A Meaning Representation Language for Co-operative Dialogue

G. J. Doe, G.A. Ringland, M.D. Wilson
SERC Rutherford Appleton Laboratory, Chilton, Didcot, Oxon., OX11 0QX, U.K.
gjd@inf.rl.ac.uk

Abstract: A meaning representation language is described which includes a typed first-order logic with relativised quantification, using an ontology with reified events and actions. This has been developed to support the requirements of multimodal co-operative dialogue, including ambiguity, temporal and aspectual reference in natural language, reference to graphical and dialogue objects of varying complexity, explanation, and the differing approaches needed to interpret assertions, questions and commands.

With the rise of intelligent front ends to databases and knowledge bases not only must the representation of the knowledge in the application be considered, but also the representation in the front end. It has become conventional (e.g. Gazdar and Mellish, 1989; Warren and Pereira, 1982) for natural language front ends to databases to translate natural language queries into a meaning representation language (MRL) which is then used to access the database. Such an MRL must be sufficiently expressive to capture the content of the natural language utterances, whilst also being computationally tractable in evaluation as an access mechanism to the database or knowledge base (Williams and Lambert, 1988). The point at issue in this case is not inferencing over expressions of the language, but evaluating them against a separate knowledge base.

When the underlying application is not a database but an active knowledge base, and the user's task is not that of data retrieval but the broader one of design, the demands on the expressiveness of the MRL are increased, for this task requires not only retrieval, but also updating. Further demands are imposed when the mode of interaction is not only a constrained natural language interface using one language, but one which includes several languages as well as graphical display, direct manipulation input. In the MMI[2] project, a demonstrator system has been built which designs computer networks for buildings, and interacts with users through English, French or Spanish natural languages,

* The work reported in this paper was partly funded by the Commission of the European Communities under the ESPRIT programme through project grant P2474 (MMI[2]). The partners in the project are BIM (Belgium), Intelligent Software Solutions S.A. (Spain), SERC/Rutherford Appleton Laboratory (UK), ADR Centre de Recherche en Informatique applique aux Sciences Sociales (France), Ecole des Mines de Saint-Etienne (France), Institut National de Recherche en Informatique et en Automatique (France), University of Leeds (UK).

Command Language, Graphics, Direct Manipulation, Design Gestures and non-verbal audio (Binot et al., 1990). The underlying application uses a frame-based object-oriented representation of building and network objects, and inferences on this to design a computer network to given requirements, and to analyse networks for several different domain criteria (e.g. the cost of the network, or the relationships between client and server machines).

MMI² uses an MRL to represent the meaning of user utterances, but also to represent the output from the system which can be presented in natural language, graphics, audio or a combination of these. This range of modes provided reflects the range of methods used by human designers in discussing the domain. Furthermore, the user does not make an exclusive choice between modes for any interaction, but may use two or more modes, for instance a natural language coupled with manipulation of graphical diagrams of the current design. This range of methods of creating the MRL and of using it place further demands upon it of consistency and clarity as a representation of meaning. The MRL in MMI² is called the Common Meaning Representation (CMR) since it is common to all modes of interaction with the application KBS for input and output rather than merely between one natural language and a database for expressing queries.

Clearly the CMR will represent certain features of meaning which are typical of some particular mode or modes; this is especially true of NL since this is a source of considerable semantic complexity by comparison with, for instance, the less rich command language. Therefore it is essential that it be adaptable enough to represent more or less complex expressions from the various modes equally appropriately and effectively. As such its development in response to the needs of the different modes has had a major role in facilitating the integration of the various modes and other modules in the architecture of the interface.

The approach to representing meaning in CMR is largely logical. The propositional content is expressed in a form based on standard typed first-order logic with general reification. However, not all relevant aspects of communication acts can be satisfactorily captured in this form, so the CMR expression includes, as well as a logical formula, a marking of illocutionary force and other annotations as necessary. It is passed between modules in a larger data structure which also incorporates information regarding time, source mode, processing status, etc. The time information, for instance, is an important factor in determining deictic reference. To begin, we concentrate here on the logical formula at the heart of the packet.

A characteristic feature of the logic used is that, wherever possible, events and situations within the formula are reified. That is to say that we treat them as objects in the universe of discourse, and quantify over them. The decision to use general reification was taken to ensure a degree of uniformity which would be difficult to achieve if reification were used only in certain cases. This results in conceptual terms in an extremely rich variety of objects and types - what has been described as a promiscuous ontology (Hobbs, 1985). Thus quantifiers range over sets of objects which include the situations which hold true when a given predicate is true of a given set of arguments.

It should be noted, however, that quantifying over a given situation does not imply that it holds true in the world being described. The quantifiers range over everything that can be described, regardless of its existence or otherwise. There is a distinction between objects which are present in the formal domain and those which are held to exist in the real world. This is best understood as a distinction drawn in this formalism between the assertional and the propositional content of an utterance. Real-world existence is represented in Hobbs' formalism by the predicate exist/1. The relation of this predicate to reification is defined in the following axiom schema; for any n-ary predicate p,

(Vx1,...,xn) p(x1,...,xn) <=> (Ee)exist(e) & p'(e,x1,...,xn).

where e is the condition of p being true of the arguments x1,...,xn. Hence exist(e) is the assertion that the situation e holds; p'(e,x1,...,xn) gives the proposition defining what e is. Note that this precludes reifying every predicate, since an infinite recursion on exist/1 would be inevitable.The use of reification in the CMR is based on this definition, and to that extent draws on Hobbs' work for one of its characteristic features, but offers a good deal more also.

The usefulness of this process of reification can be seen in a simple example. Suppose we have the following sentence of English:

“A man reads a book.”

This can (arguably) be quite satisfactorily represented in conventional first-order predicate logic.

(Ex,y) man(x) & book(y) & read(x,y)

However, problems arise when more complicated sentences are used, as they frequently are in NL. The following sentence poses substantial problems:

“A man tries to write a book quickly.”

One of the problems is that we wish to refer to the writing of the book as an argument to the trying; a further complication is that we wish to do so without committing ourselves to the implication that this writing takes place, nor indeed that the book actually exists. Reification, without the real-world existence assumption, makes this quite straightforward:

(Ew,x,y,z) exist(w) & exist(y) & man(w) & book(x) & try'(y,w,z)
& write'(z,w,x) & quick(z)

Here w is a man and x a book; z is w's putative writing of x, while y is w's attempt to bring about z. Only w and y - the man and the attempt - are held to exist. No commitment is made to the existence or otherwise of x and z. It is possible to use axioms which could state, for example, that any object in the domain of discourse which is the subject of a trying can be inferred to exist.This is referred to as the 'transparency' of the 'try' predicate in a given argument, and would mean here that the existence of w need not be explicitly stated. The advantages offered by reification are clear in the ability it offers to quantify over, and predicate attributes and relations of, situations.

One modification of Hobbs' definition has been the decision to restrict predicates, other than those expressing the assertional content and the types of objects, to binarity. This is to be achieved by associating what would otherwise be the arguments of higher-arity predicates to the reified predicate objects by means of explicit argument-label predicates. Given an appropriate mapping between these predicates and the argument ordering of n-ary predicates, the two are conceptually equivalent. Binarity has been chosen on the grounds that it makes it possible to equate argument predicates with case roles, while improving the efficiency of the formal evaluation of the formulae. Thus the form

p(x,y,z)

of standard first-order predicate logic without reification, which when reified would be represented by

exist(e) & p'(e,x,y,z)

becomes in this formalism

exist(e) & p''(e) & arg1(e,x) & arg2(e,y) & arg3(e,z).

The actual mapping between argument predicates and cases used in any real situation would depend on the predicate involved.

Then by representing the quantifiers explicitly we have;

(Ee : p'')
(Ex,y,z)
(exist(e) & arg1(e,x) & arg2(e,y) & arg3(e,z)).

Or to give a specific example,

(Ee:shooting)
(Ex:man)

(Ey:dog)
(Ez:rifle)
(exist(e) & arg1(e,x) & arg2(e,y) & arg3(e,z))

would represent the sentence

"A man shoots a dog with a rifle."

Here, arg1 can be viewed as the agent role for the predicate 'shooting', arg2 the object/patient and arg3 the instrument. Thus the predicates in the main body of the formula may be limited to the existence predicate and the argument labels.

However, in order to fully express the meaning of some of the more complex input that can be foreseen, especially from the NL modes, some extension of this framework is required. In particular, those areas of meaning expressed by the grammatical features of tense and aspect are not covered by the formalism described thus far. However, this framework can be extended to represent descriptions of time and aspect in a quite straight-forward manner without departing from first-order logic.

Since the situation is already represented as an object, it is a simple matter to define not one but three predicates carrying assertional force, representing past, present and future existence of the object of which they are predicated. Present/1 is equivalent to exist/1, while the others allow the representation of the notion of time, which Hobbs hints at without exploring it in any detail. Otherwise these existential predicates are treated exactly like any other. Most importantly, they too may be reified; it is quite possible to quantify over the present existence or the pastness of a situation, and thus to attribute properties or relationships to them. Hence utterances can be easily represented which describe the existence of a given situation as surprising or likely, for example. Note that by the definition of reification

(Vx1) past(x1) <=> (Ex2) present(x2) & is_past(x2, x1)

where is_past/2 is the reified version of past/1.

Where it is possible to represent time, it is necessary, in order properly to carry out the reasoning involved with tensed utterances, to represent also the meaning expressed in NL by aspect. One facet of meaning which needs to be made explicit in the CMR is the distinction between different ways of describing and perceiving situations. We have divided situations into four sorts, namely state, process, event and habitual. Of these, states and events are primitive, while the others are composites made up of these two primitives. However, it is not always desirable to break these composites down as the details of the breakdown are not needed and often not available. Note that a situation is not essentially one of these four kinds; the classification depends on the description of the situation, rather than representing any philosophical claims about its nature. Thus the same situation might at different times be treated as two or more different kinds, depending on the perspective of the describer.

To give some detail of what is meant by these descriptions; a state is a situation which is perceived as existing rather than occurring. It is taken to be continuous and unchanging throughout its duration. An event, on the other hand, is an instantaneous happening, conceived of as being without internal temporal structure. A process is then defined as a composite structure made up of an ordered sequence of events and states, treated as a whole. Like a state it is durative, but it is not homogeneous. A habitual is a series of repeated similar events or processes which together can be treated as a unitary durative situation. An example would be the statement “I sell insurance”; during the period in which this is true, for the most part the related statement “I am selling insurance” would not be true. The latter describes a process or event, the former a situation of the type we classify as 'habitual'. These four kinds of situation are represented by using the predicates state/1, event/1, process/1 and habitual/1, with a situation - some object created by reifying a predicate - as the argument. These unary predicates can then themselves be reified to retain the uniformity of representation desired.

A further distinction, also important for determining what inferences are valid, can be drawn between accomplished and unaccomplished situations. These are often described in terms of progressivity and perfectivity, but they are basically linguistic concepts, so the more neutral terms are preferred, given the all-pervasive influence of the requirements of commonality of representation for all modes. Similarly there is a distinction between processes with a set terminal point and those without, which must be adequately captured in this representation. This is covered by the predicates culminative/1 and nonculminative/1. The influence of this distinction may be seen from the fact that it may affect what objects are inferred to exist in the world, as with such culminative processes as drawing a circle, or making some object.

We therefore have eleven predicates, divided into four groups, to represent the meanings of various kinds of situation descriptions.

past(x) - present(x) - future(x)
state(x) - event(x) - process(x) - habitual(x)
accomplished(x) - unaccomplished(x)
culminative(x) - nonculminative(x)

Predicates in the same group cannot, of course, co-occur applying to the same situation. The groups are not entirely independent; for example, states are generally unaccomplished and events accomplished, and the notion of culmination applies to processes only. However, accomplishment and culmination are independent of one another, and any of four pairings are possible. These predicates are a fundamental part of the CMR and will remain when any alterations are made to more domain-specific predicates.

In fact they give rather more expressive power than is required for the current application, since the underlying knowledge base does not currently have the capability to represent temporal considerations in the domain. However, this degree of redundancy allows for the use of the interface in other domains. It also provides a basis for ongoing work in providing this functionality. To some extent this may involve representing time in some rudimentary sense in the network design domain - storing previous phases of a design and treating them as states of the world at ordered time co-ordinates. Of perhaps more interest is the possibility of dynamically modelling the structure of the man-machine dialogue and referring to previous dialogue events, which the combination of general reification and enunciation predicates will allow us to represent and reason over as fully-fledged objects in our ontology. The power of this capability can be seen in the possibility of referring back to previous utterances in requesting and creating explanations of the system’s behaviour and of the characteristics of the domain. It is also worth noting that though the use of temporal concepts in interpreting user input may be limited, the CMR language is also used for representing the meaning of system output, prior to natural language generation or graphical visualisation, and here there is further scope for using the enunciation predicates described.

The very rich ontology which is made possible by reification also has advantages in supporting the expressiveness required for flexible multimodal interaction (Wilson and Falzon, 1991). An example of this lies in the ability of the MMI^2 system to treat text as graphical rather than linguistic objects. Thus the user may select a string of text displayed on the screen and it will be represented in the CMR as an object in its own right. This makes it relatively straightforward to handle the use of designer shorthand gestures to capitalise or underline text, for example. An area for future investigation is the possibility of combining the representation of text and dialogue objects, enabling the user to select text on the screen and refer to its meaning or dialogue function as well as its surface form.

As well as text strings, more complex graphical structures may also be represented as objects in the CMR language. An example which demonstrates the utility of this capability is the representation of the sentence:

“Show a bar graph of the costs of machines on the network.”

Here the predicate “display” is represented as having as one of its arguments an object of type “bar_graph”. This object is in itself a complex interface object, having as its content a set of data - specifically, a set of machine-cost pairs, and this content is itself represented as an object in the CMR. Thus we take a higher-order construct, but are spared the potential problems of the tractability of interpreting it because interpretation in the graphical domain is carried out by specialised procedures.

It is worth mentioning here the fact that the CMR also allows the use of a limited number of second-order predicates in the application domain itself which are interpreted in a general non-ad-hoc way. These are predicates such as "count" which take set terms as arguments. Such predicates are vital to give the CMR language adequate expressivity, since without them there would be no way of expressing questions such as:

"How many machines are there on the network?"

if the domain of discourse is limited to individuals as in first order-logic. However, we have endeavoured to make the extension to first-order predicate logic as modest as possible, by the constraint that sets may only be referred to by means of a set term, not by constants or variables. This effectively prevents quantification over sets, and avoids the problems associated with more radical extensions of the logic. Thus the evaluation of such questions, or statements using similar constructs, is relatively unproblematic.

The purpose of the CMR was to offer the expressivity required to support multimodal, co-operative dialogue, using as an example for evaluation the test domain of computer network design. After studying human-human dialogues simulating human-computer interaction for this task, a number of specific requirements were identified, including the following particularly challenging areas: ambiguity, temporal and aspectual reference in natural language, reference to graphical and dialogue objects of varying complexity, explanation, and the differing approaches needed to interpret assertions, questions and commands input to the interface in natural language modes. The definition of the CMR as a complex data structure incorporating a logical formula has made such expressivity possible whilst facilitating efficient interpretation of input relative to the underlying knowledge base. For example, the syntax of a CMR expression includes a 'slot' for the marking of utterance type. This slot is filled by the input mode by reference to the surface form of the NL input, and allows the interface's CMR interpretation module to adopt differing strategies according to whether the utterance is an assertion requiring updating of the knowledge base, a yes-no query, or a wh-query. Furthermore the expression may include more than one logical formula, which makes it possible to represent all possible meanings of a given ambiguous utterance as alternatives within a single structure. These features of the CMR, along with those described in the body of this paper, allow it to express those aspects of dialogue content required by a system such as the MMI^2 demonstrator, without excessively reducing the efficiency of its interpretation.

Acknowledgements

We are grateful to Professor Jacques Rouault (ADR/CRISS) for his stimulating contributions. We thank him.

References

Binot, J-L., Falzon, P., Perez, R., Peroche, B., Sheehy, N., Rouault, J. and Wilson, M.D. (1990). Architecture of a multimodal dialogue interface for knowledge-based systems. In *Proceedings of Esprit '90 Conference*, 412-433. Kluwer Academic Publishers: Dordrecht.

Gazdar, G. and Mellish, C.S., (1989). Natural Language Processing in PROLOG: an introduction to computational linguistics. Addison-Wesley: Wokingham.

Hobbs, J.R. (1985) Ontological Promiscuity. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, June 1985, 61-69.

Warren, D.H.D. and Pereira, F.C.N. (1982). An efficient, easily-adaptable system for interpreting natural language queries. *American Journal of Computational Linguistics*, 8(3-4), 110-22.

Williams, A. and Lambert, S.C. (1988). Expressive Power and Computability. In Ringland, G.A. and Duce, D.A. (eds.), *Approaches to Knowledge Representation*, 223-234. Research Studies Press: Letchworth, Herts.

Wilson,M.D. and Falzon, P. (1991) Multimedia and Multimodal Systems: Architectures for Co-operative Dialogue, *Proceedings of the ERCIM Workshop on Multimedia System, Lisbon*, Nov. 1991, 149-153.