Natural Language Engineering **3** (2/3): 231–253. Printed in the United Kingdom

© 1997 Cambridge University Press

NKRL, a knowledge representation tool for encoding the 'meaning' of complex narrative texts

GIAN PIERO ZARRI

Centre National de la Recherche Scientifique
EHESS-CAMS, 54 boulevard Raspail, 75270 PARIS Cedex 06, France
e-mail: zarri@cams.msh-paris.fr

(*Received 20 March 1995; Revised 17 May 1995*)

Abstract

In this paper, we describe NKRL (Narrative Knowledge Representation Language), a language designed for representing, in a standardized way, the semantic content (the 'meaning') of complex narrative texts. After having introduced informally the four 'components' (specialized sub-languages) of NKRL, we will describe (some of) the data structures proper to each of them, trying to show that the NKRL coding retains the main informational elements of the original narrative expressions. We will then focus on an important subset of NKRL, the so-called AECS sub-language, showing in particular that the operators of this sub-language can be used to represent some sorts of 'plural' expressions.

1 Introduction

The aim of NKRL (Narrative Knowledge Representation Language) is that of proposing effective solutions for the set up of a *standard, language-independent description* of the semantic content (the 'meaning') of non-trivial narrative documents. With the terms 'narrative documents', we denote here non-preedited, Natural Language (NL) texts of an industrial and economic interest, corresponding, for example, to news stories, telex reports, corporate documents (memos, policy statements, reports and minutes), normative texts, intelligence messages, etc. With the terms 'description of the semantic content', we denote the construction of a normalized picture, suited for machine reasoning, of the network of factual and logical relationships underlying a piece of narrative data.

NKRL is a fully implemented language. The most recent versions have been realized in the context of two European projects: NOMOS, Esprit P5330, and COBALT, LRE. See Zarri (1994) for some information about the particularities of the implementation(s).

NKRL follows a sort of 'interlingua' (i.e. language independent) approach. Its philosophy is, then, radically different from that of some recent proposals (e.g. see Iwanska (1993) and Yokoi (1993)), advocating the direct utilization of natural

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

language as the best possible knowledge representation language ever created in the narrative documents field. Moreover, NKRL shows some peculiarities with respect to the choice of the inspiring principles for the realization of the intermediate, formal language. A popular position in the community of the researchers interested in knowledge representation and reasoning for NL processing and understanding consists, in fact, in arguing that "... if the domain of representation is natural language, then the (minimally, syntactic) form of the natural language should strongly influence the representation" (Ali 1995). This position is largely influenced by the Jackendoff's thesis, according to which the 'meaning' problem cannot be dealt with properly without also examining in depth the 'correspondence' problem. According to Jackendoff's terminology, the 'problem of meaning' concerns the development of "a theory of conceptual structures that is sufficiently expressive for the semantic distinctions among sentences and for the inferences that sentences support", whereas the 'problem of correspondence' concerns "a theory of the relation between conceptual structures and syntax" (Jackendoff 1990). In NKRL's approach, on the contrary, only the problem of meaning is addressed, by developing a formal notation for expressing conceptual contents that is independent from the search for an optimal form of correspondence (e.g. see Jackendoff's 'θ-Criterion') with the surface form these contents can assume in a particular NL. Obviously, in our approach, the correspondence problem still exists, but it is tackled *a posteriori* and in a very pragmatic way, e.g. by finding computerized tools to create NKRL-coded information from the NL sources: we can note, in this respect, that the NKRL code can now be produced in a largely automated way (Zarri 1995*a*).

The paper is organized as follows. In section 2, we introduce the four components of NKRL: a more detailed analysis of some specific NKRL data structures will be carried out in section 3. In section 4, we supply information about an important subset of NKRL, the AECS sub-language, and its role in the representation of 'plural' expressions. In section 5, we outline the arguments that, in our opinion, express the main advantages of our approach.

2 General features of NKRL

2.1 The lower layer

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

In any given domain, there are some important notions that we would like to represent: moreover, the most general among them, like 'human being' or 'physical entity', are common to a majority of domains. The definitional component of NKRL supplies the tools for representing these notions that, by convention, are then called 'concepts'; in NKRL, a concept is, therefore, a definitional data structure associated with a symbolic label like *physical_entity*, *human_being*, *taxi_* (the general class including all the taxis, not a specific cab), *city_*, etc. These definitional data structures are, substantially, frame-like structures (see section 3.2). Moreover, all the

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

NKRL concepts are inserted into a generalization/specialization (tangled) hierarchy that, for historical reasons, is called H_CLASS(es), and which corresponds well to the usual ontologies of terms.

From a more speculative point of view, we must note that, to give rise to well-formed ontologies of NKRL concepts, the original notions must be conceived in terms of 'sets' and 'subsets' – where, as usual, asserting that *a* pertains to the set *A*, (*a* : *A*) is equivalent to assert that *a* is characterized by the type *A*. No confusion is allowed between subsets – giving rise to standard H_CLASS concepts, like *european_city* which is a specialization of the concept *city_* – and instances, like paris_; instances are still associated with H_CLASS as terminal symbols, but they are in the domain of another NKRL component, the enumerative component (see below). Moreover, a fundamental assumption about the organization of H_CLASS concerns the differentiation between the "notions which can be directly realized (instantiated) into enumerable specimens", like 'chair' (a 'physical object') and "notions which cannot be instantiated into specimens", like 'gold' (a 'substance') – please note that a notion like 'white gold' is a specialization of gold, not an instance. The two high-level branches of H_CLASS stem, therefore, from two concepts that – adopting the terminology used, for instance, by Guarino, Carrara and Giaretta (1994) – we can label as *sortal_concepts* and *non_sortal_concepts*. The specializations (subsets) of the former, like *chair_*, *city_* or *european_city*, can have direct (immediate) instances (chair_27, paris_), whereas the specializations of the latter, like *gold_*, or *colour_*, can admit further specializations (subsets), see *white_gold* or *red_*, but do not have direct instances (see, again, section 3.2).

The enumerative component concerns the instances (concrete examples, see lucy_, wardrobe_1, taxi_53, paris_, not subsets) of the sortal concepts; their formal representations take the name of 'individuals'. Individuals are characterized by the fact of being countable, of being associated with a temporal dimension (and, often, with a spatial dimension), and of possessing unique symbolic labels (lucy_, wardrobe_1, taxi_53). Throughout this paper, we will use the slanted type to represent a *concept_*, the sans sérif type to represent an individual_. As already stated, individuals are terminal symbols which cannot give rise themselves to further instances (e.g. viewpoints: Paris of the tourists, Paris as a railway node, Paris in the *Belle Epoque*). If need be, viewpoints are realized according to a solution which can be referred to the seminal paper by Minsky (1975) about frames. This consists in introducing specialized concepts like *tourist_city*, *railway_node*, *historical_city* that all admit the individual paris_ as an instance; paris_ inherits then from each of them particular, 'bundled' sets of slots, e.g. UndergroundStations, TaxisBaseFare, EconomyHotels, TypesOfMerchandise, DailyCommutersRate, etc.

We can note immediately that in NKRL, the concepts (definitional component) and their instances (individuals, enumerative component) are kept conceptually distinct, even if they are represented using the same data structures. The main reason for adopting this type of organization is linked with the very different epistemological status of concepts – which define a generic, abstract mould for some useful notions to be taken into consideration in a given domain – with respect to the individuals, which represent specific entities occurring in the context of some

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

concrete events of the domain. In this respect, concepts are permanent, at least in the context of a given application; individuals (occurrences) represent, on the contrary, unpredictable, randomly-occurring entities stored sequentially into an NKRL-based system according to their order of arrival in the environment of this system. Note that individuals are also linked with the H_CLASS hierarchy, where they appear as the leaves of particular sortal concepts; they are, therefore, indexed by the H_CLASS hierarchical structure.

The events proper to a given domain (i.e. the dynamic processes describing the interactions among the important notions of the domain (concepts and individuals)) are represented by making use of the 'descriptive' and 'factual' tools.

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', 'having a negative attitude towards someone', 'be present somewhere'. In the context of the descriptive component, the events taken into consideration must be 'structured events', i.e. they must be characterized by the explicit mention of an actor, an object, an instrument, etc. Correspondingly – and in opposition to the binary data structures used for concepts and individuals – predicative templates are characterized by a *threefold format* where the central piece is a semantic *predicate*, i.e. a named relation that exists among one or more *arguments* introduced by means of *roles*:

$$(P_i(R_1a_1)(R_2a_2)...(R_na_n)),$$

(see the examples in section 3.1). Presently, the predicates pertain to the set {BEHAVE, EXIST, EXPERIENCE, MOVE, OWN, PRODUCE, RECEIVE}, and the roles to the set {SUBJ(ect), OBJ(ect), SOURCE, DEST(ination), MODAL(ity), TOPIC, CONTEXT}. Note that in NKRL, a predicate or a role is used only because it seems to express a functional relationship strictly necessary for the full appraisal of the meaning to be represented. In this sense, our roles are more similar to the 'correlators' of Ceccato's (1961, 1967) operational linguistics – where to each correlator corresponds a set of mental operations – than to the (syntactically constrained) Jackendoff's thematic roles. Templates are structured into an inheritance hierarchy, H_TEMP(lates) (see Figure 3, which corresponds, therefore, to a 'taxonomy (ontology) of events').

The instances ('predicative occurrences') of the predicative templates (i.e. the NKRL representation of single, specific events like "Tomorrow, I will move the wardrobe", "Lucy was looking for a taxi", "Mr. Smith has fired Mr. Brown", "Peter lives in Paris") are in the domain of the factual component. The reasons for distinguishing between descriptive and factual component are identical to those already examined in a definitional/enumerative context .

2.2 The upper layer

The upper layer of NKRL consists of two parts. The first is a catalogue which gives 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)

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

associated with the language. Presently, the basic templates are more than 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 specialization operations it is then possible to obtain from the basic templates the 'derived' templates that could be concretely needed to implement an application (e.g. "move a specific industrial process") and the corresponding occurrences. The set of legal templates included in the catalogue can be considered, at least in a first approach, as fixed. See section 5 for the advantages of this solution.

Note, however, that (if need be) the possibility of inserting new basic elements in the NKRL catalogue is not totally excluded. For example if, even in a general context of social interactions, it were absolutely necessary to put special emphasis on everyday life acts like 'eating' or 'drinking' (which are difficult to express in a simple way by making use of the seven predicates above), we should simply introduce a new predicate in the style of the Schankian INGEST, and add the corresponding new templates (derived, for example, from the syntactic structures of particular PRODUCE or RECEIVE templates) to the catalogue.

The second part of the upper layer is given by the general concepts which pertain to the upper levels of H_CLASS – such as *sortal_concepts*, *non_sortal_concepts*, *physical_entity*, *human_being*, etc. They are, as the upper levels of H_TEMP, invariable, i.e. they are not subjected to change when another application in a different domain is taken into account. These concepts form, therefore, a sort of upper-level, invariable ontology which can be compared with Bateman's 'Generalized Upper Model' (Bateman, Magnini and Fabris 1995), even if our selection criteria are, once again, strictly functional more than linguistically motivated. On the other hand, the lower level concepts are particular to a specific domain (e.g. *tax_on_agricultural_revenues* is an example of concept proper to a particular application in the fiscal law domain), and are subject to change.

3 Additional Information about the NKRL data structures

3.1 Descriptive and factual components

Figure 1 shows a simple example of NKRL code. It translates the NL fragment (a fragment of a Reuters' news story typical of the COBALT project) : "Brussels, July 2, 1993. British Airways Plc President Colin Marshall said in a Belgian newspaper interview the company's indebtedness was low following a capital increase".

In Figure 1, c1, c2 and c3 are the symbolic names of three 'predicative' occurrences, instances of basic NKRL templates (see also Figure 3); c4 is a 'binding occurrence'. Binding structures (i.e. tagged lists whose elements are symbolic labels, c2 and c3 in Figure 1) are second-order structures used to represent the logico-semantic links which can exist between predicative templates or occurrences. The binding occurrence c4 – meaning that event c2, the main event, has been caused by event c3 – is tagged using one (CAUSE) of the four operators which define together the taxonomy of causality of NKRL (Zarri 1992*a*, p. 705).

MOVE, EXPERIENCE and PRODUCE are predicates; SUBJ, OBJ,

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

```
c1)  MOVE          SUBJ      (SPECIF colin_marshall (SPECIF chairman british_airways)): (brussels_)
                   OBJ       #c4
                   DEST      (SPECIF newspaper_1 belgian_)
                   MODAL     interview_1
                   date-1:   2_july_93
                   date-2:

c2)  EXPERIENCE    SUBJ      british_airways
                   OBJ       (SPECIF indebtedness_1 low_)
                   [ obs ]
                   date-1:   2_july_93
                   date-2:

c3)  PRODUCE       SUBJ      british_airways
                   OBJ       (SPECIF capital_increase_1 british_airways)
                   date-1:   before_2_july_93
                   date-2:

c4)  (CAUSE c2 c3)
```

Fig. 1. An example of NKRL code.

DEST(ination), MODAL(ity) are roles. With respect to the arguments, colin_marshall, british_airways, brussels_, newspaper_1, interview_1, indebtedness_1, capital_increase_1 are symbolic labels of individuals (enumerative component); this means that they act as coreference links allowing us to access, from a descriptive component environment, information stored in the frame-like structures proper to the enumerative component. *chairman_*, *belgian_* and *low_* are labels of concepts (definitional component) which pertain to the *property_* subtree of the H_CLASS hierarchy. They allow us to access the corresponding frame-structures of the definitional component. (See section 4.2 for some comments on the SPECIF(ication) lists (lists of properties).) The arguments, and the templates/occurrences as a whole, may be characterized by the addition of particular codes which give further details about their significant aspects. For example, the 'location attributes' are associated with the arguments by using the colon code, ':', and are represented as lists (see occurrence c1). The presence of a 'temporal modulator', 'obs(erve)', leads to an interpretation of c2 as the description of a situation that, at this particular date, is observed to exist (Zarri 1992*b*).

For clarity's sake, we reproduce in Figure 2 the basic template (descriptive component) that gives rise to the occurrence c2 of Figure 1. Optional elements are in round brackets (e.g. the role SOURCE introduces the possible 'origin' of the situation). In the corresponding occurrences, the variables (x, y, u, etc.) are replaced by concepts (definitional component) or individuals (enumerative component) according to the associated constraints; constraints are, in turn, expressed in terms of the high-level elements (concepts) of the H_CLASS hierarchy (NKRL upper level).

Figure 3 is an extremely condensed representation of he H_TEMP hierarchy: we only give here some of the templates which pertain to the branches BEHAVE, MOVE and PRODUCE of the hierarchy – see Appendix A in Zarri (1992*a*) for more details. Given the space limitations, the syntactic description of the templates

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

{ EXPERIENCE1.1 ; 'being affected by a quantified type of situation'

IsA : EXPERIENCE1 ; 'being in a given state'

EXPERIENCE	SUBJ	*x* : [<*location_*>]
	OBJ	(SPECIF *y z*)
	(SOURCE	*u* : [<*location_*>])
	(MODAL	*v*)
	(EVENT	*w*)
	({'modulators'})	
	[date-1: <*date_*>, 'initial date' | 'observed date']	
	[date-2: <*date_*>, 'final date']	

x	=	<*human_being_or_social_body*>
y	=	<*situation_*>
z	=	<*property_*>
u	=	<*human_being_or_social_body*>
v	=	<*modality_*> | 'label of a predicative occurrence'
w	=	<*situation_framework*> }

Fig. 2. An example of template.

is also particularly sketchy, and only some very general indications are supplied. In particular, the symbolic labels have been suppressed, and the templates are discriminated making use merely of their associated NL comments. For a more realistic picture of the format and syntax of a template, see Figure 2. The code '!' means 'mandatory', and '≠', 'forbidden' (e.g. in the 'transmit an information to someone' template, MOVE subhierarchy, the presence of a DEST(ination) role is expressly required).

To conclude now about the NKRL structures of Figure 1, we can note that the MOVE template at the origin of the occurrence c1 in Figure 1 must be necessarily used to translate any sort of 'enunciative' situation, here, the transmission of an information ("Colin Marshall said ..."). It makes use of what, in NKRL's terminology, is called a 'completive construction'. According to this construction, the filler of the OBJ(ect) slot in the occurrences (here, c1) which instantiates the enunciation template is always a symbolic label (c4) which refers to another occurrence (here, a binding one) bearing the informational content to be spread out ("... company's indebtedness was low following a capital increase").

The enunciative situation can be both explicit, as in the 'Colin Marshall' example, or implicit. For example, the completive construction can be efficiently used to deal with a problem raised by Nazarenko (Nazarenko-Perrin 1992), and concerning the correct rendering of a causal situation where the general framework of the antecedent consists of an (implicit) speech situation. Let us examine one of the Nazarenko's examples (Nazarenko-Perrin 1992, p. 881): "Peter has a fever since he is flushed". As Nazarenko correctly remarks, 'being flushed' is not (obviously) the 'cause' of 'having a fever', but that of an implicit enunciative situation where we claim that someone has a fever. Using the completive construction, the above example is easily translated in NKRL by using the four occurrences of Figure 4.

As a further example of descriptive/factual structures, we give now, Figure 5,

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

. **H_TEMP** ("hierarchy of predicative templates")

.. **BEHAVE templates** ; predicate: BEHAVE

... "external manifestation of the subject" ; ≠ OBJ
.... "acting in a particular role" ; MODAL <*role_* [ex: *rugby_player*]>
.... "manifest a particular quality" ; MODAL <*quality_*>

... "focus on a result" ; ≠ OBJ, DEST, ARG ; ≠ mod. "against, for" ; ! GOAL binding structure
.... "act explicitly to obtain a result" ; ≠ modulator "ment"
.... "wishes and intentions" ; ! modulator "ment"

... "concrete attitude toward someone..." ; ! OBJ, MODAL; ≠ DEST, GOAL ; ≠ mod. "ment"

.. **EXIST templates** ; predicate: EXIST ; ! location of the SUBJ ; ≠ DEST

.. **EXPERIENCE templates** ; predicate: EXPERIENCE ; ! OBJ ; ≠ DEST

.. **MOVE templates** ; predicate: MOVE

... "moving a generic entity" ; OBJ <*entity_*>
.... "move a material thing" ; OBJ <*physical_entity*>
..... "change the position of something" ; ≠ DEST (ex: "move the wardrobe")
..... "transfer something to someone" ; ! DEST (ex: "send a letter to Lucy")

... "generic person displacement" ; SUBJ = OBJ = <*human_being*> ; ! loc. of SUBJ, OBJ

... "transmit an information to someone" ; ! DEST
.... "transmit a generic information" ; OBJ <*type_of_information* [ex: *message_*]>
.... "transmit a structured information" ; OBJ "label of binding/predicative occurrence"

.. **OWN templates** ; predicate: OWN ; ! OBJ ; ≠ DEST

.. **PRODUCE templates** ; predicate: PRODUCE ; ! OBJ

... "conceive a plan or idea" ; ! modulator "ment"

... "creation of material things" ; OBJ <*physical_entity* > ; ≠ modulator "ment"

... "perform a task or action" ; OBJ <*action_name* >
.... "acquire, buy" ; OBJ <*purchase_* >
.... "sell" ; OBJ <*sale_* >

... "relation involvement" ; SUBJ (COORD...) ; OBJ *mutual_agreem.* MODAL <*relationship_*>

... "production of events by an active cause" ; SUBJ <*active_cause* > ; OBJ <*event_*>

.. **RECEIVE templates** ; predicate: RECEIVE ; ! OBJ ; ≠ DEST

Fig. 3. A very condensed representation of the H_TEMP hierarchy.

an NKRL interpretation of the narrative sentence: "We have to make orange juice" which, according to Hwang and Schubert (1993, p. 1298), exemplifies several interesting semantic phenomena.

Figure 5 illustrates the standard NKRL way of representing the 'wishes, desires,

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

```
c5)   MOVE          SUBJ    human_being_or_social_body
                    OBJ     #c6

c6)   EXPERIENCE    SUBJ    peter_
                    OBJ     fevered_state_1

c7)   EXPERIENCE    SUBJ    peter_
                    OBJ     flushing_state_1
                    [ obs ]

c8)   (CAUSE c5 c7)
```

Fig. 4. Using a completive construction to represent an implicit enunciative situation.

```
c9)   BEHAVE        SUBJ    (COORD informant_1 (SPECIF human_being
                                (SPECIF cardinality_ several_)))
                    [oblig, ment]
                    date1:  observed date
                    date2:

c10)  PRODUCE       SUBJ    (COORD informant_1 (SPECIF human_being
                                (SPECIF cardinality_ several_)))
                    OBJ     (SPECIF orange_juice (SPECIF amount_))
                    date1:  observed date + i
                    date2:

c11)  (GOAL c9 c10)
```

Fig. 5. Representation of "wishes and intentions".

intention' domain. To translate the idea of 'acting in order to obtain a given result', we use:

(a) A (predicative) occurrence (here, c9), instance of a basic template pertaining to the BEHAVE branch of the H_TEMP hierarchy (see Figure 3), and corresponding to the general meaning of 'focusing on a result'. This occurrence is used to express the 'acting' component, i.e. it allows us to identify the SUBJ(ect) of the action, the temporal co-ordinates, possibly the MODAL(ity) or the instigator (SOURCE), etc.
(b) A second predicative occurrence, here c10, an instance of a template including a different NKRL predicate (e.g. PRODUCE in Figure 5) and which is used to express the 'intended result' component.
(c) A 'binding occurrence', c11, linking together the previous predicative occurrences and labelled with GOAL, another operator pertaining to the 'taxonomy of causality' of NKRL (Zarri 1992*a*).

'oblig' and 'ment' in Figure 5 are, like 'obs' in Figures 1 and 4, 'modulators' (Zarri 1992*a*), i.e. particular codes 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',

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

'obs(erve)' (Zarri 1992*b*). 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 c9 of Figure 5, 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, intention' (see Figure 3), the absence of the 'ment(al)' modulator in the BEHAVE occurrence means that the SUBJ(ect) of BEHAVE takes some concrete initiative (acts explicitly) to fulfil the result: if 'ment' is present, as in Figure 5, no concrete action is undertaken, and the 'result' reflects only the wishes and desires of the SUBJ(ect). *several_*, see c9 and c10, is used conventionally in NKRL to represent the cardinality of sets of totally undefined size, like those corresponding to a generic plural referent, as in 'men' or 'books'. In c10, *i* denotes an unknown time increment with respect to the time stamp ('observed date') that characterizes c9; for the concrete details concerning the representation of temporal data in NKRL, see again Zarri (1992*b*).

3.2 Definitional and enumerative components

Figure 6 gives an extremely abridged representation of the upper level of H_CLASS (hierarchy of concepts, definitional component). Note that in this figure, the presence of a sortal concept *event_*. As already stated, the events represented by means of templates and occurrences must be structured events, i.e. they must be characterized by the specific indication of an actor, an object, etc. If we only want to explicitly mention an event (e.g. president_clinton_interview_49), without giving all its structural details, we will simply make use of a particular individual, i.e. of an instance of *event_*. Within the predicative occurrences, we typically find such reified events as fillers of the role CONTEXT. We can also note in this figure that *substance_* and *colour_* are regarded in NKRL as examples of nonsortal concepts. For their generic terms, *pseudo_sortal_concepts* and *characterising_concepts*, we have adopted here the terminology of Guarino *et al.* (1994).

With respect to *substance_*, and with reference to the Lenat and Guha's example of 'peanut butter' (Guha and Lenat 1990), we will have in NKRL: (i) *butter_* is, like *gold_* or *wood_*, a *substance_*; (ii) *peanut_butter* is a specialization (not an instance) of *butter_* (and, *a fortiori*, of *substance_*) and it is still, therefore, a nonsortal concept without immediate instances; (iii) *piece_of_peanut_butter* is not an instance of *peanut_butter*, nor a specialization of this last concept, but a totally new concept deriving (through, for example, the *foodstuff_* subtree of *physical_entity*) from *sortal_concepts* – it is, in fact, possible to buy a particular instance (a pat of butter) of *piece_of_peanut_butter* (or of *loaf_of_bread*) in a supermarket.

There is, of course, a relationship between the sortal concept *piece_of_peanut_butter* (*piece_of_gold*), which accepts direct instances, and the nonsortal concept *peanut_butter* (*gold_*). Some 'intrinsic' properties of *peanut_butter* (*gold_*), i.e. some slots of the 'attribute' type (see below), like ColourOf and MeltingPoint, and their values – must also be associated, in fact, with *piece_of_....* To this end, NKRL makes use of specific 'user-defined relations', such as GetIntrinsicProperties, where the inheritance semantics (i.e. the information passing characteristics) indicating which

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

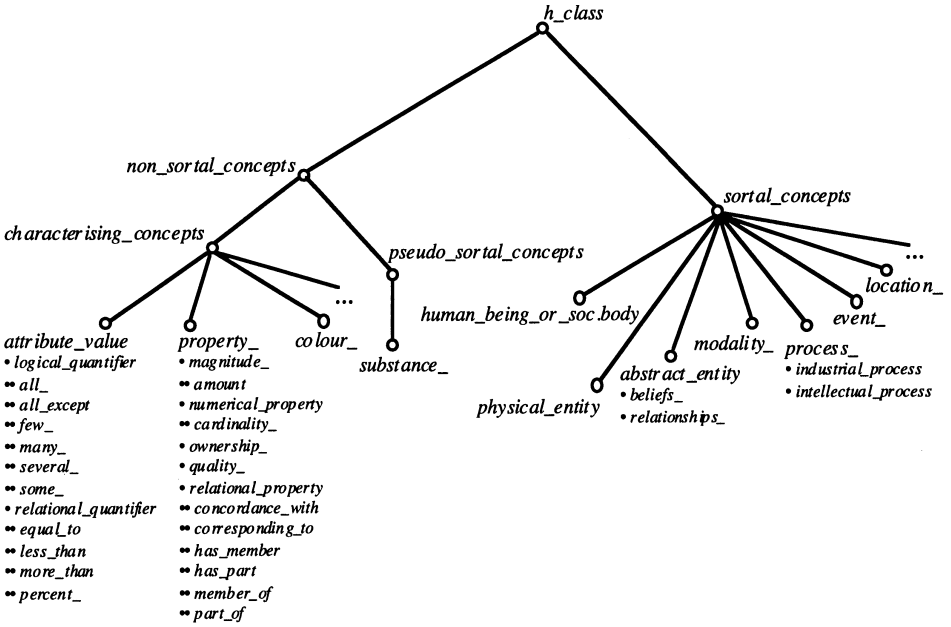


Fig. 6. An abridged view of the 'upper level' of H_CLASS.

slots and values can be inherited over that relation must be explicitly specified. This way of dealing with the problem of representing 'chunk of matters' allows us to, for example, avoid some of the intricacies of the ontology of CYC.

With respect to the nonsortal concept *colour_*, we deal with the problem of 'pseudoinstances' like "the red of Mary's coat" by representing this last example as "(SPECIF coat_1 mary_ *scarlet_*)" if we can determine the very precise specialization of *red_* that characterizes the coat – or simply as "(SPECIF ... *red_*)" if this is not possible. Note also that a notion like 'colour' (and its specializations) can be typically (and simultaneously) used in two different functions. For example, if we want to make use of the definition/description of this notion *per se*, we must then have recourse to the nonsortal concept *colour_*. If, at the same time, we are defining the notion of 'rose' (the flower), and we desire to utilize the notion of 'colour' to better define the sortal concept *rose_*, we must associate a property (in NKRL, attribute slots) ColourOf to this last concept (Zarri 1995*b*).

Coming now to the data structures used for concepts and individuals, they are essentially frame-based structures and their design is relatively traditional. These structures are composed of an OID (object identifier), and of a set of characteristic features (slots). Three different types of slots are used, 'relations', 'attributes' and 'procedures' (see Figure 7).

The slots of the 'relation' type, system-defined or user-defined, show the relationships of an NKRL entity, concept or individual, with other entities. For the time being, NKRL provides for eight system-defined relationships. They are: IsA, and the inverse HasSpecialization, InstanceOf, and the inverse HasInstance, MemberOf (HasMember) and PartOf (HasPart). Note that a concept or an individual

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

```
{ OID
        [ Relation      (IsA | InstanceOf :
                         HasSpecialisation | HasInstance :
                         MemberOf | HasMember :
                         PartOf | HasPart :
                         (UserDefined_1 :
                         ...
                         UserDefined_n : )
          Attribute     (Attribute_1 :
                         ...
                         Attribute_n : )
          Procedure     (Procedure_1 :
                         ...
                         Procedure_n : ) ] }
```

Fig. 7. A general schema for NKRL concepts and individuals.

cannot make use of all eight relations. The relation IsA, and the inverse HasSpecialization, are reserved to concepts. HasInstance can only be associated with a concept; InstanceOf with an individual. Moreover, MemberOf (HasMember) and PartOf (HasPart) can only be used to link sets (concepts) with concepts or instances with instances, but not concepts with instances (Schiel 1989). The slots of the 'attribute' type are used to represent some characteristic, intrinsic properties of a concept/individual, like the ColourOf property of the sortal concept *rose_* (see above).

Note also that we make use of only two meronymic relations, MemberOf and PartOf (and of their inverses) – our basic criterion for differentiating between the two is the homogeneity (HasMember) or not (HasPart) of the component parts. As is well known, six different meronymic relations are defined in (Winston, Chaffin and Herrmann 1987): component/integral object (corresponding to our PartOf), member/collection (corresponding to our MemberOf), portion/mass, stuff/object, feature/activity, place/area. Note that Winston *et al.* (1987) is still the reference paper for people interested mainly in the practical implications of making use of meronymic concepts; for an overview of some, more theoretical (and description logics-oriented) approaches, see Artale *et al.* (1996). The justification of our approach is twofold:

- Firstly, we want to keep the language as simple as possible. In this context, the only 'relations' that are absolutely necessary (in addition to IsA, InstanceOf and their inverses) are MemberOf and HasMember, given that they are systematically used to represent plural situations (see section 4.3).
- On the other hand, dealing systematically with the examples of 'non-NKRL relations' given by Winston and his colleagues by using only the existing NKRL tools leads to results which are not totally absurd, even if, sometimes, some aspects of the original meaning are lost. For example, "this hunk is part of my clay" (portion/mass) should also be interpreted as a MemberOf relation, like "this tree is part of the forest". We can, in fact, interpret 'my clay' as an

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

individual, e.g. generic_portion_of_clay_1, instance of *generic_portion_of_clay* that is a specialization of *physical_entity* and, therefore, of *sortal_concepts* – *clay_* is, in fact, a *substance_* (*non_sortal_concepts*) which cannot have direct instances in the clay_1 style. We can now see generic_portion_of_clay_1 as formed by several hunks, hunk_1 ... hunk_n which, like the trees in the forest, are all homogeneous (instances of the same concept *hunk_of_clay*) and play no particular functional role (as in the 'part of' examples) with respect to the whole represented by generic_portion_of_clay_1. Note that there is, once again, a relationship between the sortal *generic_portion_of_clay* and the nonsortal *clay_*, which concerns only the sharing of some 'intrinsic properties' (colour, density, etc.). Passing now to some different examples, "A martini is partly alcohol" (stuff/object) can be easily rendered using, for example, an 'attribute' slot, and "paying is part of shopping" can be rendered by making use of 'procedures' (see below). "An oasis is a part of a desert" (place/area) can be represented as a PartOf (there is no particular homogeneity amongst 'oasis', 'dune', 'dromedary' ...) ; etc.

Finally, the 'procedure' slots are used to store information about the dynamic characterization of (particularly important) concepts and individuals: we can find there the description of their typical behaviour, the 'instructions for the use', etc. Procedures can be pieces of procedural code implemented by using some usual mechanisms, like the 'demons'. However, a fundamental characteristic of NKRL consists in the possibility of implementing the procedures in a declarative style using the descriptive component templates (see Zarri (1995*b*) for an example, and section 4.3 (Figure 13)).

4 Structured arguments ('expansions') in NKRL

4.1 The AECS sublanguage

In both Figures 1 and 5, the arguments made up of SPECIF(ication) and/or COORD(ination) lists are examples of NKRL structured arguments (or 'expansions').

In NKRL, structured arguments of templates and occurrences are built up making use of a specialized sublanguage, AECS, which includes four binding operators, the 'disjunctive operator' (ALTERNative = A), the 'distributive operator' (ENUMeration = E), the 'collective operator' (COORDination = C), and the 'attributive operator' (SPECIFication = S), (Zarri 1994; Zarri and Gilardoni 1996). Their intuitive meaning is given in Table 1. Accordingly, structured arguments in NKRL are lists of undefined length, which may include both concepts and individuals and which are labelled by making use of the AECS operators.

From a formal semantics point of view, we can note that, for example:

$$\text{(a)} \quad (\text{SPECIF } e_1\ a\ b) = (\text{SPECIF } e_1\ b\ a)\ ;$$
$$\text{(b)} \quad (\text{ENUM } e_1\ e_2) = (e_1\ \text{AND}\ e_2\ \text{AND}\ \neg(\text{COORD } e_1\ e_2)).$$

The first formula says that the order of the properties $a, b, \ldots$ associated with an

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

Table 1. *NKRL operators for structured arguments*

Operator	Mnemonic description
ALTERN	The 'disjunctive operator'. It introduces a set of elements, i.e. concepts, individuals, or lists labelled with other expansion operators. Only one element of the set takes part in the particular relationship with the predicate defined by the role-slot to be filled with the expansion; however, this element is not known.
COORD	The 'collective operator': all the elements of the set take part (necessarily together) in the relationship with the predicate defined by the role-slot.
ENUM	The 'distributive operator': each element of the set satisfies the relationship, but they do so separately.
SPECIF	The 'attributive operator'. It is used to link a series of attributes (properties) with the concept or individual that constitutes the first element of the SPECIF list, to better characterize this last element. Each attribute appearing inside a SPECIF list can be recursively associated with another SPECIF list.

entity e_1, concept or individual, is not significant. The second enunciates in a formal way what already stated in Table 1: the main characteristic of the ENUM lists consists in the fact that the entities $e_1, e_2, \ldots$ take part obligatorily in the particular relationship between the structured argument and the predicate expressed by the role which introduces the argument, but they satisfy this relationships separately.

Because of their recursive nature, the AECS operators could give rise to very complex expressions, difficult to interpret and disentangle (unify). Therefore, to build up well-formed NKRL expansions, the definitions of Table 1 are used in association with the so-called 'priority rule', which can be visualized by using the following expression:

(ALTERN (ENUM (COORD (SPECIF))))

This is to be interpreted as follows: it is forbidden to use inside the scope of a list introduced by the binding operator *B*, a list labelled in terms of one of the binding operators appearing on the left of *B* in the priority expression above, e.g. it is impossible to use a list ALTERN inside the scope of a list COORD. Depending on this rule, the event represented by the NL expression: "Mr. Brown attended the reception at the White House; Mr. Smith also was there, accompanied by his daughter Jennifer (but the accompanying lady could also be his wife Lucy)", will be translated according to the representation of Figure 8 (note that Mr. Brown and Mr. Smith went separately to the reception, as shown by the use of ENUM).

Of course, in Figure 8, each of the single coding elements (mr_brown etc.) inside the expansions could be accompanied by a list of attributes (a SPECIF list). The specifications of a query language for AECS structures can be found in Zarri and Gilardoni (1996).

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

```
c12)  EXIST     SUBJ      (ALTERN (ENUM mr_brown (COORD mr_smith jennifer_smith))
                           (ENUM mr_brown (COORD mr_smith lucy_smith))):
                           white_house
                CONTEXT   reception_15 : white_house
```

Fig. 8. An example illustrating the 'priority rule'.

```
c13)  PRODUCE     SUBJ      berliner_bank
                  OBJ       (SPECIF purchase_1 (SPECIF stake_in_1 optimus_bank
                                  (SPECIF amount_ (SPECIF percent_ 50.00
                                  (SPECIF share_capital_1 optimus_bank)))))
                  date-1:   before_15_october_93
                  date-2:
```

Fig. 9. Examples of use of standardized SPECIF sublists.

4.2 Some peculiarities of the attributive operator, SPECIF

Before examining the relationships between expansions and plural expressions, we would like to add some comments on the use of the SPECIF(ication) operator, which is of crucial importance in NKRL. Particular, well-characterized ways of specifying the features of a given NKRL entity, concept or individual, are expressed by making use – within a general SPECIF list associated with this entity – of special SPECIF sublists labelled in a normalized way. These sublists utilize in fact, as their first element, a limited number of *characterising_concepts* pertaining to the *property_* and *attribute_value* subtrees of H_CLASS (see Figure 6): these concepts play, therefore, the role of specialized SPECIF suboperators. Among them, some of the most commonly used are listed in Table 2.

To illustrate the modalities of use of those specialized sublists, let us consider the coding, see Figure 9, of this fragment of financial news : "Berlin, October 15, 1993. Berliner Bank AG said it acquired a 50 percent stake in the Optimus Bank für Finanz-Service GmbH". The first part of this fragment, "Berliner Bank said ...", is translated by making use of a MOVE occurrence, similar to c1 in Figure 1. The message issued by Berliner Bank is represented by the occurrence c13 of Figure 9. According to the typical NKRL modality for dealing with 'actions', the purchase is represented making use of a PRODUCE construction, where the 'action name' ('acquisition, purchase') that corresponds to the surface verb ('acquire') is transformed into an individual (purchase_1), instance of an appropriate H_CLASS concept (Zarri 1992*b*). The most external SPECIF list represents, according to the general meaning of the 'attributive operator', some properties which can be asserted about its first element, purchase_1. In the present case, we say simply that the purchase concerns a financial interest in a company, stake_in_1. A second SPECIF list is used to give details about this stake. This second list specifies (i) the name of the company, optimus_bank, and (ii) the importance of the stake. In this last case, we use a new SPECIF list introduced by one of the *charactering_concepts* illustrated in Table 2, *amount_*; the elements of this list are used, as usual, to specify some properties of *amount_*, etc.

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

Table 2. *Concepts used as specialized SPECIF suboperators*

Concepts	H_CLASS	Remarks
all_except	*attribute_value*	The terms belonging to a SPECIF sublist where the first term is *all_except* represent the exceptions that characterize a given set. *all_except*, like *all_*, *few_*, etc., is one of the nonsortal concepts pertaining to the *logical_quantifier* subtree of H_CLASS, see Figure 6 above, that are often used to build up plural expressions, see below.
*amount_*	*property_*	A SPECIF sublist introduced by *amount_* is used to represent a feature expressing, implicitly or explicitly, a numerical quantity.
*cardinality_*	*property_*	A SPECIF sublist introduced by *cardinality_* is used to represent a feature expressing, implicitly or explicitly, the number of elements in a given set.
equal_to	*attribute_value*	*equal_to*, like *more_than*, *less_than* and *percent_*, pertains to the *relational_quantifier* subtree of *attribute_value*, see Figure 6.
has_member	*property_*	*has_member*, like *concordance_with*, *corresponding_to*, *has_part*, *member_of*, *part_of*, pertains to the *relational_property* sub-tree of *property_*, see Figure 6.
*ownership_*	*property_*	Adding to an NKRL entity a feature expressed by a SPECIF sublist introduced by *ownership_* makes it clear that this entity is owned by the character(s) or social bod(y/ies) following *ownership_* in the sublist.
*percent_*	*attribute_value*	A SPECIF sublist introduced by *percent_* is used to represent a feature expressing, implicitly or explicitly, a numerical percentage. These sublists are typically, but not necessarily, used within an *amount_* list.

4.3 Structured arguments, and plural entities and expressions

The definitions given in Table 1 for the COORD and ENUM operators show clearly that they must intervene in the NKRL representation of plural expressions. They correspond, in fact, to two of the 'plural quantifiers' normally used to represent the reading variants of plural nouns phrases : COORD represents a 'collective reading', and ENUM a 'distributive reading'.

Researchers working on the treatment of plurals and plural expressions have usually focused their efforts on the representation, according to a strict logical approach, of the ambiguities which affect quantification of plural sentences like "Three boys bought five roses". We can mention, in this context Link (1983), Webber (1983) and Scha and Stallard (1988). An, at least partial, exception to this 'predicate calculus extensions' approach is represented by Franconi's work (Franconi 1993), executed mainly in a terminological logic context. In NKRL, we have adopted

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

```
c14)    PRODUCE        SUBJ    Mary_
                       OBJ     (SPECIF purchase_27 book_1 book_2 book_3)

        [ book_1
                InstanceOf  :  book_
                HasMember   :  nil ]

        [ book_2
                InstanceOf  :  book_
                HasMember   :  nil ]

        [ book_3
                InstanceOf  :  book_
                HasMember   :  nil ]
```

Fig. 10. A simple example of NKRL plural representation.

a strictly pragmatic approach to the 'plurals problem', that shares some similitudes with Sowa's solutions (Sowa 1991).

Our proposal does not require the introduction of extra operators, but relies only on the standard NKRL tools used for the representation of sets, namely:

- At the level of the descriptive and factual components, the four binding operators of the AECS sublanguage discussed in section 4.1.
- At the level of the definitional and enumerative components, the relation slots, MemberOf and HasMember, introduced in section 3.2.

The basic NKRL modality for translating NL plural expressions *that concern real 'factual' events* consists then in the two following steps:

(a) to construct, within an occurrence, a structured argument (expansion) where the conceptual labels of the concepts and/or individuals which make up the plural set evoked by the NL expression are explicitly mentioned, and
(b) to create, at the definitional/enumerative level, the 'frames' which supply the definitions (concepts) or the descriptions (individuals) of the entities denoted by those labels.

For example, the narrative expression "Mary has bought three books" will cause the creation of the four structured objects of Figure 10, an occurrence, c14, and three individuals, book_1, book_2 and book_3. We suppose here that the acquisition of the three books happened in the framework of a single purchase event (purchase_27). In Figure 10, *book_* represents a sortal concept; the empty slot HasMember in the three individuals of this figure makes it clear that each of them represents a singleton, i.e. a specific, unique instance of the concept *book_*.

However, this sort of representation cannot be systematically used, especially when:

- The NL plural expression to be represented requires the use of general quantifiers like *some_*, *many_*, *all_*, ..., as in "Mary has bought some books". In this case, the elements of the set cannot be explicitly enumerated and associated with specific individuals.

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

```
c15) PRODUCE   SUBJ   Mary_
               OBJ    (SPECIF purchase_27 (SPECIF book_17 (SPECIF cardinality_
                        many_)))

[ book_17
        InstanceOf  :  book_
        HasMember   :  many_ ]
```

Fig. 11. Representation of a plural expression including a general quantifier.

- The elements of the set could be explicitly enumerated, but they are too numerous and, at the same time, 'uninteresting', so that the explicit construction of an individual for each of them is both uneconomic and uninformative. An example of this situation could be "Mary has bought thirty-two books".

The standard NKRL representation of an NL plural expression consists then of a generalization of the above basic proposal, more precisely:

- In the occurrence, we make use of an unique individual (of its label) to represent the set in its entirety (e.g. the set including the three books in the example above). The individual is accompanied by a *cardinality_* sublist – see Table 2: this particular SPECIF sub-list will include the quantifier, or the numeral, expressing the cardinality of the set.
- The same numeral or quantifier must be inserted in the HasMember slot of the frame associated with the (unique) individual mentioned in the occurrence.

In this way, an example such as "Mary has bought many books" will have the representation of Figure 11. The non-empty HasMember slot in the individual, book_17, of Figure 11, makes it clear that this individual is referring, now, to several instances of the concept *book_*. As already stated, *several_* is used conventionally in NKRL to represent the cardinality of sets of totally undefined size.

From a more theoretical point of view, we can remark that individuals like book_17 are 'collections' rather then 'sets', given that the 'extensionality axiom' does not hold here: this axiom says that two sets are equal iff they have the same elements. In our framework, two collections, say book_17 and book_18, can be co-extensional, i.e. they can include exactly the same elements, without being necessarily considered as identical if created at different moments in time in the context of totally different events (Franconi 1993).

We can now illustrate how it can be possible to use the above mechanisms for representing the ambiguities which affect quantification of plural sentences: as already stated, this is the touchstone of any formal theory concerning plural expressions.

Webber says that a statement like "three boys bought five roses" (Webber 1983, p. 346) can be used to convey a 'distributive' meaning (each one of the three boys bought five roses), a 'collective' meaning (the three boys, formed into a consortium, bought five roses), or a 'conjunctive' meaning (the total of rose-buying boys is three, the total number of roses, each of which was bought by some boys, is five, but the speaker either does not know or does not care to tell the listener

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

```
c16)   PRODUCE    SUBJ    (ENUM (SPECIF boy_42 (SPECIF cardinality_ 3)))
                  OBJ     (SPECIF purchase_75 (SPECIF rose_97 (SPECIF cardinality_ 5)))

c17)   PRODUCE    SUBJ    (COORD (SPECIF boy_42 (SPECIF cardinality_ 3)))
                  OBJ     (SPECIF purchase_75 (SPECIF rose_97 (SPECIF cardinality_ 5)))

c18)   PRODUCE    SUBJ    (SPECIF boy_42 (SPECIF cardinality_ 3))
                  OBJ     (SPECIF purchase_75 (SPECIF rose_97 (SPECIF cardinality_ 5)))

        [ boy_42
                InstanceOf : boy_
                HasMember : 3 ]

        [ rose_97
                InstanceOf : rose_
                HasMember : 5 ]
```

Fig. 12. NKRL representation of Webber's example about boys and roses.

how boys match up with roses: this can constitute the default interpretation). She gives a formal representation for the three cases by extending standard logic with the introduction of the operators 'set', 'cardinality' and 'equality' (Webber 1983, pp. 350–353). Sowa (1991, p. 178) calls explicitly 'Default' the 'conjunctive' reading. From a formal point of view, the 'generic plural' referent is translated by Sowa with '{*}', where the symbol '{}' is used for sets and the asterisk for the undefined referent. The symbol '{*}' alone represents then the 'Default' interpretation, consistent with either the collective or the distributive readings; these last readings are represented by explicit prefixes, giving rise then to complex symbols like 'Dist{*}' and 'Coll{*}'. Franconi (1993, pp. 456–459) maintains an explicit denomination for Webber's conjunctive reading, which is now called 'cumulative'. He makes use of special operators, called 'plural quantifiers', to represent the cumulative and distributive readings, and leaves the collective expressions non-quantified.

In NKRL, we assume, like Sowa, the conjunctive/cumulative reading as the 'default' interpretation, which does not require the use of any particular symbol/operator. To represent the residual two readings, no need exists for introducing new *ad hoc* entities: according to the definitions given in Table 1, the ENUM operator represents perfectly the 'distributive reading' and COORD the 'collective reading'. Thus, the NKRL coding of Webber's example is given in Figure 12: occurrence c16 represents the distributive reading, c17 the collective reading, c18 the default. The NKRL representations of Sowa's examples, "nine ladies are dancing" (default interpretation), "nine ladies are each dancing" (ENUM) and "nine ladies are dancing together" (COORD), or "five blocks are supported by three pyramids" (Sowa 1991, pp. 178–181), can be solved by using the same techniques – the extension, using ALTERN (Table 1), to a possible formulation in terms of "one of nine ladies is dancing", is straightforward. It is evident that, according to this type of representation, the operators ENUM and COORD are added only when it becomes absolutely necessary, for a correct interpretation of the event examined, to indicate explicitly the type of reading that has been favoured.

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

We can now conclude this discussion with some general remarks. If we consider some examples in the style of "many elephants that perform in a circus earn money" (Sowa 1991, p. 175), or "every trailer truck has 18 wheels" (Sowa 1991, p. 177) (or, in the same vein, "some cotton T-shirt is expensive" (Webber 1983, p. 348)), we can easily realize that we are confronted here with quantified plural expressions that – at the difference of those considered until now – do not refer to concrete elementary events characterized by precise spatial and temporal co-ordinates. These examples do not concern the domain of the factual component: given their appearance of 'general wisdom', and the lack of references to any specific character(s) operating in a well-defined, spatio-temporal framework, they concern rather the *definitional component* domain, and more precisely the definition of sortal concepts like *circus_elephant* and *trailer_truck* (or *cotton_T_shirt*).

Sowa deals with the lack, in his theory, of this sort of explicit differentiation by using the usual, 1st-order, logic-inspired tools. In the 'many elephants' example, he utilizes a 'λ-expression' to represent the concept node "elephant that performs in a circus". λ-expressions are, in conceptual graphs, the standard way to introduce new definitions, here, the subtype of elephants defined in the relative clause. In the 'every trailer truck' example, the 'referent field' of the concept node having 'TRAILER-TRUCK' as the type field is filled with the universal quantifier.

From an NKRL point of view, all we need to represent the example every trailer truck is to add, in the frame corresponding to the concept *trailer_truck*, an 'attribute slot' (see section 3.2) like 'NumberOfWheels', having '18' as default value. For *circus_elephant*, if, on the one hand, a solution in terms of an attribute slot 'EarnsMoney' seems too trivial and, on the other, we want to fully preserve the possibility of explicitly expressing the alternatives among 'every elephant', 'some elephants' and 'many elephants', an elegant and complete solution consists in adding a 'declarative procedure' (section 3.2), e.g. 'Procedure121', to the defining frame of *circus_elephant*. The representation of Procedure121 is given by the three predicative templates of Figure 13, specializations of a basic RECEIVE template, and by a binding template t4. Note that, in accordance with the fact of being now in a 'definitional' and not 'factual' context, the structures of Figure 13 are characterized by the presence of 'concepts', like *circus_elephant* and *money*, instead of 'individuals' (see, for example, purchase_75 or rose_97 in Figure 12).

5 Conclusion

The problem of defining a reliable and useful semantic (conceptual) representation of the 'meaning' of narrative texts is, of course, not really new, and it has been studied for decades in the NL and AI communities. Several (at least partly) implemented formalisms exist that claim to be able to represent extensive chunks of natural language (NL) semantics – see Conceptual Graphs (Sowa 1984), CYC (Lenat and Guha 1990), LILOG (Herzog and Rollinger 1991), SNePS (Shapiro and Rapaport 1992), Episodic Logic (Hwang and Schubert 1993), etc. Moreover, since Schank's work (Schank 1973), several authors have tackled the peculiar problem of representing structured events – in addition to the specific applications of the previous

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

```
[ circus_elephant
                IsA : elephant_
...
                Procedure121 : t4
...
t1)     RECEIVE     SUBJ      (SPECIF circus_elephant (SPECIF cardinality_ all_))
                    OBJ       money_
                    MODAL     earnings_

t2)     RECEIVE     SUBJ      (SPECIF circus_elephant (SPECIF cardinality_ some_))
                    OBJ       money_
                    MODAL     earnings_

t3)     RECEIVE     SUBJ      (SPECIF circus_elephant (SPECIF cardinality_ many_))
                    OBJ       money_
                    MODAL     earnings_

t4)     (ALTERN t1 t2 t3) ]
```

Fig. 13. Using 'declarative procedures' to define an NKRL concept.

formalisms, we can also mention here some papers in the description logics tradition (see, for example, DiEugenio (1994)), or the recent proposal by Park (1995), 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, temporal information, etc.

In defence of the introduction of the NKRL formalism, we can put forward (at least) the two following arguments:

- Making use of a relatively simple, intuitive and easily manageable formalism, NKRL offers some interesting solutions to very hard problems concerning the 'practical' aspects of the knowledge representation endeavour. These include, for example, the representation of implicit and explicit enunciative situations, of wishes, desires and intentions (modals), of plural situations, of causality, and of other second order, intertwined constructions. We can also note that the architecture of NKRL allows now to distinguish exactly between a (possibly very detailed) representation of events *per se* – as instances of the structures (templates) proper to the event ontology – and the customary use of reified events as instances of the concepts pertaining to the *event_* subtree of the concept ontology.
- A second, important characteristic of NKRL is represented by the fact that the catalogue of basic templates (see section 2.2) can be considered as part and parcel of the definition of the language. This approach is especially important for practical applications, and it implies, in particular, that: (i) a system-builder does not have to create himself the structural knowledge needed to describe the events proper to a (sufficiently) large class of narrative texts and documents; (ii) it becomes easier to secure the reproduction or the sharing of previous results.

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

It could be asserted, finally, that at least some of the representation principles used in NKRL reflect sufficiently well some intrinsic characteristics of the narrative domain; they should, therefore, be taken into consideration when trying, in an information interchange context (see Neches *et al.* (1991) and Gruber (1993), etc.) to set up tools for representing narrative information in a standardized ('interlingua') way.

References

Ali, S. S. (1995) ANALOG: A Knowledge Representation System for Natural Language Processing. In: E. A. Yfantis (ed.), *Intelligent Systems*. Kluwer Academic.

Artale, A., Franconi, E., Guarino, N. and Pazzi, L. (1996) Part-whole relations in object-centered systems: An overview. *Data & Knowledge Engineering*, **20**: 347–383.

Bateman, J. A., Magnini, B. and Fabris, G. (1995) The generalized upper model knowledge based organization and use. In: J. I. Mars (ed.), *Towards Very Large Knowledge Bases – Knowledge Building and Knowledge Sharing*. IOS Press.

Ceccato, S. (ed.) (1961) *Linguistic Analysis and Programming for Mechanical Translation*. Gordon and Breach.

Ceccato, S. (1967) Concepts for a new systematics. *Information Storage and Retrieval*, **3**: 193–214.

Di Eugenio, B. (1994) Action representation for interpreting purpose clauses in natural language instructions. In: J. Doyle, E. Sandewall and P. Torasso (eds.), *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*. Morgan Kaufmann.

Franconi, E. (1993) A treatment of plurals and plural quantifications based on a theory of collections. *Minds and Machines*, **3**: 453–474.

Gruber, T. R. (1993) Ontolingua: A mechanism to support portable ontologies. *Knowledge Acquisition*, **5**: 199–220.

Guarino, N., Carrara, M. and Giaretta, P. (1994) An ontology of meta-level categories. In: J. Doyle, E. Sandewall and P. Torasso (eds.), *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*. Morgan Kaufmann.

Guha, R. V. and Lenat, D. B. (1990) Cyc: A midterm report. *AI Magazine*, **11**(3): 32–59.

Herzog, O. and Rollinger, C.-R. (eds.) (1991) *Text Understanding in LILOG – Integrating Computational Linguistics and Artificial Intelligence*. Springer-Verlag.

Hwang, C. H. and Schubert, L. K. (1993) Meeting the interlocking needs of LF-computation, deindexing and inference: An organic approach to general NLU. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence – IJCAI/93*. Morgan Kaufmann.

Iwanska, L. (1993) Logical reasoning in natural language: It is all about knowledge. *Minds and Machines*, **3**: 475–510.

Jackendoff, R. (1990) *Semantic Structures*. MIT Press.

Lenat, D. B. and Guha, R. V. (1990) *Building Large Knowledge Based Systems*. Addison-Wesley.

Link, G. (1983) The logical analysis of plurals and mass terms: A lattice-theoretical approach. In: J. A. G. Groenendijk, T. M. V. Janssen and M. B. J. Stokhof (eds.), *Meaning, Use and Interpretation of Language*. De Gruyter.

Minsky, M. (1975) A framework for representing knowledge. In: P. H. Winston (ed.), *The Psychology of Computer Vision*. McGraw-Hill.

Nazarenko-Perrin, A. (1992) Causal ambiguity in natural language: Conceptual representation of 'parce que/because' and 'puisque/since'. *Proceedings of the 15th International Conference on Computational Linguistics – COLING 92*, Nantes, France.

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press

Neches, R., Fikes, R., Finin, T., Gruber, T. R., Patil, R., Senator, T. and Swartout, W. (1991) Enabling technology for knowledge sharing. *AI Magazine*, **12**(3): 36–56.

Park, B. J. (1995) A language for ontologies based on objects and events. *Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada.

Scha, R. and Stallard, D. (1988) Multi-level plurals and distributivity. *Proceedings of the 1988 Annual Meeting of the Association for Computational Linguistics*. ACL.

Schank, R. C. (1973) Identification of conceptualizations underlying natural language. In: R. C. Schank and K. M. Colby (eds.), *Computer Models of Thought and Language*. W. H. Freeman.

Schiel, U. (1989) Abstractions in semantic networks: Axiom schemata for generalization, aggregation and grouping. *ACM Sigart Newsletter*, (107): 25–26.

Shapiro, S. C. and Rapaport, W. J. (1992) The SNePS Family. In: F. Lehmann (ed.), *Semantic Networks in Artificial Intelligence*. Pergamon.

Sowa, J. F. (1984) *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.

Sowa, J. F. (1991) Toward the expressive power of natural language. In: J. F. Sowa (ed.), *Principles of Semantic Networks*. Morgan Kaufmann.

Webber, B. L. (1983) So what can we talk about now. In: M. Brady and R. C. Berwick (eds.), *Computational Models of Discourse*. MIT Press.

Winston, M. E., Chaffin, R. and Herrmann, D. (1987) A taxonomy of part-whole relations. *Cognitive Science*, **11**: 417–444.

Yokoi, T. (1993) Very large-scale knowledge bases: From lexical knowledge to world knowledge. *Proceedings of the 13th International Joint Conference on Artificial Intelligence – IJCAI/93*. Morgan Kaufmann.

Zarri, G. P. (1992*a*) The 'Descriptive' component of a hybrid knowledge representation language. In: F. Lehmann (ed.), *Semantic Networks in Artificial Intelligence*. Pergamon.

Zarri, G. P. (1992*b*) Encoding the temporal characteristics of the natural language descriptions of (legal) situations. In: A. Martino (ed.), *Expert Systems in Law*. Elsevier.

Zarri, G. P. (1994) A glimpse of NKRL, the 'Narrative Knowledge Representation Language'. In: S. Ali (ed.), *Knowledge Representation for Natural Language Processing in Implemented Systems – Papers from the 1994 Fall Symposium*. AAAI Press.

Zarri, G. P. (1995*a*) Knowledge acquisition from complex narrative texts using the NKRL technology. In: B. R. Gaines and M. Musen (eds.), *Proceedings of the 9th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada.

Zarri, G. P. (1995*b*) An overview of the structural principles underpinning the construction of 'ontologies' in NKRL. *Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada.

Zarri, G. P. and Gilardoni, L. (1996) Structuring and retrieval of the complex predicate arguments proper to the NKRL conceptual language. In: Z. Ras and M. Michalewicz (eds.), *Proceedings of the 9th International Symposium on Methodologies for Intelligent Systems – ISMIS'96*. Springer-Verlag.

https://doi.org/10.1017/S1351324997001794 Published online by Cambridge University Press