AN APPROACH TO NATURAL LANGUAGE IN THE SI-NETS PARADIGM

Amedeo Cappelli, Lorenzo Moretti
Istituto di Linguistica Computazionale-CNR
Via della Faggiola, 32
56100 Pisa - Italy

ABSTRACT

This article deals with the interpretation of conceptual operations underlying the communicative use of natural language (NL) within the Structured Inheritance Network (SI-Nets) paradigm. The operations are reduced to functions of a formal language, thus changing the level of abstraction of the operations to be performed on SI-Nets. In this sense, operations on SI-Nets are not merely isomorphic to single epistemological objects, but can be viewed as a simulation of processes on a different level, that pertaining to the conceptual system of NL. For this purpose, we have designed a version of KL-ONE which represents the epistemological level, while the new experimental language, KL-Conc, represents the conceptual level. KL-Conc would seem to be a more natural and intuitive way of interacting with SI-Nets.

I GOALS

The goal of our work is to interpret conceptual operations underlying the communicative use of natural language within the Structured Inheritance Networks (SI-Nets) paradigm. In other words, this means using epistemological primitives such as Concepts, Roles and Structural Descriptions (Brachman, 1979), to represent these conceptual operations.

On the one hand, epistemological formalism, which is explicit and clear, can clarify the behaviour of conceptual operations of NL.

By the use of SI-Nets formalism as a means of description, a new perspective can be brought out, since this formalism makes it possible to represent objects as data types structured in a complex way instead of considering them as mere atomic elements. This feature is likely to change the nature of the operations to be carried out on objects thus leading us to deal with the complexity of many phenomena in a more adequate way.

On the other hand, this can lead to an investigation of the relationships between the conceptual aspects of NL and the epistemological primitives, in order to discover how the latter are used by the previously mentioned operations. In fact, we attempt to find out whether an isomorphism exists between objects and operations of NL and those used by epistemology.

According to Brachman (1979), five different approaches to the representational problem can be established: implementational, logical, epistemological, conceptual and linguistic. Each of them uses its own primitives so that the five levels can be interpreted as a hierarchy where each level involves different degrees of abstraction.

By virtue of this interpretation, we have tried to extend epistemology in a conceptual perspective. Our current approach considers epistemology as a starting point, thus looking at the conceptual level as one of the possible target points. This goal can be achieved by changing the level of abstraction of the operations to be performed on SI-Nets. Consequently, operations on SI-Nets could assume a different aspect, that is to say they could be viewed not as merely isomorphic to single epistemological objects but as a simulation of operations lying on a different level, for instance, that pertaining to the conceptual system of NL. This hypothesis can reduce SI-Nets to the level of an internal mechanism covering only abstract data representation, whose structure is not transparent to the user. In this case the user interacts with the internal system by means of a separate external framework.

In order to achieve this goal we have designed and implemented a language, KL-Magma which represents our epistemological level. We are now designing and implementing an experimental language, KL-Conc, which should cover the conceptual level and which uses KL-Magma as one of its internal components.

The rest of the article will be devoted to a description of these two languages introducing considerations concerning their relevance to linguistic analysis and knowledge representation. We are confident that our approach can have interesting implications for both these fields, since KL-Conc functions can be used to describe linguistic entities in terms of conceptual operations and may be viewed as a more natural way of interacting with SI-Nets.

II KL-MAGMA

KL-Magma is a version of KL-ONE implemented in MAGMA-Lisp (Asirelli et al. 1975).

It is a KR language similar to the one described in Brachman (1979), Brachman et al. (1978), which also takes into account the versions given in Cappelli and Moretti (1982) and Porta and Vinchesi (1982). As in KL-One, KL-Magma formal objects are Concepts, Roles and Structural Descriptions.

Concepts are descriptional structures providing an intensional representation of the domain, i. e. prototypes and individuals.

Roles are descriptional structures representing parts of Concepts, i. e. properties of prototypes and individuals.

Structural Descriptions are sets of relationships between Roles which give a wholistic structure to Concepts.

Objects are connected with one another via Cables and Wires, thus realizing Structured Inheritance.

In our current approach KL-Magma is mainly used as a declarative model of abstract data structures. It has no mechanism like the MSS Algorithm (Woods, 1981) or the KL-One Classifier (Schmolze and Lipkis, 1983) which cover procedural aspects lying within epistemology, thus reaching valuable results in discovering new types of logic by deeply exploiting SI-Nets semantics. Instead, we have tried to discover types of procedurality external to the epistemological level and pertinent to the level we intend to represent. At any rate, we intend to govern epistemological processes by the external mechanism. In other words, this means assuming, for instance, the logic of subsumption , which is peculiar to epistemology, not as an autonomous deductive mechanism, but, instead, as a possible process controlled by the functions of the higher level language.

III WHAT TYPE OF CONCEPTUAL OPERATIONS ?

The conceptual operations of NL we intend to interpret are, for instance, individuations of objects, evaluations of objects, evaluations of properties of objects, evaluations of configurations of objects and so on.

Operations of this kind are triggered by articles, adjectives, prepositional phrases, relative clauses and so on. These operations, already intuitively described in classical Linguistics, have been given more attention by investigations based on Logic. In the logic paradigm they can be viewed as classes, sets, predicates etc..

In our opinion, the nature of these operations and, consequently, the description we intend to give of them, are not completely covered by logical analysis. Interesting results have been obtained by combining traditional logical systems with extensions of lambda calculi (Webber, 1978;1981). However, the types of complex procedurality peculiar to the operations have not yet been given a precise description; that is to say, procedurality has not been reduced to definite sets of restricted and clear procedures.

Let us now introduce an example. The Italian definite and indefinite articles (il, un) can be described as follows:

a) individuation of a specific object;

b) individuation of any one object;

c) reference to an abstract prototype.

In terms of logical description a) and b) may correspond to the iota operator and the existential quantifier of Logic; c) is similar to the universal quantifier even if the notion of a prototype is different, since it has an intensional nature.

However, we think that the three possible descriptions of Italian articles may include types of operations not covered by the use of the above mentioned logical operators. The article, like many other linguistic entities, integrates different kinds of operations which, at the same time, manipulate descriptions of prototypes and individuals, search into different kinds of memory, etc.

Let us introduce a new example. The adjective is one of the more complex phenomena of NL which cannot be reduced to the notion of predicate since it triggers a set of reasoning processes, that is to say, the manipulation of parts of knowledge.

The following NP:

1. un bambino rosso

may be interpreted as: a child has hair, hair has a color, the color can be red. This NP cannot be literally translated into English without adding more information; the appropriate translation is : a red-haired child.

In terms of SI-Nets this process can be represented as shown in figure 1, assuming that every lexical item of the NP has its own intensional representation.

Figure 1

However, the adjective does not specify all the steps of the reasoning process that it triggers, but only indicates, together with the name, the two extreme points of the chain leaving the intermediate undefined. The entire process, using generic knowledge as the reference point, is shown in Figure 2.

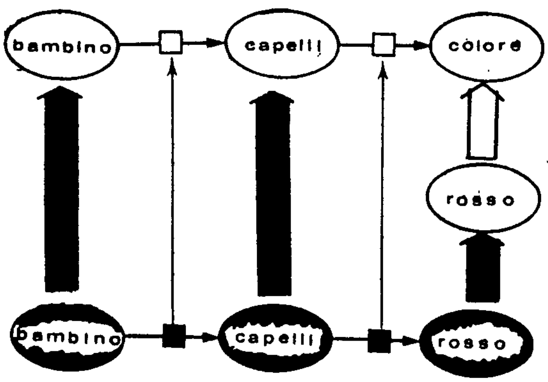

Figure 2

It would be oversimplifying, as stated above, to use the notion of predicate to interpret this complex process as well as the other possible interpretation of the adjective: the one corresponding to the notion of "type of" as in the NP "a red color" (see figure 3).

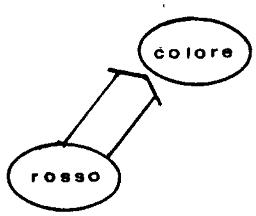

Figure 3

This type of phenomena can be investigated by deeply exploiting the structure and the semantics of SI-Nets. The structure of a role can be used as configuration of objects which are likely to be manipulated by complex processes not yet deeply investigated from any other viewpoint than the epistemological one. Once considered as a complex link, as it actually is, a role may be the locus where different processes can be triggered. It may be used simply to satisfy a structure of another role lying higher within the network or to trigger the complex processes we were talking about. The two behaviours mentioned exhibit different levels of abstraction; in the former case this means performing epistemological operations, while in the latter we simulate processes of a conceptual system used by NL.

IV WHY A NEW LANGUAGE?

The question now arises whether it is possible to reduce these types of operations to a set of functions of a formal language each of which covers a well defined process which corresponds to a well defined set of operations on SI-Nets - to a set of KL-Magma functions.

The choice of a new language has many motivations:

a) from the conceptual viewpoint, this means reducing operations to functions that are well defined from a semantic viewpoint which lend clearness to the process to be represented.

a) from the epistemological viewpoint, it is reasonable to think that a language, such as KL-Magma, may be extended by another language thus achieving a higher degree of abstraction.

c) a language is a uniform mechanism for the integration of interpreters of several symbolic processes. This integration is likely to bring out more clearly relevant phenomena of the process represented.

V KL-CONC

On the basis of the linguistic assumptions previously outlined and using KL-Magma as a language which handles SI-Nets, we are now designing and implementing an experimental language, KL-Conc, whose functions try to simulate the conceptual operations previously described.

A. KL-Conc: Internal Organization

Before describing KL-Conc functions in detail, it is worth while discussing its internal organization.

In the framework of KL-ONE, a relevant distinction has been drawn between the Terminological Box (T-Box) and the Assertional Box (A-Box) (Brachman, 1981). The T-Box maintains the detailed description of the objects while the A-Box contains the set of the assertions on the objects. The former corresponds to the ability of describing by the use of NPs, and the latter to that of constructing complex sentences.

A discussion has arisen whether it is possible to handle the two boxes, which correspond to two different areas of memory, using the same language.

In KL-ONE, new functions have been added in order to give it an assertional power (nexus, context) (Woods 1979).

A recent extension of KL-ONE (Brachman et al., 1983) has adopted the solution of creating two distinct languages: one for the T-Box and the other for the A-Box. The former is a sort of KL-ONE viewed in a functional way while the latter is a language based on First Order Predicate Logic.

KL-MAGMA is only able to handle the T-Box and it has no assertional power. Instead, by KL-Conc we are trying to design a language which covers both terminological and assertional aspects, even if it is more biased towards assertionality. It is our intention to handle the T-Box mainly in an assertional way.

In order to achieve this goal we have introduced the distinction between Long Term Memory (LTM) and Working Memory (WM) which in part covers the traditional one between T-Box and A-Box.

The LTM is represented in KL-Magma data structures; this contains descriptional knowledge about generic and individual objects.

The WM contains the history of the objects organized in a structured way. This is the central component of our current hypothesis. The WM contains the traces of contextual relationships between objects, as well as operations triggered on and by objects; it can also contain other symbolic systems. The task of the WM is mainly to hold hypotheses to be mapped onto the LTM which requires the cooperation of several interpreters.

The introduction of a larger number of memory spaces increases the power of the language. For instance, a structured WM is likely to improve the number of symbolic systems interacting with one another. This makes it possible to insert into the language functions based on different processes. Taking for instance the history of the objects as a reference point, the objects themselves can be accessed according as they appear in the time flow. The function:

<LAST arbitrary_name>

returns the last object, created or manipulated, belonging to the class named by arbitrary_name. In other words, this allows the user to refer to objects using anaphorical references, that is to say using a symbolic system which is organized and represented in a different way from epistemology.

By the WM we are trying to create the basic mechanism to handle these types of processes.

B. KL-Conc: External Organization

KL-Conc functions handle real world objects, so the user only needs to know a set of functions to be applied to objects. In this way, the structure of the SI-Net which internally organizes the data, is hidden; the only data which are transparent are objects, which may be individual or generic, together with syntactic rules for combining functions. These last are very flexible. Objects can be accessed using arbitrary names or by means of syntactic combinations which conceptually correspond to complex tests on the nature of objects, the configuration of objects etc.. Objects can be accessed according as they appear in the time flow.

The user can use the same name both for generic and individual objects. This is made possible by means of an internal generator of names which, starting from the name of a generic object, provides any individual of that class with a different name. This feature covers the part of the naming system of NL which uses the same name for individuals and prototypes. This does not cover the use of proper names which has been taken in JARGON (Woods, 1979) as the only means for naming individuals, thus oversimplifying the real system used by NL (Mark, 1981).

Objects can be accessed without the use of names, but by means of functions or combinations of functions in order to perform complex tests on the nature of objects. This means referring to objects by testing properties or configurations.

C. KL-Conc Functions

KL-Conc has functions for creating, testing and retrieving objects. This is the list of the functions so far designed:

GEN
NEWIND
JUSTONE
ANYONE
SOME
ALL
LAST
ADD_PROPERTY
ADD_CONFIGURATION_OF_PROPERTIES
TEST_PROPERTY
TEST_CONFIGURATION_OF_PROPERTIES

The semantics of some KL-Conc functions may now be described in order to clarify how they realize our linguistic assumptions. The semantics is given in terms of operations on SI-Nets.

As far as generic knowledge is concerned, the function:

<GEN arbitrary_name>

returns the generic concept named by arbitrary_name. If the concept does not exist in the LTM a new generic concept is created. The new concept is then returned. This function works both as a predicate and as a creating function. It is worth noticing that in KL-Magma there are two distinct functions, one for the predicate (<Generic_Concept_P anything>), and the other for creating (<Create_Concept name type_of_concept>).

The function

<NEWIND arbitrary_name>

creates a new individual concept and establishes it as an individuator of the generic concept named by arbitrary_name; if the generic concept does not exist in the LTM it is created. An internal generator provides the newly created individual concept with a name. This function corresponds to the following set of KL-Magma functions:

(Create_Concept X1 individual)

((Not (Generic_Concept_P X) (Create_Concept X generic))

(Establish_as_Individuator X1 X)

This is one of the most "declarative" functions since it creates a new individual concept without searching in the LTM. In other words, the user must be conscious that the new object is added to the LTM and it is different from all the other objects. A more psychological oriented behaviour would require to test in advance the nature of the new object in order to decide whether the object is similar to or coincides with an individual object already inserted into the LTM. The same problem has been overcome in KRYPTON by means of the switch TELL/ASK (Brachman et al., 1983).

The function

<JUSTONE arbitrary_name>

verifies whether there exists a unique individual either named by arbitrary_name or defined by tests or combinations of tests according to KL-Conc syntax. In other words, this means verifying if the object is unique as to its name, or as to one of its properties etc. The KL-Conc expressions for the two meanings are, respectively:

(JUSTONE table)

(JUSTONE (TEST_PROPERTY table red))

This function has a complex behaviour, since, intuitively, it must verify the uniqueness of an object and must return: i) the individual if unique; ii) the list of individuals if more than one satisfies the conditions given by assertions; iii) NIL if no invividual exists satisfying the conditions (Carnap, 1947). The three answers have different meanings, since they imply different operations to be triggered on the memory spaces or, at any rate, they have different effects on the behaviour of functions where JUSTONE can be nested.

The function:

<TEST_CONFIGURATION_OF_PROPERTIES arbitrary_name1 arbitrary_name2>

verifies whether arbitrary_name2 exists in the horizontal chain of roles starting from arbitrary_name1 (see Figure 4)

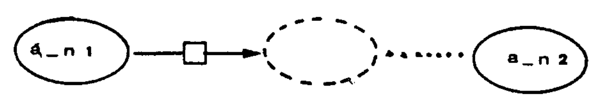

Figure 4

BY the function:

<ADD_PROPERTY arbitrary_name1 arbitrary_name2)

we intend to add roles to concepts so that the user needs not have any specific knowledge about the distinction between generic and instance roles or, seen from a different viewpoint, between properties of prototypes and properties of individuals. Taking NL as the reference point, we think that the above mentioned distinction is peculiar only to certain linguistic elements; in the case of operations on properties, no distinction is made; it is the conceptual operations governing the operations on properties that control the correct application of the adding or testing properties. Consequently, the function ADD_PROPERTY must be designed in order to make it possible to trigger the correct procedures depending on the type of objects which it is applied to. For this purpose, we intend to use a metarepresentation of KL-Magma (Cappelli et al., 1983) which, on detecting the type of object, automatically apply the appropriate procedures. This implies a system which creates or tests knowledge structures interpreting its own syntax.

Let's now briefly describe two possible behaviours of this function.

When applied to individual concepts, this creates a new instance role establishing it as a satisfier of a higher generic role of the generic concept ancestor of the individual concept. If a possible generic role does not exist it is created without inserting any V/R in the generic role, since it could be a more general concept than the generic concept ancestor of the value of the newly created instance role. The structures created by this function are shown in figure 5 by dotted lines.

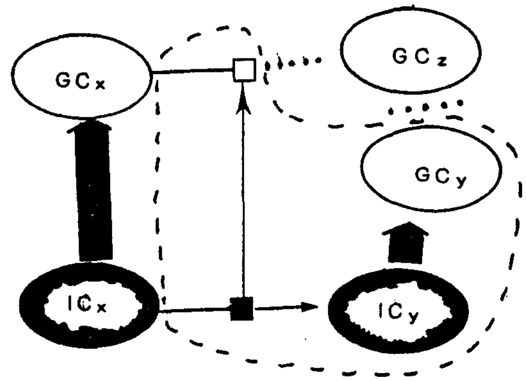

Figure 5

When applied to generic concepts, the function adds a new generic role, trying to link it with a higher generic role. If no generic role is found, a higher generic role is created without providing it with any information other than the one inferred from the structure of the newly created subrole.

VI CONCLUSIONS

Some conclusions may now be drawn both from a linguistic and a knowledge representation viewpoint.

From a linguistic viewpoint some relevant facts must be pointed out.

First of all, the level of integration reached by the construction of a uniform language, can bring out more clearly the nature of many phenomena of NL, since it is possible to put together many processes which cooperatively contribute to the realization of a single phenomenon. This means looking at the complexity of NL with the aid of a powerful symbolic instrument, capable of handling contemporaneously several aspects of that complexity, thus reaching a higher degree of adequacy. In designing KL-Conc, we aim to create a framework which can extend the possibility of investigating and representing these phenomena.

The functions described in this article represent only a subset of the operations which can be embodied in the language. In this sense, the number of KL-Conc functions is likely to be increased in order to cover new processes.

So far, we have designed the functions for those operations which exhibit the same behaviour whatever domain they are applied to, since they represent the "deep" behaviour of syntactic elements. It is to be emphasized that we have tried to reduce to the form of functions of a language, all the operations of NL which are domain-independent and which represent aspects of the abstract syntactic ability of structuring knowledge facts (Cappelli et al., 1983; Cappelli and Moretti, 1983)

Using KL-Conc it is possible to investigate how linguistic elements can be described in terms of conceptual operations. This is a further step towards the linguistic level. On reaching this level, the task will be to discover how the conceptual operations are embodied in linguistic forms.

The previously mentioned Italian articles may be described as follows:

```
(Definite Article lambda (x)
                  (or (GEN x)
                      (JUSTONE x)))


(Indefinite Article lambda (x)
                  (or (GEN x)
                      (ANYONE x)))
```

From a knowledge representation viewpoint KL-Conc would seem to be a means for interacting with SI-Nets in an intuitive way. The user is not required to have a specific knowledge of SI-Nets formalism; he only needs to know a set of functions to be applied to objects.

In this sense KL-Conc assumes a more natural aspect, thus overcoming the constraint of a structure-oriented language such as KL-Magma. This feature has been obtained by handling SI-Nets in a more compact way. KL-Conc provides the user with a set of functions which are not isomorphic to single epistemological objects but which handle pieces of network starting from discontinuous information.

This weakness, peculiar to NL, is made possible in KL-Conc by assuming the epistemological level as a reference schema, instead of a reductionist formalism. This means introducing mechanisms for relaxing the rules of KL-Magma. In this way KL-Conc can be seen as a "constructive" system (in the sense of Korner 1970) which manipulates its "factual" system (KL-Magma) in an intuitionistic way.

Finally, KL-Conc suggests a different way of exploiting spreading activation mechanisms (Quillian, 1968) using several symbolic systems organized by the WI instead of considering them as algorithmic devices internal to SI-Nets (Woods, 1981):

VII REFERENCES

Asirelli P., C. Lami, C. Montangero, G. Pacini, M. Simi, F. Turini, "MAGMA-Lisp Reference Manual", N.T. C75-13, I.E.I CNR, Pisa , 1975.

Brachman R., E. Ciccarelli, N. Greenfeld, M. Yonke, "KLONE Reference Manual", Report n. 3848, Bolt Beranek and Newman Inc., Cambridge (Mass.), July 1978.

Brachman R. J., "On the Epistemological Status of Semantic Networks", in N. Findler (ed.), Associative Networks: Representation and Use of Knowledge by Computers, New York: Academic Press, 1979.

Brachman R. J. and H. Levesque, "Assertions in KL-ONE", in Schmolze J. and Brachman R. (Eds.), Proceedings of the 1981 KL-ONE Workshop, Report n.4842, Bolt Beranek and Newman Inc.,Cambridge (Mass.), June 1982.

Brachman R. J., R. E. Fikes, H. J. Levesque, "KRYPTON: A Functional Approach to Knowledge Representation", Fairchild Technical Report n.639, FLAIR Technical Report n.16, Palo Alto (Ca.), May 1983.

Cappelli A. and L. Moretti, Introduzione al KL-ONE, ILC-NLP-1982-1, I.L.C. CNR, Pisa, 1982.

Cappelli A. , L. Moretti, C. Vinchesi, "KL-Conc: a Language for Interacting with SI-Nets", in Proceedings of the 8thIJCAI, Karlsruhe, Germany, August, 1983 .

Cappelli A., L. Moretti, "Descrizione di alcuni fenomeni dell' Italiano con un Linguaggio di Rappresentazione della Conoscenza", in Atti del XVII Congresso Internazionale di Studi della Societa' di Linguistica Italiana, Urbino, Italia, September 1983.

Carnap R., "Meaning and Necessity", Chicago: The University of Chicago Press, 1947.

Korner S., Categorial Frameworks, Oxford: Basil Blackwell, 1970.

Mark W. S., "Individuality in KL-ONE ", in Schmolze J. G. and R. J. Brachman (eds.), Proceedings of the 1981 KL-ONE Workshop, Report n. 4842, Bolt Beranek and Newman Inc., Cambridge (Mass.), June 1982.

Porta O. and C. Vinchesi, "Un Sistema per la Rappresentazione della Conoscenza: Aspetti Concettuali e di Implementazione", unpublished Thesis , I.S.I, Pisa 1981.

Quillian M. R., "Semantic Memory", in M. Minsky (ed.), Semantic Information Processing, Cambridge (Mass.): MIT Press, 1968.

Schmolze J. G. and R. J. Brachman, "Proceedings of the 1981 KL-ONE Workshop", Report n. 4842, Bolt Beranek and Newman Inc., Cambridge (Mass.), June 1982.

Schmolze J. G. and T. A. Lipkis, "Classification in the KL-ONE Knowledge Representation System", in Proceedings of the 8thIJCAI, Karlsruhe, West Germany, August, 1983.

Webber B. L., "A Formal Approach to Discourse Anaphora", 1978.

Webber B. L., "Discourse Model Synthesis", in A. Joshi, B. L. Webber and I. Sag (eds.), Elements of Discourse Understanding, Cambridge (Mass.). 1981.

Woods W. A., "Research in Natural Language Understanding", Report n. 4274, Bolt Beranek and Newman Inc., Cambridge (Mass.), 1979.

Woods W. A., "Abstract Algorithms and Architectures", in Woods W. A. (ed.), Research in Knowledge Representation for Natural Language Understanding, Report n. 4785, Bolt Beranek and Newman Inc., Cambridge (Mass.), November 1981.