AN APPROACH TO MULTILEVEL SEMANTICS FOR APPLIED SYSTEMS

Alberto Lavelli **Bernardo Magnini** **Carlo Strapparava**
IRST, Istituto per la Ricerca Scientifica e Tecnologica
I - 38050 Povo TN, Italy
e-mail: magnini@irst.it

Abstract

Multilevel semantics has been proposed as a powerful architecture for semantic analysis. We propose a methodology that, while maintaining the generality of the multilevel approach, is able to establish formal constraints over the possible ways to organize the level hierarchy. More precisely, we propose a "strong" version of the multilevel approach in which a level can be defined if and only if it is possible to characterize a "*meaningfulness*" notion peculiar to that level. Within such an architecture each level reached during the analysis computes its meaningfulness value; this result is then handled according to modalities that are peculiar to that level.

The component described in this paper was designed to be portable with respect to the application domain and so far has been tested as the semantic analysis component of two multimedial dialog systems, ALFresco and MAIA.

1. Introduction

Multilevel semantics has been proposed [Scha, 1983] as a powerful architecture for semantic analysis. In this approach, interpreting a natural language sentence is a multi-stage process, which starts out with a high-level meaning representation that reflects the semantic structure of the sentence rather directly. Then translation rules, which specify how the language-oriented semantic primitives relate to those used at deeper levels of analysis, are applied. One of the advantages of the multilevel approach is that it allows a natural decomposition of complex tasks and the functional modularization of semantic analysis. However, when multilevel architecture is used in concrete applications, a simple functional approach does not solve the problem of a clear definition of the semantics for each level. This fact is evident for applied systems whose semantic component must deal with many linguistic phenomena (e.g. lexical and structural ambiguities, quantifier scoping, anaphorical references, discourse topic and focus, referent retrieval, etc.). In such systems the definition of the semantics for a level has at least two advantages: (i) modules for specific phenomena could be easily introduced within the appropriate level, provided that the module functions contribute to the definition of the semantics for that level; (ii) a better understanding of the semantic analysis would be allowed: particularly, when a sentence is rejected at a certain level, it would mean that the semantic constraints for that level have been violated.

In this paper we suggest a methodology that, while maintaining the generality of the multilevel approach, is able to establish formal constraints over the possible ways to organize the level hierarchy. More precisely, we propose a "strong" version of the multilevel approach in which a level can be defined if and only if it is possible to characterize a "*meaningfulness*" notion peculiar to that level. Within such an architecture each level reached during the analysis computes its meaningfulness value; this result is then handled according to modalities that are peculiar to that level.

We shall show how our approach to multilevel semantics is concretely applied to organize the semantic component developed by the NLP group at IRST; this component is currently responsible for semantic analysis in two dialog systems, ALFresco and MAIA. At present two levels are included in the semantic component and they will be described in detail: the lexical level and the logical-interpretative level. At the *lexical level* the meaningfulness is defined by the *consistency* notion, which is computed by means of the lexical discrimination module; this module tries to select only the sentence readings meaningful in a given Domain Model (DM). When the propositional content of the sentence is proven to be consistent, the semantic representation produced by this level is passed to the next one; otherwise, if consistency cannot be proved, the whole sentence is rejected. At the *logical-interpretative level* the meaningfulness is defined by means of the *validity* notion, which is satisfied when referents for the sentence are identified. Three modules interact at this level: the quantification module, which finds the correct interpretation of the quantifiers, resolving possible scoping ambiguities; the topic module, which organizes the mentioned referents; the interpretation module, which identifies the part of the sentence to extensionalize and is responsible for referent retrieval. At this level, when validity cannot be proved, a special pragmatic procedure is activated.

Section 2 surveys a few relevant approaches to multilevel semantic analysis. In Section 3 the formal requirements for the "strong" multilevel semantics version are introduced. The architecture and the functional modules of the two levels of the semantic component we have developed are described in Sections 4 and 5. Finally, Section 6 deals with the two systems in which the semantic component has been used and Section 7 outlines some future developments.

2. Multilevel Semantics Applied

One of the first and most direct multilevel-based systems is the BBN spoken language system [Boisen *et al.*, 1989]. At every level of analysis, the meaning of an input utterance is represented as an expression of a logical language; the languages used at the various levels of analysis differ in that at every level the descriptive constants are chosen so as to correspond to the semantic primitives assumed at that level. At the highest semantic level, the meaning of an input utterance is represented as an expression of the English-oriented Formal Language (EFL). The constants of EFL correspond to the descriptive terms of English. An important feature of EFL is that descriptive constants are allowed to be ambiguous. The logical language used at the domain-dependent level of representation is called the World Model Language (WML). This is an unambiguous language, with an ordinary model-theoretic interpretation. Its constants are chosen to correspond to the concepts that constitute the domain of discourse. During the crossing of the EFL and the WML level (when domain dependent rewriting rules are called), the discrimination process is carried out. A type checking mechanism provides acceptance only for interpretations for which a domain knowledge compatible type has been computed. A further step of translation occurs when the WML is translated into DBL (DataBase Language) used to access the database to retrieve appropriate answers.

While having sound theoretical foundations, the main drawback of this approach is that it postpones semantic discrimination until domain knowledge is available; in the meantime, a complete sentence representation is built for each analysis the parser produces. However, IRUS-II [Ayuso *et al.*, 1989], an applicative system also developed at BBN, confirms that in a real system it is useful to connect the discrimination process to the parser. It implements a rule system that translates each syntactic constituent directly into a WML form, skipping the domain independent level of representation. While this solution improves system efficiency, lexical discrimination is carried out by domain dependent rules in a way that limits system modularity.

Another system with a clear distinction between the domain independent and the domain dependent level is XTRA [Allgayer *et al.*, 1989]. However, in this case at each level the same language (i.e. the knowledge representation language SB-ONE) is used. The domain independent level, called Functional-Semantic Structure (FSS), is intended as an intermediate structure that incorporates linguistic knowledge, substantially invariant in respect to the particular application domain. On the contrary, the domain dependent level, called Conceptual Knowledge Base (CKB), is necessary to adequately model the relations of the underlying expert system. In XTRA it is necessary that each analysis produced by the parser is consistent with the FSS level: this is achieved by means of a classification of the sentence instance with the SB-ONE mechanisms (the realizer and the matcher). If the classification succeeds, the analysis goes on to the CKB level, otherwise the syntactic analysis is rejected. In this approach the discriminatio process is profitably anticipated, and a powerful (eve though computationally expensive) consistency checkin mechanism is provided.

Both systems exploit the difference between knowledg about the application domain and knowledge that i independent of the particular domain (e.g., linguisti knowledge). Although this distinction is relevant fo allowing portability to different application domains, th semantic component described here focuses on the effect that domain dependent knowledge has on the type checkin mechanism.

To make the problem clearer, let us consider ho domain knowledge is exploited in the systems jus described. In the BBN spoken language system the typ checking is carried on by means of domain knowledge; o the other hand, within the XTRA system the discriminatio process is based only on domain independent knowledge. W think that an effective discrimination process should also b based on the application domain, it being unclear how t assign a proper meaning to a sentence without having fixe a particular context. Moreover, it seems useful to conside lexical discrimination as an incremental process: i discrimination works in parallel with the parser, it i possible to discriminate over single syntactic phrase checking the semantic content of each phrase.

From the previous remarks, it can be noted that system that employ the multilevel semantics approach can assig the same functionalities to different levels. Hence, it coul be useful trying to define the relations among each level in "stronger" way, facing the problem of coherenc maintenance.

3. Definitions of Meaningfulness

We have seen that in a multilevel semantics approac the main idea is to divide different functionalities int distinct levels. We propose a "strong" approach to such methodology in which for each level the definition of semantics is required. This is achieved by means of th assignment of a proper *meaningfulness* notion that define the semantic behavior of the level. In other words a level i a multilevel semantics hierarchy can be identified, if an only if it is possible to characterize a meaningfulness notio peculiar to that level. We have defined theoretically such notion for two levels: the lexical level and logica interpretative level (called *consistency* and *validit* respectively).

Let T be a theory of types that models our domain. I our multilevel semantics the notion of consistency is mea to demonstrate that an expression, representing th propositional content of a sentence, has type; i.e. given a expression w, it means to assign a type, if possible, to according to our type system. An expression has n meaning at the lexical level, if the type checking fails.

Validity, i.e. the meaningfulness at interpretation leve means to give a description of the objects of the typ suggested by the lexical level. Such a description can be i terms of relations, sets or intensional expressio (mandatory for infinite denotations). An expression has n meaning at the logical-interpretative level if such description cannot be found.

As the meaning of a sentence is always relative to a level in the multilevel architecture, every level manages the acceptance or the rejection of a sentence in a different manner. As examples:

(1) *A mule paints a fresco*

The components of the sentence have the following types:

a mule : `Mule`, a fresco : `Fresco`,
to paint : `Painter` → `Painting`.

Given the fact that "mules cannot paint" (only painters can), the type checking mechanism fails to assign an appropriate type and this causes the meaningfulness for the lexical level not to be satisfied.

(2) *Show me a work painted by all the painters born in Florence*

Sentence 2 satisfies the lexical level, but not the logical-interpretative one, because no description of the referents of the sentence can be proposed, i.e. there is no painting painted by *all* the painters born in Florence.

Once the functionalities of the levels are theoretically stated, the implementative choices can be very different and subject to criteria of portability. Type checking can be made using logical formalisms such as typed λ-calculi or intensional logics (possibly exploiting Curry-Howard's isomorphism between typed λ-terms and intuitionistic logic [Hindley and Seldin, 1986]). The interpretation level can retrieve the referred elements using functional applications or some algebraic formalisms. However, these approaches, although well founded, may not be the right ones from an implementative point of view, especially for large integrated systems. For example one has to define 'a priori' a theory of admissible types but when the domain changes, the theory does too. Another way is to use a hybrid knowledge representation system. As will be clear in the next section, we refer to a terminological component (Tbox) in order to obtain the type checking and to an assertional component (Abox) in order to retrieve the relations that verify the analyzed expression. This choice allows us to parameterize the type checking according to the knowledge representation. Indeed the portability of the modules encourages this alternative. Another possibility (to be explored) is to use a data base instead of the Abox, exploiting relational data theories.

4. Lexical Level

The semantic component (see Figure 1) interacts with both a parser and a hybrid knowledge representation system that includes the domain knowledge. As we have already mentioned, the semantic component consists of two levels and each level includes one or more specialized modules. In the following we will give a description of the functionalities of the various levels and modules of the semantic component.

The lexical level [Lavelli and Magnini, 1991] incrementally interacts with the parser: whenever the parser tries to build a (partially recognized) constituent, the discrimination module is triggered to check the consistency of the semantic part of such a constituent.

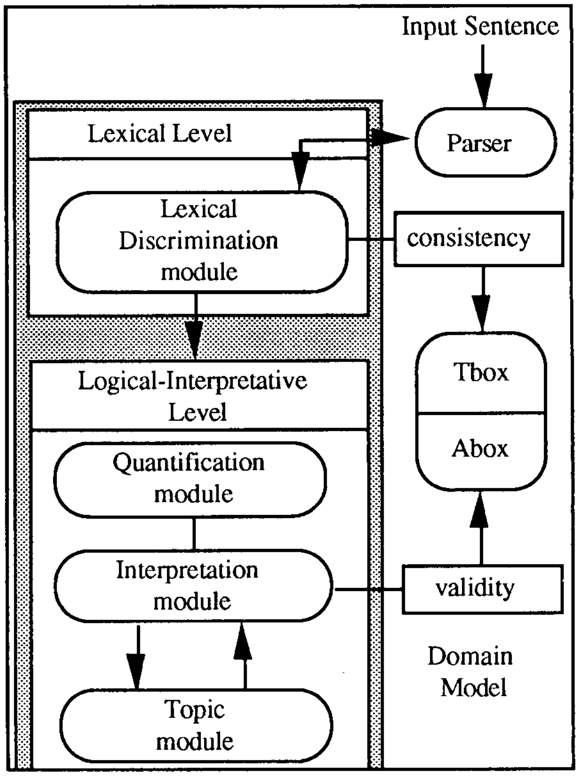

Figure 1: Sketch of the semantic process

4.1. Lexicon

The discrimination module uses semantic information from two different sources: lexical entries (which are domain dependent) and phrase-structure rules (which are domain independent). The representation produced by this module constitutes the input for the quantification module (at the logical-interpretative level) and is still neutral with respect to quantifier scopings.

Each lexical entry, along with the usual syntactic information (such as the lexical category of the word, the specification of the subcategorization frame of the entry, the superficial linguistic function that each subcategorized element holds) specifies a semantic representation and a mapping between syntactic functions and semantic functions. In such a way, within the semantic representation the syntactic distinction between the word complements (i.e. the arguments) and its adjuncts (i.e. its modifiers) is preserved.

As an example consider the simplified lexical entry for the verbal form "dipinse", *painted* (past tense) (see Figure 2). Morphological analysis enriches the information associated with the root and is able (for example in the case of passive) to change the mapping between linguistic functions and semantic functions. The semantic part of the lexical entry is built using the domain model knowledge (see Section 7 for a discussion on the portability problem) and it includes one (or more) semantic descriptions (this allows words with the same syntactic behavior, but different semantics, to be dealt with). Each semantic description contains the name of the DM concept (paint) associated with the word, along with its roles, which have a syntactic realization as arguments of the word and their restrictions (in this case, agent with restriction painter and goal with restriction painting).

dipinse
  category: V
  lingfunctions: ((subj agent) (obj goal))
  <other syntactic information>
  semantics: ((**paint** ((agent painter x)
      (goal painting y))) ^{lex}

Figure 2: Lexical entry for "dipinse": *painted.*

As for the rules, they also include both a syntactic and a semantic part. In the semantic part, the consistency is computed and the construction of the semantic representation is carried out. During this process, possible ambiguities taken from lexical items are reduced.

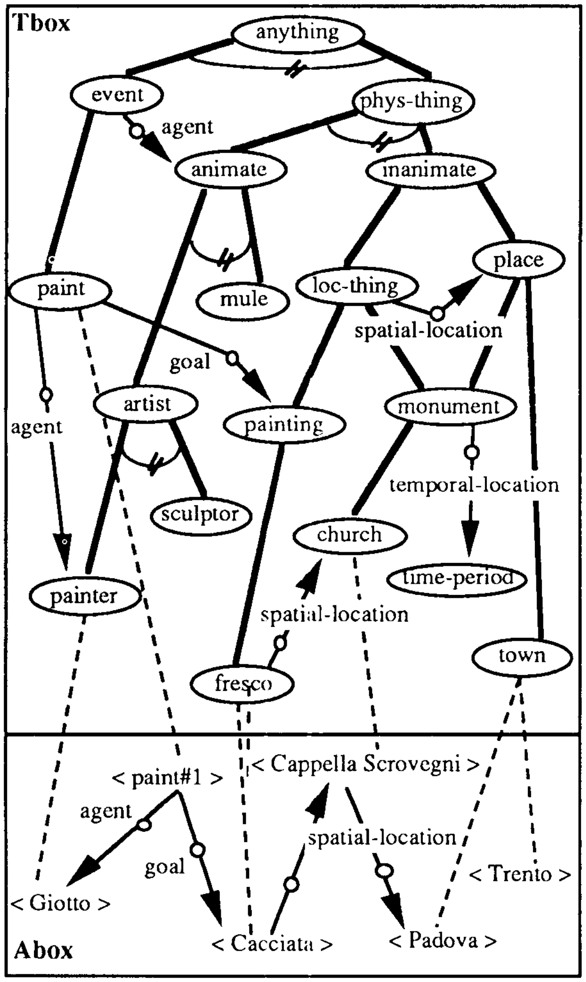

Figure 3: A fragment of the DM used in ALFresco.

4.2. Consistency checking

We define the consistency check operation such that it succeeds if selectional-restriction (i.e. the concept that represents the selectional restriction of a given argument position) denotes a concept that is compatible with the concept that semantic-head (i.e. the concept associated with the constituent which has to fill such a position) denotes.

There exist several possibilities to check the compatibility between two concepts within a terminological hierarchy. Within the JANUS system [Weischedel, 1989] the consistency is implemented by means of a double subsumption check that guarantees success both when semantic-head is a descendant of selectional-restriction and when it is an ancestor. This double subsumption test does not consider the cases, sometimes relevant, in which semantic-head is a brother concept of selectional-restriction (e.g. "Has a sculptor painted a fresco?"); this case recursively extends to all the cases in which semantic-head is a brother either of a descendant or of an ancestor for a selectional-restriction (e.g. "Which object did Giotto paint?"). This case is slightly more complex than the others. In fact, while it is always true that along the ISA hierarchy there can be a non-empty intersection between two concepts, this is not true for concepts that are brothers. If an explicit disjointness is placed between two brother concepts, there cannot be a common intersection and the consistency procedure must fail; otherwise it is assumed that a common intersection can exist, and the consistency-test procedure will succeed. KR languages with disjointness are usually provided with a specific predicate holding between two concepts when their intersection is empty. It is worth noting that this predicate includes all the subsumption cases among concepts, in which cases it is always false.

Now we will illustrate how the whole process works using Sentence (3) (in the rest of the paper, all the examples refer to the DM knowledge in Figure 3; we will use "`concept`" characters to indicate DM objects):

(3) 'Mostrami tutti gli affreschi dipinti da Giotto in un monumento di Padova'
Show me all the frescoes painted by Giotto in a monument of Padova

In this sentence there is a typical case of ambiguity, that of the preposition 'di' (of); at least two senses for 'di' are possible in DM: the spatial interpretation, in which the mapping is to the `spatial-location` role, and the temporal interpretation, in which the mapping is to the `temporal-location` role. The selection of the right interpretation (the spatial one) is carried out through the application of the consistency check between the argument selectional-restrictions (the domain and the range of a role) and the semantic-head that tries to fill the position. In this case the temporal interpretation is rejected (it does not satisfy the meaningfulness notion for the lexical level) because the range restriction (`time-period`) is not consistent with the proposed semantic-head (`Padova`).

4.3. First logical form

The final result of the lexical level is a form that uses a predicate-argument notation that allows abstracting from time and context. Omitting for the moment the intensional aspects, four relevant constructs for the resolution of quantifiers and definite/deictic referents are:

(**complex-term** <features>
    <quantifier><variable><restriction>)

(**ref-term** <features><variable><restriction>)
(**demonstr-term**<features><variable><restriction>)
(**pronoun-term** <features>
        <variable><pred-restriction>)

A complex-term represents a quantified NP (see Figure 4). A ref-term represents a definite NP. It plays an important role at interpretation level (see Section 5.3). A demonstr-term has the task of representing a demonstrative NP. The representation has to take into account the possible multimediality that the system treats at this level (the touch on the touchscreen for a deictic reference). A pronoun-term represents a pronoun. The lexical level gives a suggestion with <pred-restriction> on the type of semantic restriction that the bound variable can have. Then this information will be used by the interpretation module. The <features> keep syntactic information of the NP ready for use in the interpretation module.

```
(show hearer
  (complex-term all x
    (and (fresco x)
      (paint Giotto x)
      (spatial-location x
           (complex-term indef y
               (and (monument y)
               (spatial-location y Padova)))))
  speaker)
```

Figure 4: Output of the lexical level for Sentence (3).

The resulting form produced by the lexical level for Sentence (3), omitting the <features> information, is shown in Figure 4.

5. Logical-Interpretative Level

At this level validity of the sentence is checked using the knowledge in the DM Abox. Verifying the validity of a logical form and producing the correct interpretation is not a trivial task. We want the semantic interpreter to be independent of the domain of the knowledge representation system and of the different media through which a linguistic expression can be built. This process involves the interaction of the three modules of this level shown in Figure 1 [Strapparava, 1991].

5.1. Quantification Module

Within the quantification module, an algorithm for the resolution of quantifier scopings generates all possible readings and for each quantifier it shows its range over the rest of the sentence. However, to get an acceptable number of readings (possibly only one), the scoping generation algorithm, which takes advantage of the idea of Cooper storage, needs some heuristics based on linguistic/semantic knowledge. These rules must be seen as a whole, i.e. they strictly interact with each other. Moreover they *suggest* a disambiguation, they do not always ensure it. Some rules can be: 1) lexical relevance of the quantifiers; 2) syntactic position of quantified NPs; 3) scope markers; 4) distributive/collective semantics of predicates. The readings are put in order of soundness according to a hierarchy of rules.

The scoping resolution algorithm produces a second logical form in which all complex-terms are resolved, making their scope and body explicit. In this logical form for each quantifier a quantified formula appears with the following structure:

(**quant** var restriction body)

For example, the reading for Sentence (3) in which for each fresco there exists a monument that includes it, is shown in Figure 5.

```
(all x (indef y (and (monument y)
                (spatial-location y Padova))
          (and (fresco x)
               (paint Giotto x)
               (spatial-location x y)))
    (show hearer x speaker))
```

Figure 5: Second logical form for Sentence (3).

5.2. Interpretation Module

The interpretation of the logical form built by scoping resolution makes up a level in which the validity of a sentence is detected and eventually the relative referents are retrieved (possibly interacting with the topic module in order to get referents for linguistic expressions such as definite NPs and personal pronouns). The expressions are mapped into the KR assertional language. The main task of the interpretation module is the interpretation of the logical form operators, giving a set of possible candidates that logically satisfies the sentence for each NP. The results are then notified to other modules of the system (i.e. the pragmatic component).

The interpretation of the operators includes the quantifier interpretation (existential, universal, numerals and natural quantifiers). The restriction of a quantified formula is calculated and the result is logically verified in the body according to the semantics of the quantifier operators. Since there may be an arbitrary nesting of quantifiers in the second logical form of a sentence, the algorithm has to provide an arbitrary deep recursion of such functionalities. (Indeed the interpretation module has other important tasks. One of the improvements under development consists of embedding intensional aspects into the logical form. These intensional aspects tend to extend the characteristic of an extensional logical form by allowing references to time and contexts (indexicals) [Forbes, 1989, Stallard, 1987]. They would also include the possibility of interpreting certain NPs along the attributive/referential dimension).

For a detailed description of the algorithms of the logical-interpretative modules see [Strapparava, 1991]. Now we want to focus on the interpretation of quantifier operators. According to the semantics of these operators the interpretation module checks the validity of a sentence.

5.2.1. Semantics of quantification operators

The notation that will be used in discussing the semantics of quantification operators is given below:

- `pred[x]` indicates a well-formed form in which the variable x appears free;
- $\mathbf{ext}(\lambda x.p[x])$ indicates the extension of p in a representation domain DM;
- $\mathbb{P}(\lambda x.p[x])$ indicates the set of the parts of the extension denoted by p;
- be `I` a set, $|I|$ indicates its cardinality.

We shall show how semantics is assigned to the quantification operators in the logical form.

As seen above a quantifier is syntactically represented with the wff

$$(\mathbf{quant}\ x\ \texttt{restriction}[x]\ \texttt{body}[x])$$

that has a semantic interpretation

$$\{a \in \mathbb{P}_{\mathbf{quant}}(\lambda x.\texttt{restriction}[x])$$
$$\mid \texttt{body}[x/a] \text{ is verified in DM}\}$$

where $\mathbb{P}_{\mathbf{quant}}(\lambda x.\texttt{restriction}[x])$ is appropriately defined for each treated quantifier.

The quantification operators that can appear in the logical form are universal and existential quantifiers, wh-operators, natural quantifiers such as numerals (two, three...), exception operators (all except three ...), vague operators (many, several, most of...). As an example we show how semantics is assigned to the quantifier 'many'.

About 'molti' (many) there can be two attitudes: either one excludes this type of quantification by an extensional treatment [Keenan and Stavi, 1986] or one tries to get what 'many' means in a fixed context [Barwise and Cooper, 1981]. In our approach this second consideration was followed. Therefore

$$\mathbb{P}_{\text{many}}(\lambda x.p[x]) =$$
$$\{P \in \mathbb{P}(\lambda x.p[x]) \colon |P| = k|[\mathbf{ext}(\lambda x.p[x])]|\}$$

where the multiplier k may be fixed, for example 0.6, or may depend on pragmatic aspects or on inferences on the semantic structure of the dialog.

Also to interpret other vague operators extensionally, it is necessary to make a stipulation of cardinality (fixed or dynamic), depending on the inferential capabilities of dialog structure at our disposal. For example almost all, most of may be interpreted as semantically similar to 'except at most K', where for K considerations similar to those made for 'many' hold.

For example, we can apply the operators to sentence (3). According to the domain DM, if the interpretation module may construct a and a', i.e. to construct a description of the referents of the sentence, the validity for sentence (3) holds. If the sentence were 'Mostrami tutti gli affreschi dipinti da Giotto in un monumento di Trento' (*Show me all the frescoes painted by Giotto in a monument of Trento*) it would be consistent, but the logical-interpretative level would have found it not valid (because there are no frescoes by Giotto in Trento). The interpretation module would not be able to construct a description of the referents of the sentence.

5.3. Topic Module

The ref-terms, demonstr-terms and pronoun-terms are treated specially. The demonstr-terms coming from a deictic gestuality (i.e., in our systems a touch on a touchscreen; see Section 6) contain the entities to which the user intended to refer. These are passed to the interpretation module to verify the semantic consistency. The demonstr-terms without touch, the pronoun-terms and some ref-terms are resolved with strict interaction with the topic module. The topic module organizes the mentioned referents so that it offers plausible candidates for these operators and the interpretation module verifies their semantic soundness. For a detailed description of the topic module, see [Samek and Strapparava, 1990]. First of all, the constants in the logical form (in our example: Giotto and Padova) are passed to the topic module. Later on the topic module is asked to give a set of probable candidates for the terms in the logical form coming from a deictic gestuality and from the terms coming from pronouns. The interpretation then will test their validity.

$$\left\{ a' \in \mathbb{P}_{\text{exist}} \boxed{\begin{array}{l}\lambda y.\text{(and (monument [y])}\\ \qquad\text{(spatial-location [y] Padova)}\end{array}} \right) \mid \text{DM verifies} \boxed{\begin{array}{l}\text{(and (fresco } \mathbf{a}\text{)}\\ \qquad\text{(paint Giotto } \mathbf{a}\text{)}\\ \qquad\text{(spatial-location } \mathbf{a}\ \mathbf{a}'\text{))}\end{array}} \right\}$$

where **a** is

$$\left\{ a \in \mathbb{P}_{\text{all}} \left(\boxed{\begin{array}{l}\lambda x.\text{(indef y (and (monument y)}\\ \qquad\qquad\text{(spatial-location y Padova))}\\ \qquad\text{(and (fresco [x])}\\ \qquad\qquad\text{(paint Giotto [x])}\\ \qquad\qquad\text{(spatial-location [x] y)))}\end{array}} \right) \mid \text{DM verifies (show hearer } \mathbf{a} \text{ speaker))} \right\}$$

Figure 6: Quantification operators applied to Sentence (3).

6. Application on Different Complex Systems

The semantic component described in this paper has been used within two different prototypical dialog systems (i.e ALFresco and MAIA).

ALFresco is an interactive system for a user interested in frescoes. It is connected to a videodisc unit and a touchscreen. The videodisc includes images of Fourteenth Century Italian frescoes and relevant monuments and hypertext includes art critics' comment. A general description of the functionalities and finalities of the ALFresco system can be found in [Stock, 1991].

MAIA is the IRST global project. It is conceived as an integration of components being developed here in different fields of AI (speech recognition, natural language, KR, vision, reasoning, etc). It consists of both a mobile part (a robot moving in the corridors of the institute) and a central part (a kind of "concierge" with whom a visitor may enter into a dialog about the institute). The tasks of the concierge are: (i) giving information about researchers' activities and institute organization; (ii) supervising the robot's activities; (iii) interacting with an electronic librarian in order to find relevant books. The initial paradigm for the concierge interaction is related to that of ALFresco, but of course the situation and media are different. As the project evolves natural language dialogs will also include direct interaction with the robot (whose role is to accompany the visitor to some office or deliver parcels) and an integration with speech recognition and synthesis. Within this more complex situation, the NLP component has to increase its capabilities in order to cope with aspects such as multiple access to information and interaction with the robot planner.

Both systems have a common architecture design and have been implemented in CommonLisp within the Medley environment running on Sun 4. The main components interacting with the semantic component described here are a parser and a hybrid knowledge representation system. Both for ALFresco and MAIA the parser WEDNESDAY 2 is used [Stock, 1989], a chart-based parser for the Italian language that can cope with complex sentences, idiomatic expressions, ellipsis, and so on.

As for knowledge representation, in ALFresco the YAK system [Franconi, 1990] is used, while in MAIA the LOOM system [McGregor and Bates, 1987] is used.

7. Conclusions and Future Work

We have presented an approach to multilevel semantics that was exploited in the development of two semantic levels for a dialog system architecture: the lexical level and the logical-interpretative level. The suggested methodology is able to establish formal constraints over the hierarchy by means of a local meaningfulness notion. Such a notion was defined for the lexical and logical-interpretative level, specified as consistency and validity respectively. Then how the functionalities of each level realize their own semantic definitions was explained in full detail. Finally two systems, ALFresco and MAIA, that use the semantic component were described.

Future developments of our work concern the issue of portability to different application domains. While the general inference mechanisms employed by both the lexical and the logical-interpretative level are designed to be domain-independent, the semantic lexicon contains information strictly connected with the domain of interaction. To (at least partially) automatize the construction of this semantic lexicon (given a particular DM), the possibility of using an approach similar to the Upper Model used by the PENMAN text generation system [Bateman *et al.*, 1990] is being investigated. The Upper Model establishes a level of linguistically motivated knowledge organization specifically designed for the task of constraining linguistic realizations. Given a certain application domain, the domain knowledge is mapped (classified) into the Upper Model knowledge; in this way, for each domain object a proper lexical realization is established. As a result, changing the application domain requires that only the mapping between the domain and the Upper Model knowledge is specified.

Further developments are connected with the use of natural language in a domain which implies an interaction with the physical world (as happens in the MAIA system). This kind of application will also raise the need to access both information gathered from the physical environment and dynamically changing knowledge and of a more complex pragmatic component, thereby stressing the need for a clear architecture. We are also working on the issue of integrating such expansions within the approach to multiple underlying systems (MUS) as established by [Bobrow *et al.*, 1990, Resnik, 1989]. In the MUS approach, a user may need to combine the capabilities of more than one system (i.e. several DBs on various domains, expert systems, information retrieval systems, interfaces to simulation packages, etc.) in order to perform a general task. For dealing with MUS, not only our semantic modules must be able to represent various levels of meaning of a sentence, they must also be capable, in a transparent manner, of organizing the different applications at their disposal and choosing which combination of them to use.

References

[Allgayer *et al.*, 1989] Allgayer, J., Jansen-Winkeln, R., Reddig, C., Reithinger, N. "Bidirectional Use of Knowledge in the Multi-Modal NL Access System XTRA". In *Proceedings of IJCAI-89*, Detroit, Michigan, 1989.

[Allgayer, 1990] Allgayer, J. "SB-ONE$^+$ - dealing with sets efficiently". In *Proceedings of ECAI-90*, Stockholm, Sweden, 1990.

[Ayuso *et al.*, 1989] Ayuso, D., Donlon, G., MacLaughlin, D., Ramshow, L., Resnik, P., Shaked, V., Weischedel, R. "A Guide to IRUS-II Application Development". Report No. 7144, BBN System and Technologies Corporation, 1989.

[Barwise and Cooper, 1981] Barwise, J., Cooper, R. "Generalized quantifiers and natural language". *Linguistics and Philosophy*, 4, 1981.

[Bateman *et al.*, 1990] Bateman, John A., Kasper, Robert T., Moore, Johanna D., Whitney, Richard A. "A General Organization of Knowledge for Natural Language Processing: the PENMAN Upper Model". ISI Technical

Report, USC/ Information Sciences Institute, 1990.

[Bobrow *et al.*, 1990] Bobrow, R., Resnik, P., Weischedel, R. "Multiple Underlying Systems: Translating User Requests into Programs to Produce Answers". In *Proceedings of ACL-90*, 1990.

[Boisen *et al.*, 1989] Boisen, S., Chow, Y., Ingria, R., Roukos, S., Scha, R., Stallard, D, Vilain, M. "Integration of Speech and Natural Language Final Report". Report No. 6991, BBN System and Technologies Corporation, 1989.

[Cooper, 1987] Cooper, R. *Quantification and Syntactic Theory*. Reidel, Dordrecth, 1983.

[Forbes89] Forbes, G. "Indexicals". In: Gabbay & Guenthner (eds.), *Handbook of Philosophical Logic IV*. Reidel, Dordrecth 1989.

[Franconi, 1990] Franconi, E. "The YAK (Yet Another Krapfen) Manual". IRST Manual, Trento, Italy, 1990. Also as 'Progetto Finalizzato CNR - Sistemi Informatici e Calcolo Parallelo'.

[Hindley and Seldin, 1986] Hindley, J., Seldin J. *Introduction to Combinators and λ-Calculus*. Cambridge University Press, 1986.

[Hobbs and Shieber, 1987] Hobbs, J., Shieber, S. "An Algorithm for Generating Quantifier Scopings". *Computational Linguistics*, 13, January 1987.

[Keenan and Stavi, 1986] Keenan, E., Stavi, J. "A semantic characterization of natural language determiners". *Linguistics and Philosophy*, 9, 1986.

[Lavelli and Magnini, 1991] Lavelli, A., Magnini, B. "Lexical Discrimination within a Multilevel Semantics Approach". *Proceedings of AIIA-91*, Palermo, Italy, 1991.

[Mac Gregor, Bates, 1987] Mac Gregor, R.M., Bates, R. "The LOOM Knowledge Representation Language". Techical Report ISI/RS-87-188, USC/Information Science Institute, 1987.

[Resnik, 1989] Resnik, P. "Access to Multiple Underlying Systems in Janus". BBN Report No. 7142, 1989.

[Samek and Strapparava, 1990] Samek-Lodovici V., Strapparava C. "Identifying Noun Phrase References: The Topic Module of the AlFresco System". *Proceedings of ECAI-90*, Stockholm, 1990.

[Scha, 1983] Scha, R. "Logical Foundations for Question Answering". Philips Research Laboratories, M.S. 12.331, Eindhoven, The Netherlands, 1983.

[Scha and Stallard, 1988] Scha, R., Stallard, D. "Multilevel Plural and Distributivity". In *Proceedings of ACL-88*, 1988.

[Stallard, 1987] Stallard, D. "Answering Questions Posed in Intensional Logic. A multilevel semantics approach". BBN Report No. 6522, June 1987.

[Stock, 1989] Stock, O. "Parsing with Flexibility, Dynamic Strategies, and Idioms in Mind". *Computational Linguistics*, 15(1): 1-18, 1989.

[Stock, 1991] Stock, O. "Natural Language and Exploration of an Information Space: The ALFresco Interactive System". In *Proceedings of IJCAI-91*, Sydney, Australia, 1991.

[Strapparava, 1991] Strapparava, C. "From Scopings to Interpretation: The Semantic Interpretation within the ALFresco System". In *Proceedings of AIIA-91*, Palermo, Italy, 1991.

[Weischedel, 1989] Weischedel, R. M. "A Hybrid Approach to Representation in the Janus Natural Language Processor". In *Proceedings of ACL-89*, Vancouver, British Columbia, 1989.

[Westerstahl, 1986] Westerstahl, D. "Quantifier in Formal and Natural Language". Report No. CSLI-86-55, June 1986.