality_ x5
        modification_procedures))
```

Figure 5. An example of triggering rule.

The "standard" prototype of an NL/NKRL translation system — e.g., the COMMON LISP translator realised in the NOMOS project — is a relatively fast system which take 3 min 16s on Sun SparcStation 1 with 16Mb to process a medium-size text of 4 sentences and 150 wordforms ; it takes 1 min 06s for the longest sentence. This pure conceptual parser, however, is not suitable, *per se*, for dealing directly with huge quantities of unrestricted data. In the COBALT project, we have then used a commercial product, TCS (Text Categorisation System, by Carnegie Group) to pre-select from a corpus of Reuters news stories those concerning in principle the chosen domain (financial news about merging, acquisitions, capital increases etc.). The candidate news items (about 200) have then been translated into NKRL format, and examined through a query system in order to i) confirm their relevance ; ii) extract their main content elements (actors, circumstances, locations, dates, amounts of shares or money, etc.). Of the candidate news stories, 80% have been (at least partly) successfully translated ; "at least partly" means that, sometimes, the translation was incomplete due, e.g., to the difficulty of instantiating correctly some binding structures. Other quantitative information about the COBALT results can be found in (Azzam, 1995 ; Zarri, 1995).

5. Conclusion

Possible, general advantages of NKRL with respect to other formalisms that also claim to be able to represent extensive chunks of semantics, see, e.g., (Lehmann, 1992), are at least the following :

- The addition of a "taxonomy of events" to the traditional "taxonomy of concepts" : often, "normal" ontologies elude in fact the problem of representing how the concepts interact with each other in the context of real-life events. Recently, Park (Park, 1995) has presented a language which provides a set of ontological primitives to be used to model the dynamic aspects ("events") of a domain. However, Park's system seems to be a very "young" one, and it lacks of tools for describing essential narrative features like the relationships between events, the temporal information, etc.
- The presence of a catalogue of standard, basic templates, which can be considered as part and parcel of the definition of the language. This implies 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 and the sharing of previous results.

References

Azzam, S. (1995). "Anaphors, PPs and Disambiguation Process for Conceptual Analysis". In *Proceedings of the 14th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann, San Mateo (CA).

Hwang, C.H., and Schubert, L.K. (1993). "Meeting the Interlocking Needs of LF-Computation, Deindexing and Inference: An Organic Approach to General NLU". In *Proceedings of the 13th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann, San Mateo (CA).

Nazarenko-Perrin, A. (1992). "Causal Ambiguity in Natural Language: Conceptual Representation of 'parce que/because' and 'puisque/since'". In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 92),* Nantes, France.

Lehmann, F., editor (1992). *Semantic Networks in Artificial Intelligence.* Pergamon Press, Oxford.

Ogonowski, A. (1987). "MENTAT : An Intelligent and Cooperative Natural Language DB Interface". In *Proceedings of the 7th Avignon International Conference on Expert Systems and Their Applications (Avignon '87),* vol. 2. EC2 & Cie., Paris.

Park, B.J. (1995). "A Language for Ontologies Based on Objects and Events". In *Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing.* Department of Computer Science of the University of Ottawa.

Zarri, G.P. (1992a). "Encoding the Temporal Characteristics of the Natural Language Descriptions of (Legal) Situations". In A. Martino, editor, *Expert Systems in Law.* Elsevier Science, Amsterdam.

Zarri, G.P. (1992b). "The 'Descriptive' Component of a Hybrid Knowledge Representation Language". In F. Lehmann, editor, *Semantic Networks in Artificial Intelligence.* Pergamon Press, Oxford.

Zarri, G.P. (1995). "Knowledge Acquisition from Complex Narrative Texts Using the NKRL Technology". In B.R. Gaines and M. Musen, editors, *Proceedings of the 9th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop,* vol. 1. Department of Computer Science of the University of Calgary.

Zarri, G.P., and Gilardoni, L. (1996). "Structuring and Retrieval of the Complex Predicate Arguments Proper to the NKRL Conceptual Language". In *Proceedings of the Ninth International Symposium on Methodologies for Intelligent Systems (ISMIS'96).* Springer-Verlag, Berlin.