Nature and Structure of Terminologic Knowledge: The DRL Approach*

Nicola Guarino
LADSEB-CNR
Corso Stati Uniti 4
I-35020 Padova, Italy
guarino@pdcnr1.pd.cnr.it
6 May 1989

1 A different approach to hybrid systems

DRL (Declarative Representation Language) is the result of a revisitation of the ideas of hybrid systems like KRYPTON within a logic programming framework. A common characteristic of these systems is the distinction between terminological and (so called) assertional knowledge, but the criteria of such a distinction and its impact on the diversification of the formalisms used still constitute an open research issue. In particular, arguments based on pure epistemological distinctions are interlaced with practical considerations regarding expressivity limitations of the formalisms adopted.

Brachman and colleagues started from frames, coming to the necessity of "essential hybrid" systems [6] from an analysis of their assertional limitations [5]. We took the opposite way, starting from an assertional language like Prolog and trying to overcome the "lexical ghost" mentioned in [6]. The idea of DRL is to improve the expressive adequacy of Prolog theory with a suitable

---

*Extended Abstract (the full paper will appear in the Proc. of 1st Conf. of the Italian Association for AI, Trento, November 1989).
terminological component, expressed as a (separate) Prolog theory itself, the terminological Knowledge Base (TKB). In this way a common basic formalism is adopted for the two kinds of knowledge, yet suitably differentiated in order to capture the intrinsic differences.

LOGIN [1] is an example of this approach, which falls under Frisch’s general substitutional framework [15]: we can distinguish a logic module which is simply a Prolog theory where variables can be restricted to (complex) sorts (and is therefore a many-sorted theory), and a sortal module where the sortal structure of the domain is defined. Inferences can be drawn from a many-sorted theory via many-sorted unification, which has been proved useful to reason efficiently with problems involving taxonomic reasoning. It uses knowledge about the sortal structure to restrict the standard unification algorithm, and provides in this way a natural link between the two modules. The distinction between these two modules resembles in many ways KRYPTON’s ABox/TBox distinction, but presents a number of important differences, mainly due to the presence of individuals in the sortal module. In DRL (developed independently from LOGIN) the implications of the sorted deduction paradigm on the so-called epistemological distinctions present in hybrid knowledge representation systems are explored in detail, coming up with an analysis of the nature and the structure of terminological knowledge.

2 The nature of terminological knowledge

If we choose to use logic to represent knowledge, we have to assume that constants, functions and variables stand for (or range over) elements of a set called the domain of discourse, whose structure is independent of the logic theory itself (the object theory). If we are able to represent in some way this structure, we can rigorously regard the resulting knowledge as terminological, since constants, functions and variables are exactly the terms used by our logic theory.

Structuring a domain means identifying (relevant) subsets, and relevant relationships between them. For instance, if our domain regards people, it may be relevant to split it into men and women and represent the fact that, say, the term John refers to a man, while the term Mary to a woman. In this way we may be able to conclude that Mary is not a man by performing a pure terminological reasoning, i.e. by examining the domain structure only.
This implies that:

1. At least part of terminological knowledge has to have an assertional import, due to the presence of constant terms (individuals) in the TKB.

2. Analytical subsumption isn’t the only relevant relationship for terminological reasoning, since extensional relationships between concepts and their instances are also necessary. We adopt the term taxonomic (or terminological) relationship to refer both to subsumption and to extensional relationships.

Extensional relationships influence the structure of terminological knowledge, as it is discussed below. Regarding the assertional interpretation, we show in the paper that it does not necessarily lead to a serious loss of expressive power, but only to the absence of an epistemologic distinction between analytic and synthetic knowledge. This kind of distinction (if really necessary) turns out to be independent of the definition of terminological knowledge introduced above.

The choice to ascribe an assertional import to the terminological knowledge allows us to simplify our object theory, since we can get rid of the clauses which are a direct consequence of the domain structure. Moreover, it is possible to gain expressivity, by restricting the range of variables of the remaining clauses to suitable subsets of the domain (the increase of expressivity lies in the fact that it is possible to drop the assumption that sorts have to be non-empty). The resulting theory is then many-sorted. We say that it expresses relational knowledge, i.e. knowledge about arbitrary relationships among terms, while terminological knowledge is about taxonomic relationships between (many-sorted) terms.

3 Structuring terminological knowledge

In order to identify the structural relationships between terms we have first to decide what basic kinds of terms we want to have, i.e. what are the basic objects of predication in the RKB. In other words, we are committed to some ontological assumptions about the intrinsic nature of the domain of discourse. In the basic version of DRL, terms may denote:
1. Individuals (extensional entities), represented as Prolog atoms (e.g.: john).

2. Classes (extensional entities), intended as generic collections of individuals, and represented as Prolog lists (e.g.: [john, bob]).

3. Concepts (intensional entities, intended as names of relevant subsets of the domain), which may be elementary concepts, like "person", or derived concepts, such as boolean concepts, like "animal and thinking-being", or attribute concepts, like "phone of john".

4. Generic instances of a class or a concept, represented as qualified variables (e.g.: person::X, (phone of john):: , [a1,a2,a3]:--).

The various taxonomic relationships between terms are expressed in the TKB using two relations: the qualification relation and the extensional equivalence relation.

The fundamental taxonomic relation is the qualification relation. In its basic form, it holds between a concept and another term if the latter denotes a (generic or definite) instance of the former. The symbol used for the qualification predicate is the same used for constructing qualified terms: in this way the Prolog statement person::john expresses an extensional relationship, saying that John is an instance of the concept person, while person::student:: expresses an analytic subsumption, saying that each (generic) instance of the concept student is also an instance of the concept person. The basic qualification relation is further extended by allowing a class to appear anywhere within a qualification relation; for instance the assertion [a,b]:--a means that a is a member of the class [a,b], while [a,b]:--c:: means that each instance of c may be either a or b, making possible in this way a simple form of disjunction. On the other side, the statement c::[a,b] means that the class [a,b] is an instance of the concept c, while c::[a,b]:-- means that each of its members is an instance of c.

The above statements can be easily mapped into a pseudonatural syntax, where the qualification predicate is expressed as a copula: "john is a person", "each student is a person". We can see then that the qualification relation is a special kind of (intensional) predication relation, characterized by the fact that the object of predication is restricted to be an ontologically
predefined concept. Current work on the formal semantics of DRL is focused on intensional logics of predication, which have recently been object of philosophical studies [3,7], and seem to exhibit a very high expressive power and cognitive adequacy while maintaining (most of) the desirable properties of first-order logics. OMEGA [2] is also a language based on the predication relation, which differs however from DRL in various ways: i) it has a strictly extensional semantics; ii) it is based on structured descriptions; iii) it only represents taxonomic knowledge; iv) it has better theorem-proving capabilities.

The second taxonomic relation is the extensional equivalence relation, which holds between a concept and the class which denotes its extension, or between two concepts which share the same extension. We write for instance

\[
\text{color} := [\text{red, green, blue}]
\]

if the only possible colors are red, green and blue, and

\[
\text{man and woman} := []
\]

if men and woman are disjoint. As it is discussed in [10], the introduction of classes together with the extensional equivalence relation allows to distinguish between complete and incomplete knowledge.

4 Attributes as concepts: a uniform and coherent system

The standard way of representing attributes in first order logic [12] is to use 2-places predicates, like phone(john, 845251); we write instead (assuming phone a single-valued attribute of john) "phone of john := [845251]", using the of operator to define a derived concept which we call attribute concept, whose extension consists of the singleton [845251]. The representation of attributes as derived concepts creates a uniform and coherent system, akin to those cited in [13]. The idea of representing values as instances of attribute concepts, discussed in more detail in [10], allows us to speak about values without introducing equality, and to express different kinds of knowledge about the attribute-value relationship in a natural and rigorous way:

5
Complete and incomplete knowledge:

phone of john := [845251].
phone of john := [845251,845111].
phone of john := [].
(phone of john)::845251.
[845250,845251]::(phone of john)::.
phone of john := phone of mary.

Value restrictions, number restrictions and constraints:

man::(spouse of woman:: )::.
woman::(spouse of man:: )::.
number of child of lucy := [2].
volume of fluid::F := volume of solvent of fluid::F.

5 Interacting with the knowledge base.

Let us briefly describe now the formalism used to interact with a DRL knowledge base at the knowledge level. Within a logic programming approach the introduction of a predicate like ask should be avoided, since it has a strictly functional meaning; consistently with a rigid declarative approach (as the system’s acronym suggests) it seems more appropriate to express queries in terms of propositions regarding system’s competence, i.e. its ability to be conscious of its explicit beliefs and to derive what is implicit in them. For this purpose we introduce three distinct predicates [9], which handle homogeneously both the TKB and the RKB: k for knowing that, know for knowing what, and b for explicit belief. The predicate k(P) is true if the system knows that proposition P is true, while know(X) is true if the system knows the extension of X, i.e. the class equivalent to X if X is a term, and the truth value of X if X is a proposition. In order to introduce simple forms of quantification within TKB queries we introduce the special terms "some" and "all", which allow to ask queries like k(a::some) or k(a::b::all), with the obvious meaning. Finally, b(P) will be true if P is a fact explicitly held in the TKB or in the RKB. Notice that these predicates are used only for asking about the knowledge of a single agent (the system), not for reasoning about such knowledge. The use of Prolog allows the argument of k to be partially instantiated, making retrieval operations possible.
6 Necessary and sufficient conditions

TKB unit clauses presented so far specify necessary conditions. For instance person::parent:: states that each parent is necessarily a person. If clauses in the TKB have a body, it is possible to specify sufficient conditions in an independent way:

parent::person::X:- (child of X)::some.

In this case, if the existence of at least one child is also a necessary condition for being a parent, it has to be explicitly written as a separate statement, such as (child of parent:: )::some (some is automatically converted into a unique Skolem constant if it doesn’t appear in a query). If we introduce now the conditions for being a grandparent, we are able to classify each grandparent as a parent even if this knowledge has not been explicitly declared:

grandparent::person::X:- parent::(child of X)::some.
person::grandparent:: .
parent::(child of grandparent:: )::some.

person::parent:: .
parent::person::X:- (child of X)::some.

:- k(X::grandparent::all).
X=parent;
X=person;
NO

The algorithm for classification is similar to the "standard" one: person is the direct ancestor of grandparent, and its "primitive region" is explored in order to get a defined concept, that is parent, which may subsume grandparent. This subsumption can be computed quite efficiently, since we have only to test the sufficient condition (child of grandparent::all)::some, which is true due to the third statement.

In the paper we show how this distinction between necessary and sufficient conditions involves an assertional interpretation of frames weaker than the one proposed in [12], which has the advantage of no loss of expressive power if an assertional interpretation is given to the TKB. Expressive power is actually increased, since only sufficient and multiple sufficient (as well as only
necessary) conditions may be specified even for primitive concepts. Moreover, the distinction should decrease the computational cost of subsumption, since - instead of comparing two complex concept definitions, only the sufficient conditions of the subsumer have to be taken into account. We conjecture therefore that the complexity of subsumption should be \( O(S^{**K}) \), where \( S \) is the number of sufficient conditions (i.e., a percentage of the complexity of a KL-ONE definition).

7 Conclusions: ten years after, the epistemological level disappeared?

In conclusion, the most significant difference between DRL and frame-based description languages is perhaps the impossibility to express generic descriptions, like "a person each of whose male friends is a doctor". We feel however that - according to the classification presented in [4] - such a kind of descriptions pertain to the linguistic level, not to the epistemological level, which was the original target of KL-ONE. The reason is simply that they look like the arbitrary concepts mentioned in the last row of Brachman's table, and they are indeed referenced as independent elements of the discourse only within natural language. Their use as primitives for the epistemological level seems simply inadequate, since a finer granularity is needed in order to capture the nature of the knowledge expressed. We may conclude that either such a level is no more necessary, or that its expressive power had to be traded off with tractability and formal appeal. In DRL we have tried to keep firm in mind the expressivity requirements needed at the epistemological level. We have reached in this way the conclusion that if attributes are the primary device used to characterize the structure of knowledge, they need to be represented in a coherent and uniform way, with the relationship between them and the piece of knowledge they characterize (i.e. the necessary/sufficient conditions) clearly defined. Our preliminary results seem to show that we haven't to pay the price of classification intractability for this augmented expressive power; we strongly agree however with the arguments given by Doyle and Patil in [16], where it is argued that, even in the case of classification intractability, expressiveness should be a primary concern for representational services.

The price we we sure have to pay for this choice is the inability to ex-
press generic descriptions, together with the loss of the modularity typical of frames. It is however only a problem of abstraction level, since a modular frame-based description language (at the conceptual or linguistic level) may be easily built on top of a uniform semantic network, at the epistemological level.

Last but not least, DRL is a running system, fully implemented in LPA MacProlog. An algorithm very similar to that presented in [14] is used to implement taxonomic reasoning.

8 Cited references

[14] Agrawal, R., Borgida, A., Jagadish, H. V., Efficient management of transitive relationships in large data and knowledge bases, SIGMOD 89.
